Creating a Simulation Environment to Analyze Benefits of Real-time Taxi Flow Optimization Using Actual Data

G.J.M. Koeners and R.M. Rademaker

Delft University of Technology, Mekelweg 4, Delft, The Netherlands

To make optimal use of the available runway and taxiway infrastructure and to prevent the workload of the controllers becoming a limitation, a route-planning function that generates routes automatically is needed. The uncertainty in the taxi process is an important disturbance in today’s surface operations. To allow for deconfliction in the planning domain, the routes need to include time information. The route-planning function will need to use time-margins to provide a certain level of robustness against disturbances. A determining factor in the selection of the size of the margins is the tradeoff between the (maximum) capacity that can theoretically be achieved in the absence of disturbances and the robustness against disturbances. A key challenge is the identification of the relation needed to specify time margins as a function of traffic density and disturbance statistics. An important uncertainty of the taxi processes is the taxi speed. This paper will describe a method to obtain and quantify taxi traffic flow performance and taxi speed distribution using real world data from Amsterdam airport. Furthermore the development of a simulation environment to evaluate the performance of various route-planning functions for selectable time-margins and disturbance statistics is described. In addition a method is described to extract surface traffic scenarios from real world data. The scenarios are used to verify the fidelity of the surface traffic simulation.

I. Introduction

Nowadays, traffic control on the airport surface is tactical rather than strategic, preventing the optimal use of the available taxiway and runway infrastructure. This leads to delays, extra fuel consumption and unnecessary burdening of environment. The goal of the project “Safe Airport Navigation Implementation using an Evolutionary ICT Enabled Approach” (SANI) is to contribute to the international effort that focuses on the development of technological systems to enable optimization of the airport surface traffic flow. In the ideal situation, ATC is capable of realizing traffic flows that are optimal with regard to usage of the available taxi- and runway infrastructure. To realize these optimal traffic flows and to prevent the workload of the controllers becoming a limitation, a route-planning function that generates routes automatically, while allowing for manual modifications is needed. The research described in this paper is part of the multi year SANI project.

Several research projects have addressed the automatic generation of taxiroutes, but typically the solution is limited to the geographical domain. But to allow for deconfliction in the planning domain, the routes need to include time information. Part of the timing optimization was looked at by other research by focusing on open-loop optimization of arrivals and departures, leading to so-called Arrival Managers (AMAN) and Departure Managers (DMAN) that support traffic control in scheduling the different aircraft. These tools typically advice a controller when an aircraft should depart the gate for example. This open-loop approach assumes a fixed taxi speed or models the taxi speed based on previous observations. The uncertainty in taxi speed is an important factor in today’s taxi operation and as a consequence it is necessary to create a buffer of airplanes at the runway entry to prevent the runway not being utilized efficiently. To reduce taxi time uncertainties in Ref. 9 a method is proposed to optimize the traffic flow by determining optimal times for each aircraft to reach significant control points along its taxi route. But this open-loop approach assumes the exact execution of the generated plan. In the future, support systems on board the aircraft can be used to aid the pilot in complying with the 2DT route. However it is expected that the variance in aircraft compliance with 2DT routes will be larger than the compliance errors of the autopilots flying

