Assessing Tactical Scheduling Options for Time-Based Surface Metering

Shannon Zelinski and Robert Windhorst
NASA Ames Research Center
Moffett Field, California, USA
shannon.j.zelinski@nasa.gov, robert.d.windhorst@nasa.gov

Abstract—**This paper presents a parametric analysis of the most recent tactical scheduler design for NASA's Airspace Technology Demonstration 2 (ATD-2) sub-project, committed to demonstrating time-based surface metering at Charlotte Douglas International Airport (CLT). The tactical scheduler design is implemented in a fast-time simulation model of CLT using NASA's Surface Operations Simulator and Scheduler. The tactical scheduler is supported by three basic functions: trajectory prediction, runway scheduling, and advisory generation. A key parameter of the advisory generation function is the taxi time delay buffer used when calculating target gate pushback times from runway schedule. Multiple simulations that varied the amount of taxi time delay buffer were analyzed to determine the effect on tactical scheduler performance. The results show an improvement in tactical scheduler performance when the buffer is made sufficiently large to release departures from their gates early enough to maintain scheduler predicted runway throughput.**

Keywords-surface operations; surface metering; tactical scheduling

I. Introduction

NASA is committed to demonstrating a concept of integrated arrival, departure, and surface operations by 2020 under the Airspace Technology Demonstration 2 (ATD-2) sub-project [1]. This will be accomplished in three phases, starting with a demonstration of flight specific time-based surface metering at Charlotte Douglass International Airport (CLT) [2]. ATD-2 tactical metering capability is based on NASA's Spot And Runway Departure Advisor (SARDA) which has been tested successfully in human-in-the-loop simulations of CLT [3]. SARDA makes use of surface surveillance data and surface modeling to estimate the earliest takeoff time for each flight active on the airport surface or ready for pushback from the gate. The system then schedules each flight to its assigned runway and assigns a target pushback time displayed to ramp controllers as an advisory gate hold time. The objective of this method of surface metering is to move as much delay as possible to the gate to minimize surface congestion and engine on-time while keeping sufficient pressure on the runway to maintain throughput. This flight specific approach enables greater flight efficiency and predictability, facilitating trajectory-based operations, which ATD-2 aims to achieve.

Throughout ATD-2 project formulation and system development, researchers have continuously engaged with stakeholders and future users, uncovering key system requirements for tactical metering that SARDA did not address. These include:

- Pushback time advisories do not change after the pilot calls in ready for pushback so that controllers may communicate a single gate hold to the pilot at this time.
- Make use of and incentivize improved accuracy of air carrier-provided estimates of pushback ready time.
- Facilitate integration with strategic scheduling consistent with the Surface Collaborative Decision Making Concept of Operations [4].

This paper presents a benchmark fast-time simulation evaluation of the tactical scheduler designed to meet these new metering requirements. The scheduler is used to calculate gate holds for flights. Flight operations with the gate holds are simulated using a fast-time medium-fidelity simulation of CLT. A key parameter used in the generation of gate holds, the taxi time delay buffer, is analyzed to determine the value that moves the most delay from the runway queue to the gate without drying out runway operations. Fast-time simulation is used to rapidly develop and test new scheduler design features and future fast-time simulation studies will build upon this benchmark evaluation to explore other design considerations.

The rest of this paper is organized as follows. Section II presents a background of surface metering research. Then the ATD-2 tactical scheduler used for surface metering is presented in section III. Simulation details and results from a benchmark evaluation of the ATD-2 tactical scheduler are presented in sections IV and V, respectively. Finally, conclusions and future work are presented in section VI.