---

1 Researcher, Electronic Navigation Systems Group, AIAA Member.
2 Researcher, Electronic Navigation Systems Group.
airborne 4DTs. Furthermore, it is likely that significant uncertainty in flight readiness, pushback processes and airport surface operations in general will still exist, as they do today. Our proposed approach essentially closes the traffic control loop with a higher bandwidth by using a datalink to provide the pilot with speed advisories and traffic control with accurate real-time information about the aircraft position and velocity. This closed loop approach makes it possible to compensate for disturbances. The SANI project will aim to achieve this. Nowadays, in order to minimize the negative effects of disturbances, controllers adopt a strategy that involves planning ahead for certain envisioned conflicts, and by creating buffers (e.g. a number of aircrafts at the runway). Similar to the ‘planning ahead’ of the operator, the route-planning function will need to use time-margins to provide a certain level of robustness against disturbances. A determining factor in the selection of the size of the margins is the tradeoff between the (maximum) capacity that can theoretically be achieved in the absence of disturbances and the robustness against disturbances. Tighter constraints allow for higher throughput, but because of the increased sensitivity to small deviations, disturbances are more likely to cause a conflict that requires replanning. However, the adaptability enabled by the use of the datalink to dynamically provide speed advisories can compensate for the disturbances. In other words, there are two mechanisms that can be used to deal with the uncertainties in the openloop plan: pre-defined margins and closed-loop adaptability. To prevent the use of unnecessarily tight margins, an important design goal of the route-planning function is to take the expected demand for movements and statistics about the disturbances into account. The configuration of the route-planning function will impact the productivity that can be achieved. Hence, a key challenge is the identification of the relation needed to specify time margins as a function of traffic density and disturbance statistics. To help identify this relation and to assess the potential improvement that can be achieved by a traffic flow planning function, a simulation environment was needed.

This paper will describe a method to obtain and quantify taxi traffic flow performance and taxi speed distribution using real world data from Amsterdam airport. Furthermore the development of a simulation environment to evaluate the performance of various route-planning functions for selectable time-margins and disturbance statistics described. In addition a method is described to extract surface traffic scenarios from real world data. The scenarios are used to verify the fidelity of the surface traffic simulation.

II. Current Taxi Flow

Surface operation constraints have an influence on the traffic flow optimization. As a case study we looked at Amsterdam airport. The current taxi process at Amsterdam airport was analyzed to identify the relevant surface operation constraints and to establish rating criteria to be used in future simulations of new traffic flow optimization algorithms. A method is described to obtain traffic performance data using real-data from Amsterdam airport.

A. Current Taxi Process

An understanding of the taxi process that is used today is needed to identify the traffic flow performance parameters and surface operation constraints. Interviews with a ground controller from Air Traffic Control the Netherlands (LvNL) and tower operation observations resulted in a schematic overview of the taxi process that is currently used at Amsterdam Airport (Fig. 1).

Figure 1. Current taxi process.
1. Surface Operation Constraints

Since at Amsterdam airport the inbound traffic flow is handled outside the ground operation the current focus of this research is on optimizing the outbound traffic with the inbound traffic as a constraint. Inbound and outbound traffic are using different runways. Figure 1 shows a schematic overview of how currently the outbound traffic is planned. The inbound traffic in this schematic overview is a constraint in the form of ‘other traffic’.

At the beginning of the taxi planning process the airline files a flight plan (FLP). At Amsterdam airport most flights are given a Calculated Take-Off Time (CTOT) by the Central Flow Management Unit (CFMU). The CTOT depends on airspace and airport use and thus is a constraint in the taxi optimization process. A CTOT will be given to a flight roughly 30 minutes before a departure. An aircraft has to be airborne within a window around the CTOT, which is normally referred to as a slot. This slot time is from 5 minutes before the assigned CTOT till 10 minutes after. If an aircraft is not capable of reaching this target the airline has to ask for a new CTOT from the CFMU. Since this has a big influence on the flight delay, getting the aircraft airborne within the slot time is an important factor in the taxi process.

The runway configuration at Amsterdam airport during normal operation is planned in advance. Factors that influence the configuration are expected traffic flow (inbound versus outbound) and noise regulations. Since the runway configuration is determined outside the taxi process it is considered as a constraint. If more than one runway is in use for departures the runway assignment is done by the departure planner and depends on the flight plan and the connected Standard Instrument Departure (SID).

An important limiting factor for the amount of traffic that can be handled at Amsterdam airport is the operation on the runway. The sequence of aircraft that depart from a runway has an influence on the throughput. The minimum separation between two departing aircraft (and thus the time the runway cannot be used) depends on the type of aircraft and used SID. If two aircraft departing from the same runway are using the same SID then the runway separation time is longer than when two different departure routes are used. The different separation times depending on the aircraft sequence are shown in Table 1. At Amsterdam airport the aircraft category ‘light’ typically does not use the main runways and therefore does not play an important role in the scheduling process. Because of the layout of Amsterdam airport the crossing of an active runway in typical operations is not common and is not taken into account in the first phase of the SANI project.

At Amsterdam airport standard taxi routes are used. The departure gate and used runway mainly determine the taxi route.

<table>
<thead>
<tr>
<th>Aircraft sequence</th>
<th>Minimum Separation (in minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>medium following medium</td>
<td>1</td>
</tr>
<tr>
<td>heavy following medium</td>
<td>1</td>
</tr>
<tr>
<td>heavy following heavy</td>
<td>1.5</td>
</tr>
<tr>
<td>medium following heavy</td>
<td>2</td>
</tr>
<tr>
<td>same SID any combination</td>
<td>2</td>
</tr>
<tr>
<td>different SID any combo</td>
<td>1</td>
</tr>
</tbody>
</table>

2. Current Traffic Flow Optimization

Today the traffic flow is optimized by the controllers (the blue area in Fig. 1). Their goal is to get the aircraft safe to the runway and to maximize the capacity of the airport. When the startup controller receives a startup request, he/she uses aircraft start location, CTOT, aircraft type, SID, other traffic, estimated startup time and estimated taxi time to estimate the sequence on the runway(s). If the aircraft fits into the expected runway sequence and other (expected) traffic on the taxiways will not conflict, the controller gives a startup clearance. After push back the ground controller repeats the sequencing process and gives taxi instructions to the pilot. When the aircraft arrives at the runway the tower controller can influence the departure sequence by using different runway entry points. Because of uncertainty in taxi times or disturbances the controllers plan the aircraft in such a way that there are always a number of aircraft at the runway. The use of these buffers minimizes the chance that the runway is not used because there is no aircraft ready at the runway. It also gives the opportunity to optimize the sequence close to the runway (since taxi time uncertainties are smaller here). In case of a disturbance (e.g. aircraft is not ready for taxi in time to fit in the estimated sequence) the controllers will not distribute the delay over the other aircraft. If a new solution is not found within the slot window the delayed aircraft has to ask for a new CTOT from the CFMU.
3. Traffic Flow Performance Parameters

Based on the analysis of the taxi process the performance of a taxi flow optimization function can be measured by:

- number of aircraft that miss CTOT
- runway throughput
- resulting runway sequence (compared to optimal)
- amount of time aircraft wait at the runway
- amount of time aircraft wait for other traffic at taxiways

Missing a CTOT influences the traffic flow enroute and at the arrival airport and is therefore an important parameter. Since the runway is a limiting factor for airport capacity the runway throughput is a performance measurement for a taxi flow function. The throughput can be influenced by lack of aircraft supply at the runways or used sequence. Both are results of a taxi flow function and thus a performance measurement. The amount of time aircraft have to wait at the runway normally depends on buffers used by the traffic flow function and the used tradeoff between optimal sequence and aircraft waiting time. The amount of time aircraft have to wait on taxiways could be an indication of the coordination with other traffic including arriving traffic.

B. Analyzing Recorded Data

The current taxi flow data from Amsterdam airport were analyzed to get a baseline data set of performance measurements and to obtain a speed profile distribution. Surveillance data and data logged by various tower systems were used to analyze current traffic flow. The surveillance data consist of track information obtained by the Multilateration (MLT) system. In case of Amsterdam airport the MLT data are in the All Purpose Structured Eurocontrol Surveillance Information Exchange (ASTERIX) category 10 format. Figure 2 shows an overview of the elements used to extract the data from logged surveillance data and tower data.