II. Background

Prior demonstrations of automation aided surface metering in the U.S. have focused on strategic approaches to enable Collaborative Decision Making (CDM) between flight operators, Air Navigation Service Providers, and other stakeholders in surface and departure operations as defined in the Surface-CDM Concept of Operations [4]. Both the Collaborative Departure Queue Management (CDQM) approach evaluated at Memphis International Airport [5] and the Ground Management Program

(GMP) at John F. Kennedy International Airport [6,7] used ration by schedule algorithms to allocate Active Movement Area (AMA) entry slots to different carriers to manage the length of departure runway queues below a threshold value. The methods differed in their mechanism to enable carrier flexibility. CDQM abstracts the individual flight based slots by allocating to carriers a specific number of aircraft for each that may enter the AMA within 10-minute intervals. The carrier then has the flexibility to decide which of its flights to fill each allocated slot. On the other hand, the GMP assigns each slot to a specific aircraft. Air carriers are still given the flexibility to swap flights according to their priorities and manage their own ramp areas, but this flight specific method also allows the system to monitor compliance with AMA entry times to ensure that everyone is treated fairly.

The most tactical form of surface metering field demonstrated in the U.S. is a test of pushback rate control at Boston Logan International Airport [8,9]. This method is not flight specific, suggesting the number of aircraft that should be allowed to push back in the next 15 minutes to maintain a count of active aircraft predetermined to place sufficient pressure on the runways without over-congesting the surface. Flights are released on a first-come-first-served basis as they call in ready for pushback.

Departure MANager (DMAN) [10] is a time-based tactical surface metering system operational in Europe. A demonstration of DMAN implemented at Athens Airport provided published results [11]. DMAN calculates a runway time and corresponding start-up (gate pushback in U.S.) time for each flight to maintain departure queues of two or three aircraft at the runways. Conceptually, DMAN is intended to collaborate with surface and arrival management counterparts (AMAN and SMAN) to establish arrival and departure sequences and optimize surface movement plans [12]. The trials at Athens Airport implemented DMAN alone, using flight clearances as events to update calculations. Thus, as flights were issued start-up clearances, the start-up advisories for all flights still at the gate were updated.

SARDA also employs time-based tactical surface metering, but has thus far been demonstrated in human-in-the-loop simulation [3], not in the field. Whereas DMAN scheduler updates in the Athens demonstration were event based, the SARDA scheduler is updated every 10 seconds using real-time surface surveillance data. These updates ensure the most accurate information is used to predict runway usage. However, rapid updates also open the potential for fluctuating advisories, which Ramp controllers at a busy airport like CLT find challenging. Therefore, a new requirement for ATD-2 tactical metering is that all advisories must freeze as soon as the pilot calls in ready for pushback so that Ramp controllers may communicate a single hold time when responding to pilot ready calls.

ATD-2 aims to bridge the gap between strategic CDM and tactical metering by using air carrier provided estimates of pushback or start-up time called Earliest Off Block Time (EOBT). SARDA developers found that historically available EOBT updates based on air carrier “L-times” were too inaccurate for tactical metering. Therefore, SARDA used EOBTs only to define the group of flights within the tactical planning horizon (within 10 to 15 minutes of EOBT). Flights within the tactical time horizon that had not yet called in ready were scheduled opportunistically so that hold advisories would be available for each in case it was the next to become ready. In anticipation of improved EOBT accuracy from American Airlines, the hub operator at CLT, the ATD-2 tactical scheduler will schedule flights within the tactical time horizon that have not yet called in ready for pushback more realistically, and incorporate S-CDM principles into the scheduling prioritization scheme to prevent EOBT gaming and stabilize tactical scheduler advisories in the presence of updating EOBTs.

III. TACTICAL SCHEDULER

The tactical scheduler is supported by three basic functions: trajectory prediction, runway scheduling, and advisory generation. These functions may be performed differently according to the runway usage prediction accuracy of flights in different states of surface operation. The most basic set of flight groups in order of descending runway usage predictability are

- Landing Arrival - arrivals predicted to land on the scheduled runway or any other runway that imposes spacing constraints on the scheduled runway.
- Taxi Arrival - arrivals predicted to taxi across the scheduled runway.
- Taxi Departure - departures actively taxiing on the airport surface (already pushed back from the gate) predicted to take off from the scheduled runway.
- Gate Departure Ready - departures for which the pilot has called in ready for pushback but the flight is being held at the gate.
- Gate Departure Planned - departures at the gate with EOBT within the tactical scheduling horizon but for which the pilot has not yet called in.
- Gate Departure Uncertain - departures at the gate with EOBT outside the tactical scheduling horizon and for which the pilot has not yet called in ready.