As mentioned before the uncertainty in taxi times is an important factor that prevents the optimal use of the available taxiway and runway infrastructure. The obtained speed profile distribution can be used in the development of traffic flow functions. It is expected that due to operational constraints aircraft will not follow a precise speed profile, instead the average speeds between key points can be used. For traffic flow functions it is important to know at what time an aircraft will arrive at certain key points on the airport. The airport layout database in Fig. 2 consists of a set of nodes and connections describing the taxiway centerlines. The function block ‘Record Used Path’ in figure 2 uses this database and aircraft positions from the surveillance data to construct a set of nodes that describe the path the aircraft took. The function block ‘Record Speed Profile’ uses the same surveillance data and airport layout database to identify the speeds the aircraft taxied in between the nodes. This method results in a taxi route and used speed profile of every aircraft. The speed profiles can be used to construct a database of different speeds that were used between nodes. The result is a speed profile distribution database.
Using the ‘Performance Monitor’ block in Fig. 2 a baseline performance data set was obtained. The following methods were used to obtain the earlier mentioned performance parameters:

1. **Amount of Time Aircraft Wait at the Runway**

   For all the runway locations that are used for takeoff an area was defined. Figure 3 shows an example of the takeoff area of runway 36C. The area where the performance data were measured is indicated by the yellow box. All aircraft that traveled through the yellow box and took off from runway 36C are identified. The function block ‘Compute Queue Times’ in Fig. 2 uses the aircraft positions from the surveillance data to identify all the aircraft that have at least one position within the yellow box. To help identify the aircraft that took off from 36C a second box is used. All aircraft that also have at least one position in the red box are identified as aircraft that took off from runway 36C. The amount of time that an aircraft had to wait in the runway area is measured by the amount of time the speed of the aircraft was below 5 knots. At some runways at Amsterdam airport controllers can line up more than one aircraft at once. The aircraft at the front of the line up will take off first while the aircraft behind waits on the runway. The time that the aircraft is waiting on the runway is also taken into account in the delay. Figure 4 shows an overview off all the takeoff areas that were defined for Amsterdam airport.

2. **Amount of Time Aircraft Wait for Other Traffic at Taxiways**

   Similar to measuring the amount of delay at the runways an area is defined to measure the delay on the taxiways. Figure 4 shows the green box that is used by ‘Compute Queue Times’ in Fig. 2 to identify all traffic positions located in the specified area. The time aircraft wait at the gates is not included by excluding the positions located around the parking areas defined by the blue boxes. Traffic that entered the runway areas (yellow boxes) and did not takeoff from the runway in that area but did get delayed is counted as taxi delay.

3. **Runway Throughput**

   The runway throughput is measured by using the threshold areas (the red boxes in Fig. 4). The block ‘Detect Takeoff’ in Fig. 2 detects a takeoff when an aircraft leaves the red box. The time difference between the moment a new aircraft leaves the red box and the time the previous aircraft left the box is used to calculate the amount of takeoffs per hour.

4. **Resulting Runway Sequence and Number of Aircraft that Miss CTOT**

   The method that is used to detect a departing aircraft to measure the runway throughput is used to identify the runway sequence and aircraft that miss CTOT. When a departing aircraft is detected, information from the tower data is used to register the weight class and used SID. This way the resulting runway sequence is identified and can be compared to the optimal sequence. The same method is used to compare the CTOT information in the tower data to the time the departure of the aircraft was detected.
C. Results

The dataset that was available to us for this initial analysis only contained 2 hours of logged MLT data. During these 2 hours runway 36L was used for departure. We will receive a larger dataset in the near future.

Figure 5 shows a histogram showing what percentage of aircraft had a certain amount of total delay waiting in the queue area to takeoff from runway 36L. Figure 6 shows for that same group of aircraft the additional delay while waiting for other aircraft at the taxiways.

Figure 7 shows the runway throughput of 36L. Table 2 shows the minimal takeoff separation that was required as a result of the takeoff sequence. The amount of times that a certain aircraft sequence occurred is multiplied by the amount of delay required as shown in table 1. In the 2 hours of logged data none of the aircraft that took off from 36L missed their CTOT.

Table 2. Minimal takeoff separation 36L that was required as a result of the takeoff sequence.

<table>
<thead>
<tr>
<th>Aircraft sequence</th>
<th>Minimum Separation (in minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>medium following medium</td>
<td>40 x 1</td>
</tr>
<tr>
<td>heavy following medium</td>
<td>7 x 1</td>
</tr>
<tr>
<td>heavy following heavy</td>
<td>1 x 1.5</td>
</tr>
<tr>
<td>medium following heavy</td>
<td>6 x 2</td>
</tr>
<tr>
<td>same SID any combination</td>
<td>7 x 2</td>
</tr>
</tbody>
</table>
III. Desired Simulation Environment

To evaluate the performance of various route-planning functions with different configurations and disturbance statistics, a simulation environment was needed. The environment had to simulate the taxi process indicated by the blue area in Fig. 1. Similar to the process in Fig. 1, the simulation had to be able to generate a stream of flights including a slot, aircraft type, runway and SID. To simulate this process, the traffic simulation needed to be more than simulating the movements of aircraft. To fully simulate the traffic process, the traffic simulator had to also be able to simulate requests (startup, taxi, runway clearances) and simulate the task of the pilots to not run into other traffic. To be able to verify the robustness of a planning function, the simulation environment had to be able to insert disturbances into the aircraft movement simulation. These disturbances need to be identified and modeled correctly and will be a topic in future research. One important uncertainty in the taxi process is the variation in travel time from the gate to the runway caused by different speed profiles. To simulate this uncertainty, different speed profiles must be simulated. The first implementation of the simulation environment can help identify and model the other relevant uncertainties and disturbances. The potential improvement of the traffic flow is influenced by the margins used to compensate for these disturbances and uncertainties. To prevent the use of unnecessarily tight margins, one of the design goals of the route-planning function is to take the expected demand for movements and statistics about the disturbances into account. It is expected that this will lead to various implementations of the planning function. The simulation environment had to be able to configure the planning function and switch between different function types. To evaluate the performance of the planning function, the simulation environment had to contain a performance monitor similar to Fig. 2.

A. Taxi Operation Simulation

Figure 8 shows the main elements of the desired simulation environment. The traffic simulation generates the stream of flights including the request generation. The planning function simulates the controllers who plan the sequence of aircraft and execute that plan by providing the traffic simulation with timed instructions of startup time, taxi instructions and takeoff instructions. The traffic simulation uses the taxi routes provided by the planning function to drive the simulations of aircraft and generates similar data as obtained from the logged surveillance data. This way, the same performance monitor can be used to compare the performance parameters with the baseline. The planning functions element is where different configurations of new traffic flow optimization algorithms will be implemented. Since no human operators are involved, these simulations can be performed faster than real-time.

Figure 8. Overview of the main elements in the desired simulation environment.
The traffic simulation starts with reading a simulation script. The script contains a list of aircraft including the type of aircraft (weight class), the planned destination runway, planned SID and a slot time if applicable. The simulation script specifies when aircraft become active by generating requests. Next a planning function can use the aircraft list, possible taxi routes and incoming requests to generate taxi instructions in order to optimize the traffic flow. The planning functions element can contain different algorithms and configurations. As mentioned in paragraph II.A.1. the planning function will initially use standard taxi routes that are defined by the departure gate and used runway. All of the common possible taxi routes are a-priori generated and are available for the planning function to utilize. These taxi routes are created by a sequence of connected nodes. The location of the nodes and their connection graph is provided in the airport layout database. These connected nodes are processed into a taxi trajectory with smooth curves. This taxi trajectory is used for the actual simulation of the aircraft. The aircraft basically follows the trajectory using a discrete time step to update the position along the trajectory based on the speed of the aircraft. A database of speed profiles simulates the different speeds in between the geographical nodes of the taxi route and can differ per simulated aircraft. The simulation script can be used to create disturbances by making changes to the speed profile or taxi route. When in future research all the relevant disturbances are identified it might be necessary to extend the capability of the simulation environment to insert additional types of disturbances.