Other groups may be included to distinguish flights with emergency or exempt status or departures affected by traffic management initiatives, but these cases are not analyzed in this paper.

Flights may jump from one group to another between scheduler calls as their state of surface operation changes.

A. Trajectory Prediction

Trajectory prediction uses flight state and intent information to generate an Earliest Runway Usage Time (ERUT) for all flights predicted to use each scheduled runway. Four types of runway use are predicted: arrivals landing on the scheduled runway, arrivals landing on other runways which impose spacing constraints on the scheduled runway (e.g., intersecting or converging runways), arrivals taxiing across the scheduled runway on the way to their arrival gates, and departures assigned to takeoff from the scheduled runway.

Trajectory prediction calculates a flight's ERUT differently depending on its flight group. Table I summarizes how ERUTs are calculated for each flight group. These calculations are explained in more detail below.

In the field, Time-Based Flow Management (TBFM) [13], the currently operational tool responsible for arrival metering, will calculate Scheduled Times of Arrival (STAs) for arrivals and send them to the tactical scheduler. Therefore, the Landing Arrival group, consisting of the first two runway usage types, uses landing STAs as ERUTs.

For all other groups, nominal taxi paths between gates and runways and nominal speeds for various aircraft types in different areas of the airport surface are used to calculate an Unimpeded Taxi Time (UTT) between current position and the scheduled runway for each flight. For the Taxi Arrival and Taxi Departure groups, trajectory prediction adds UTT to current time to get each flight's ERUT. If a Taxi Arrival has not yet landed, UTT is added to its landing STA for its landing runway to get the ERUT for its taxi crossing runway.

The advisory time at which to release a flight for pushback, known as the Target Off Block Time (TOBT), is frozen once a flight calls in ready. Therefore, trajectory prediction for Gate Departure Ready flights adds UTT to the flight's frozen TOBT to get ERUT.

For Gate Departure Planned group flights, trajectory prediction adds UTT to the EOBT to get ERUT. Because Gate Departure Uncertain group EOBTs are not expected to be very accurate, trajectory prediction adds UTT to current time to get ERUT just in case one of these flights calls in ready and jumps straight to the Gate Departure Ready group.

TABLE I. FLIGHT GROUP TRAJECTORY PREDICTION

Flight Group	Position	ERUT Calculation
Landing Arrival	airborne	Landing STA
Taxi Arrival	airborne	Landing STA + UTT
	surface	Current time + UTT
Taxi Departure	surface	Current time + UTT
Gate Departure Ready	gate	TOBT + UTT
Gate Departure Planned	gate	EOBT + UTT
Gate Departure Uncertain	gate	Current time + UTT

B. *Runway Scheduling*

All flights predicted to use the runway are scheduled one at a time in an order that depends on their flight group. Flight groups are scheduled in priority order of decreasing runway usage predictability listed in the previous section. All flights within a single flight group are scheduled before moving on to the next flight group. Within each flight group, flights are scheduled in the order of their ERUTs. The only exception is the Gate Departure Planned flight group, which is ordered by the airline posted Scheduled Off Block Time (SOBT) in compliance with S-CDM ration-by-schedule convention. Using SOBT rather than EOBT to order the flights for scheduling prevents EOBT gaming and stabilizes tactical scheduler advisories in the presence of updating EOBTs.

Time-based spacing requirements between each leader-follower pair of runway use types and aircraft weight categories are adapted to a particular airport and used as constraints for runway scheduling. Only wake separation requirements between departures on the same runway depend on aircraft weight categories. Distance-based rules were converted to time-based separations between departures and arrivals on the same and interdependent runways, and flights taxiing across runways.

ATD-2 also considers other constraints from traffic management initiatives such as Miles-in-Trail (MIT), Expect Departure Clearance Time (EDCT), and Approval Request/Call for Release (APREQ/CFR). However, these are not included in this study.