Figure 9. Simulated taxi operation.

A simulation of the taxi operation was implemented. Figure 9 gives an overview of how several aspects of the taxi operation are simulated using the implementation of the simulation environment from Fig. 8. As mentioned before the simulated operation starts with a taxi request from the simulation script. The pushback procedure is not simulated. The simulated aircraft are instructed to start taxiing (after pushback) at the same time the actual aircraft started their taxi (t1=t1s). The simulated aircraft uses a speed profile that defines average speeds between geographical nodes. Although the speed of the simulated aircraft will not be exactly the same as the actual aircraft for every point along the route, the arrival time of the simulated aircraft at the geographical points of the route will be the same as the actual aircraft (t2=t2s). As mentioned before the crossing of an active runway is not taken into...
Crossings of (in)active runways are treated as normal taxiways. In the future the simulation can be expanded by adding a runway crossing procedure. To simulate that aircraft will wait for other aircraft the positions from the movement simulation are used to detect potential collisions. If other traffic is detected within a configurable range and field of view a new speed command is sent to the movement simulation to prevent collision. Future planning functions can use the active taxi routes (including time constraints) to deconflict taxi routes a-priori and reduce the delay caused by waiting for other traffic on the taxi way as shown in Fig. 6. In the implemented simulation environment aircraft will only wait for traffic in front of them. To mimic runway behavior simulated traffic must wait for a runway clearance before entering a runway. However the amount of time an aircraft has to wait to enter a runway before taking off is an important factor in determining the throughput of a runway and needs to be simulated correctly. As mentioned before the runway throughput is limited by separation requirements that are a result of the aircraft sequence (table 1). When a simulated aircraft approaches a runway holding point, the runway clearance module checks the weight class and used SID of the previous runway user. Given the weight class and SID of the simulated aircraft table 1 is used to determine the amount of needed separation time. When the time difference between the moment the previous runway user entered the runway and the current time is smaller than the required separation time the aircraft must hold. In that case the runway clearance module in Fig. 8 will send a speed command to the movement simulation to let the aircraft stop. So, for these initial simulations only one runway entry is used and the clearance priority is based on first come first serve. With the waiting time completely moved to the area in front of the runway, it is no longer needed to simulate the actual line up and takeoff. Future planning functions can utilize the multiple runway entries to help optimize the take-off sequence. When the simulation of the multiple runway entries is added, line up and wait time on the runway will have to be added as well. The current simplifications of the simulation were chosen to make it possible to get initial results.

The data generated by the simulation is stored similar to the surveillance data and thus the same performance monitor can be utilized that was used to analyze the recorded data. This makes it possible to validate the simulation environment by comparing the performance statistics of the simulated surface operation with the performance statistics from the recorded data.

### B. Extracting Surface Traffic Scenarios

To evaluate prototype planning functions, realistic traffic scenarios are needed. Furthermore, to validate the simulation environment the simulated traffic scenario must be similar to the actual scenario that generated the logged surveillance data. The traffic scenario in the developed simulation environment is created by the simulation script, the taxi routes and the speed profiles (the green blocks in Fig. 8). To create a traffic scenario the taxi routes, speed profiles and simulation script are derived from the surveillance data, tower data and airport layout. Figure 10 shows a schematic overview of the traffic scenario extracting process.

![Figure 10. Schematic overview of the surface traffic scenario extracting process.](image)