Within a single scheduling call, each scheduled flight creates spacing constraints for all flights scheduled after it. If a flight's ERUT does not meet all constraints, the flight is delayed and rechecked for constraint violations. The first time meeting all constraints is assigned to the flight as its Target Runway Usage Time (TRUT) and is used to define spacing constraints for subsequently scheduled flights. The Gate Departure Uncertain group is the only group that is scheduled opportunistically, meaning that these flights are inserted into runway slots when available and do not create constraints for subsequent flights when scheduled.

C. *Advisory Generation*

When all flights have been scheduled and assigned a TRUT, TOBTs are calculated for all departures still at the gate by subtracting the gate-to-runway UTT times a constant, *A*, and delay buffer, *B*, from the TRUT. TOBT is calculated as

$$\text{TOBT} = \text{TRUT} - A(\text{UTT}) - B, \tag{1}$$

where $A \geq 1$ and $B \geq 0$. Factor *A* accounts for congestion delay the flight may encounter along it's taxi route and assumes this delay is greater the more unimpeded time the flight has left to travel. Factor *B* adds queue delay where flights line up from the end of the runway.

The resulting TOBTs are presented to ramp controllers as advisory hold times for metering.

D. *Scheduler Dynamics*

The tactical scheduler is called every 10 seconds updating TRUTs and TOBTs. Flights move from the Gate Departure Uncertain group to the Gate Departure Planned group when their EOBTs move within the tactical planning horizon.

Flights that become ready between scheduler calls are moved from the Gate Departure Uncertain or Gate Departure Planned group to the Gate Departure Ready group and their TOBTs are frozen at the last TOBT update given when they were in their previous group. This ensures that a static hold advisory is available for a flight as soon as the pilot calls in ready.

Because the Gate Departure Planned group flights create constraints for subsequently schedule flights, they can't jump in front of each other by calling in earlier than their EOBTs. Because the Gate Departure Uncertain group flight do not create constraints, these flights can jump in front of each other by calling in early. But they can't jump in front of Gate Departure Planned flights (because those flights are scheduled earlier and create constraints) unless there is a sufficient gap in the schedule due to low demand.

IV. SIMULATION

Fast-time simulations were conducted to evaluate tactical metering performance for varying scheduler parameters.

A. Simulation Environment

The Surface Operations Simulator and Scheduler (SOSS) [14] is used to simulate surface operations at CLT. SOSS simulates both departure and arrival aircraft movements within a network of nodes and links representing the airport surface. Fig. 1 shows a map view of the SOSS CLT surface model identifying the different types of nodes and links. In South flow, runways 18R and 23 are used only for arrivals. Runways 18C and 18L are used predominantly for departures with only occasional use for arrivals to offload 18R.

Departure routes begin at a gate node, transition through ramp, spot, taxiway, and departure queue nodes, and takeoff from departure nodes. Arrival routes begin at an arrival node, transition through taxiways, a spot, and ramp nodes, ending at a gate. Some arrivals may need to taxi across a runway via crossing nodes, such as 18R arrivals crossing 18C. If a gate conflict between and arrival and departure is predicted, the arrival will be diverted and held at a hardstand node until the departure vacates the gate [15]. For the purposes of this study, only arrivals are sent to hardstands to avoid gate conflict induced grid-lock without impacting departure scheduling.

Arrival landing STAs and subsequent entry into the surface model are predetermined by the traffic scenario. Whereas departure SOBTs are predetermined by the traffic scenario, actual gate pushback may be controlled by scheduler TOBT assignments. Departures with assigned TOBTs are held at the gate until current time is greater than or equal to TOBT. Otherwise, departures are released as soon as they are ready. For this evaluation, EOBTs remain static and no ready time uncertainty was modelled. Therefore, all flight ready times are equal to their EOBTs which are equal to their SOBTs as specified in the traffic scenario.

Aircraft are allowed to progress along predefined routes through the node-link network as long as they do not violate separation constraints specific to each part of the network. To avoid violating separation constraints, flights may stop at taxiway intersections and form queue lines at departure runways, resulting in taxi-time uncertainties due to surface congestion. Although SOSS has the capability of modeling uncertainty in various areas of the airport surface, the only uncertainties simulated were surface congestion related taxi-time uncertainties. This minimum level of uncertainty modeling was desired to perform a benchmark fast-time simulation evaluation of the ATD-2 tactical scheduler.