The tower data contain flight information which is used to add a list of aircraft containing type of aircraft (weight class), the planned destination runway, planned SID and a slot time to the simulation script. As mentioned before, the simulation script instructs the simulated aircraft to start taxiing at the same time the actual aircraft started to taxi. This is based on the movement of that aircraft in the surveillance data. The ‘detect movement’ block in Fig. 10 checks the speeds of the aircraft in the surveillance data. When an aircraft moves faster than 10 knots for the first
time a start of the taxi process is detected. The time is stored in the simulation script and used in the simulation to timely start the taxi process. The threshold of 10 knots is chosen to filter out the pushback movement since the pushback procedure is not simulated. To create taxi routes the positions logged in the surveillance data are used. The block ‘detect used route’ uses the position data to collect the points in the airport layout database that were passed by an aircraft. The resulting list of points is a description of the taxi route and is used by the aircraft simulation to simulate the aircraft movement. The list of points is also used to construct a speed profile. The block ‘calculate speed profile’ calculates the average speed the aircraft taxied between two points. The results are stored in a speed profile that is used by the movement simulation to simulate the speed between two points of a taxi route. This method limits the resolution of the speed profile to the resolution of the taxi route and therefore the resolution of the airport layout database. But the airport layout database is created in such a way that with the minimal set of data points every taxiway crossing has a data point. This results in a speed profile resolution that ensures a correct arrival time at each taxiway crossing. Since ultimately the goal of the SANI research project is to plan the taxi flow on a taxiway crossing resolution the described method of creating and using a speed profile seems sufficient.

C. Simulation Validation

To validate the simulation environment the simulation data and actual surveillance data were analyzed and compared using the performance monitor. As mentioned in II.A.2 currently the optimization process takes place before the aircraft starts to taxi and at the runway. Instead of modeling the optimization strategy the controllers use before the aircraft starts to taxi, the simulation script is used to mimic the behavior. The start taxi time from the simulation script is used to time the start of the aircraft movement simulation and ensures that the simulated aircraft starts to move at the same time the actual aircraft started to move. This mimics the result of the planning behavior of the controller. At the runway the exact minimal runway separation is simulated which results in an optimal runway operation for a given aircraft sequence. Disturbances that increase the time between the minimal separation and the actual takeoff are not yet taken into account and could result in an increased performance outcome versus the actual surveillance data. Also the optimization of the sequence using multiple runway entries is not yet simulated and this could cause differences in resulting takeoff sequence and reduced performance outcome. In a future implementation of the simulation environment the runway model will be extended.

Since the actual line-up and takeoff of the aircraft are not simulated the ‘detect takeoff’ block in the performance monitor was modified for analyzing the simulated data. For the simulation analysis a takeoff is detected when an aircraft entered the runway. As a consequence there is no waiting time on the runway but an increased waiting time at the runway hold line. It is expected that this has no effect on the performance outcome.

With the taxi movements starting at the same time, the speed profiles being matched to the average speed, and the traffic being handled as a first come first serve sequence, it is expected that the simulation results will be similar to the results of the actual data. This will be the case, unless there are not-simulated disturbances that have significant impact.

IV. Simulation Results

The implemented simulation environment was tested using the scenarios extracted from the surveillance data. The results are plotted together with the results from the analysis of the surveillance data. Figure 11 shows the histogram displaying what percentage of aircraft had a certain amount of total delay waiting in the queue area to takeoff from runway 36L. The results from the MLT data are shown in blue and are the same as in Fig. 5. The results from the simulation are in red. Figure 11 shows that the simulated taxi operation results in less delay in the runway queue area than was observed in the MLT data. Almost 65% of the aircraft had a delay of 60 seconds or less in the simulation versus almost 50% of the aircraft in the MLT data. Furthermore aircraft had a maximum delay of 240 seconds in the simulated environment whereas the MLT data show delays up to 360 seconds.

Figure 11. Amount of time aircraft wait at runway 36L.
Figure 12 shows the throughput of runway 36L expressed in the amount of aircraft per hour over the simulation period. The blue line is the result from the MLT data and is the same as in Fig. 7. The red line is the result from the simulation. The simulated and actual throughput follow the same trend till 5000 seconds into the simulation. At that point in time the simulated throughput drops to around 20 aircraft per hour whereas the actual throughput remains around 35 aircraft per hour. This difference might be caused by a difference in takeoff sequence as a result of the optimization of the sequence using multiple runway entries.