The simulation time step was set to 0.5 sec and the scheduler was called every 10 seconds.

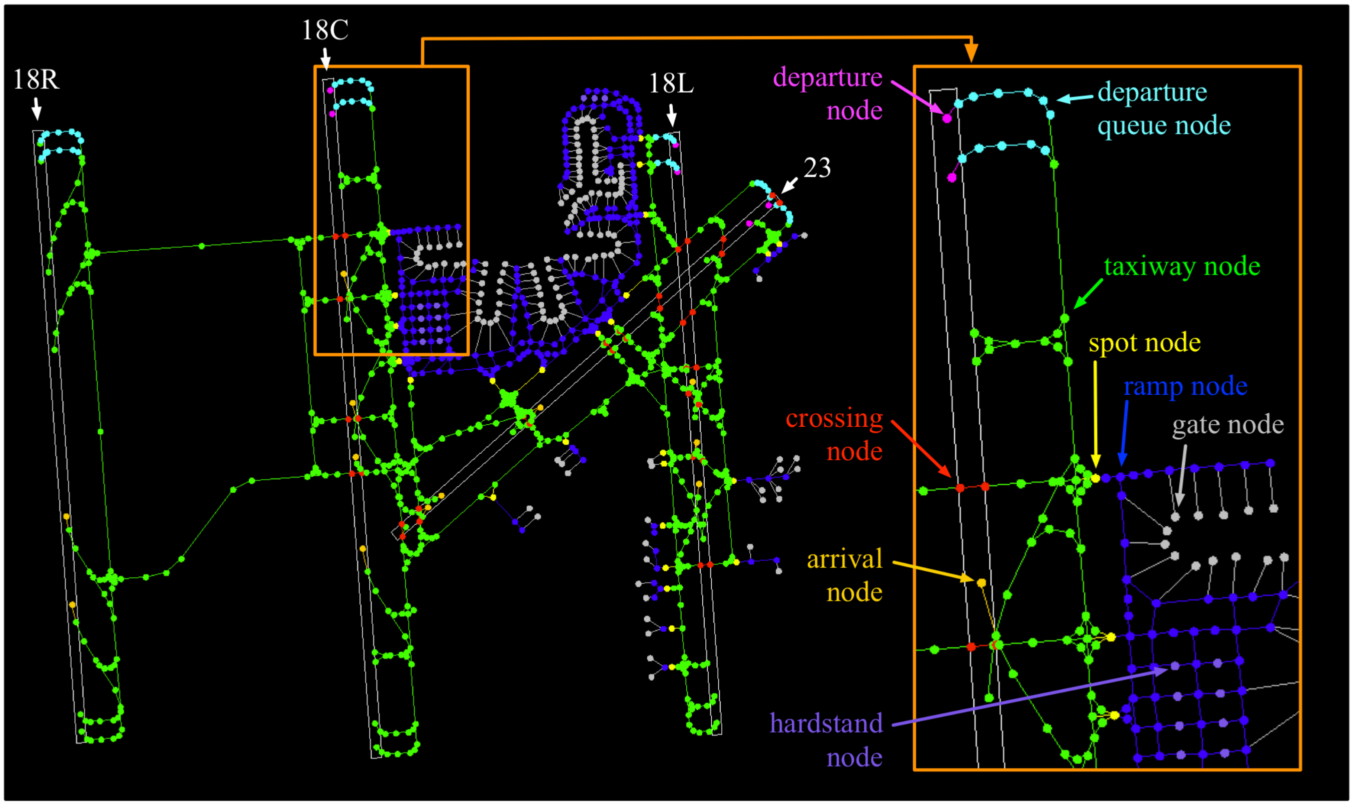

Figure 1. CLT Surface Model

B. *Traffic Scenario*

As a hub airport for American Airlines, CLT traffic is characterized by tightly spaced departure and arrival banks. A four-hour traffic scenario was generated using CLT surface surveillance data from March 11, 2016, covering two such banks when CLT was operating in South flow. Fig. 2 shows the arrival and departure demand on each runway in 15-minute bins.

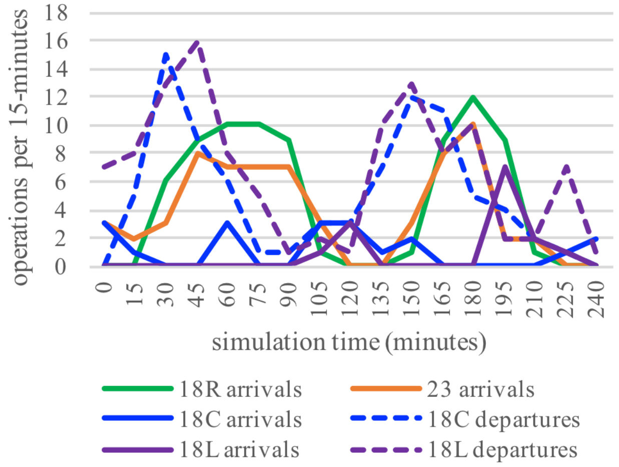

Figure 2. Runway Demand

C. *Scheduler Parameters*

Scheduler parameters include the planning horizon and taxi time delay buffer parameters. The start of the planning horizon was set to 10 minutes prior to EOBT.

Delay buffer parameters include values A and B from (1). Value A was set to 1.05 to account for a minor amount (5%) of congestion related delay. Value B is the key parameter which determines how much delay may be absorbed on the surface, passing any excess delay to the gate. To analyze how the amount of delay pushed back to the gate affects scheduler performance, multiple simulations were completed, each with a different B value. Value B was increased from 0 in 1-minute intervals until the simulation resulted in no delay passed to the gate as would be the case with no surface metering.

D. *Evaluation Metrics*

The goal of surface metering is to minimize surface delay and move as much delay as possible from active taxi in the ramp of AMA to the gate with engines off, without negatively impacting runway throughput. To that end, surface metering tries to regulate the flow of departures to the runway to meet the predicted runway departure rate and maintain a queue of flights at the end of the runway just long enough to account for uncertainty. The following delay, runway usage time, throughput, and departure queue metrics are designed to evaluate the performance of time-based surface metering utilizing tactical scheduling in the presence of taxi time uncertainties.

1) *Delay*

Total departure delay d_{dep} is measured as the difference between a departure flight's ERUT and Actual Runway Usage Time (ARUT). Surface metering attempts to move a portion of the delay from taxiing (in the ramp or AMA) to the gate. Thus, total delay is segregated into gate delay d_{gate}, ramp delay d_{ramp}, and AMA delay d_{AMA}. Gate delay is measured as the difference between a flight's EOBT and Actual Off Block Time (AOBT). The transition from ramp to AMA occurs at the Movement Area entry Time (MAT). Ramp delay is obtained by subtracting gate delay from the difference between a flight's earliest and actual MAT (AMAT – EMAT). AMA delay is then obtained by subtracting gate and ramp delay from total departure delay.

$$d_{\text{dep}} = \text{ARUT} - \text{ERUT} \tag{2}$$

$$d_{\text{gate}} = \text{AOBT} - \text{EOBT} \tag{3}$$

$$d_{\text{ramp}} = \text{AMAT} - \text{EMAT} - d_{\text{gate}} \tag{4}$$

$$d_{\text{AMA}} = d_{\text{dep}} - d_{\text{gate}} - d_{\text{ramp}} \tag{5}$$

Total arrival delay d_{arr} is the difference between an arrival flight's Earliest Gate Arrival Time (EGAT) and Actual Gate Arrival Time (AGAT), where EGAT is calculated by adding UTT between runway and gate to arrival STA. In this paper, because no arrival time uncertainty was modeled, arrival STA is the same as Actual Time of Arrival (ATA). Total arrival delay is segregated into AMA delay d_{AMA} and ramp delay d_{ramp}. The transition from AMA to ramp occurs at the Ramp Entry Time (RET). AMA delay for arrivals is the difference between a flight's earliest and actual RET (ARET – ERET). Ramp delay for arrivals is then obtained by subtracting AMA delay from total arrival delay.