Table 3 shows the minimal takeoff separation that was required as a result of the takeoff sequence and is compared to the results from the MLT data as shown in table 2. The difference in sequence shows that especially less aircraft with the same SID followed each other in the MLT data. As a result the minimal separation required based on the sequence is 6 minutes less than the simulated sequence.

Since in the implemented simulation environment aircraft only wait for other aircraft that are directly in front of them, there is no additional delay caused by waiting on the taxiways outside the runway queue area. Just as in the MLT data none of the simulated aircraft that took off from 36L missed their CTOT.

As described in chapter IV there are some differences in the performance data between the simulated data and the data from the surveillance data. A possible explanation for the difference in the amount of waiting time in the queue area is the earlier mentioned possible difference between the required minimal separation at takeoff and the actual separation. To investigate this possible explanation the actual separation at takeoff was calculated. Figure 13 shows a plot of the actual separation at takeoff from the MLT data in blue and from the simulated data in red. The implemented simulation environment simulates the exact required minimal separation and therefore the actual separation times are close to the times specified in table 1. (60, 90 or 120 seconds). The variation around these values is caused by the aircraft movement simulation. The values in between indicate that for a certain amount of time there were no aircraft at the runway. The results from the surveillance data show that the actual separation time is higher than the required minimal separation. For example the most common minimal required separation time is 60 seconds, but Fig. 13 shows a separation distribution for the actual data around 70-80 seconds. A possible explanation is the uncertainty between the time a controller issues a takeoff clearance and the time the aircraft actually takes off. To ensure that the minimal separation requirement is always met the controller probably adopts a strategy to time the takeoff clearance in such a way that even for the shortest amount of time between clearance and actual takeoff, the minimal separation requirement is met. As a result the average of the actual takeoff distribution for traffic that requires a minimum of 60 second separation time is in this case moved to around 70 seconds.
The possibility to change the takeoff sequence by using multiple runway entries was not simulated. This resulted in a difference in takeoff sequence as shown in table 3. Although the takeoff sequence from the MLT data reduces the minimal required separation by 6 minutes the data showed that some aircraft had to wait for several minutes in order to change the sequence. This could explain the longer waiting times in Fig. 11.

To further analyze the factors that play a role in the takeoff procedure and taxi operation future research will use the simulation environment to analyze bigger datasets. The results will be used to expand the simulation environment with models of the found behavior.

VI. Summary & Conclusion

By observing the current taxi process the traffic flow performance parameters and surface operation constraints at the Amsterdam airport were identified. A new method was described to obtain and quantify taxi traffic flow performance and taxi speed distribution using real world data from Amsterdam airport. The uncertainty in taxi times is an important factor that prevents the optimal use of the available taxiway and runway infrastructure. The obtained speed profile distribution can be used in the development of traffic flow functions.

Furthermore a simulation environment was developed to evaluate the performance of various future route-planning functions for selectable time-margins and disturbance statistics. To validate the simulation environment the outcome of the performance monitor was compared to the outcome of the analysis of the actual surveillance data. Instead of modeling the optimization strategy the controllers use a simulation script was used and the runway behavior was simulated using the minimal separation requirements. Furthermore the obtained speed distribution data were used to minimize the uncertainty in taxi time. In addition a method was described to extract surface traffic scenarios from real world data. The resulting scenarios can be used to evaluate future planning functions and one scenario was used to verify the fidelity of the surface traffic simulation.

The simulation results show some differences in the performance data between the simulated data and the data from the surveillance data. The simulation of the takeoff clearance behavior and the runway sequence behavior are identified as possible causes. The actual data showed a more optimal sequence that required 6 minutes less separation time, but the simulated data showed shorter overall wait times. More simulation and analysis is needed to answer the questions that are raised by these results. To further analyze the factors that play a role in the takeoff procedure and taxi operation future research will use the simulation environment to analyze bigger datasets. The results will be used to expand the simulation environment with models of the found behavior.

References


American Institute of Aeronautics and Astronautics