$$\text{EGAT} = \text{STA} + \text{UTT} \tag{6}$$

$$d_{\text{arr}} = \text{AGAT - EGAT} \tag{7}$$

$$d_{\text{AMA}} = \text{ARET - ERET} \tag{8}$$

$$d_{\text{ramp}} = d_{\text{arr}} - d_{\text{AMA}} \tag{9}$$

2) *Ruway Usage Time Prediction*

Because TOBT is frozen at flight ready time, surface metering is dependent on runway scheduler predictions of TRUT made at flight ready time. Let a flight's predicted runway usage time error be

$$e = \text{ARUT - TRUT}(t_{\text{ready}}), \tag{10}$$

where t_{ready} is the flight ready time.

3) *Throughput Prediction*

Due to taxi time uncertainties and lack of control beyond gate push back, the tactical scheduler is unlikely to predict the exact sequence of flight operations on the runway, which will negatively impact e. Tactical scheduler performance is evaluated in a less sequence sensitive way by calculating runway departure throughput prediction errors. Let t be current time and τ be a future time for which departure throughput for a given runway is predicted. Let runway departure throughput rate $R(\tau,t)$ be the number of departures scheduled to use the runway between time τ-15 minutes and τ, calculated at time t. $R(t,t)$ is the actual departure rate calculated at time t, when all flights included in the rate have already departed. $R(t + 15,t)$ is the earliest rate that is based entirely on predictions or flights that have not yet departed. Let prediction error be

$$E(\tau,t) = R(\tau,t) - R(t,t). \qquad (11)$$

At CLT, nominal taxi times can take up to 25 minutes. Allowing for some gate hold time, prediction errors for τ up to ~35 minutes greater than t may affect surface metering performance. The most relevant throughput predictions to surface metering are analyzed by computing $E(\mathrm{ARUT}, t_{\mathrm{ready}})$ for each flight.

4) *Departure Queue*

Another metric for tactical scheduler evaluation is departure queue size. Queues are desired to be shorter to minimize congestion related delay. However, they should be long enough to account for taxi time uncertainty and keep the runways from going dry. Four queue values per runway are used: ramp count q_{ramp}, AMA count q_{AMA}, taxi count q_{taxi}, and the queue line q_{line}. The first three queues are simple counts of departures taxiing in the ramp, AMA, or either (ramp or AMA) bound for each runway. The queue line is the number of aircraft lined up from the end of the runway waiting to depart. This is calculated in post processing by identifying the uninterrupted cascading chain of departures scheduled to use the runway that are within close proximity (200 meters) of one another extending out from the runway entrance node.

The amount of delay buffer used to calculate TOBTs from ERUTs directly influences the average queue size. Good departure throughput predictability should enable surface metering to provide more queue stability, which should enable smaller queues to maintain maximum runway utilization.

V. Results

Simulations of increasing B from (1) were performed until d_{gate} was zero for all departures at a B value of 14 minutes. At a B value of 10, d_{gate} was zero for all departures using runway 18C. Therefore, results focus on B values between 0 and 10.

A. *Delay*

Fig. 3 shows a stacked plot of average departure d_{gate}, d_{ramp}, and d_{AMA} for each simulation as B was increased. Total departure delay d_{dep} can be viewed as the sum of the stacked bars. As expected, d_{gate} decreases and active taxi delay ($d_{\mathrm{ramp}} + d_{\mathrm{AMA}}$) increases as B is increased. This verifies that B is an effective parameter to control the amount of delay moved from active taxi to the gate. Compared to d_{AMA}, d_{ramp} changes very little. This means that most if not all of the delay the lower B values transfer to the gate is queue delay (time waiting in line in the taxiways to use the runway). Note that d_{dep} is noticeably greater when $B < 2$. When B is too low, the d_{gate} is increased more than d_{AMA} can be decreased due to starving the runway.

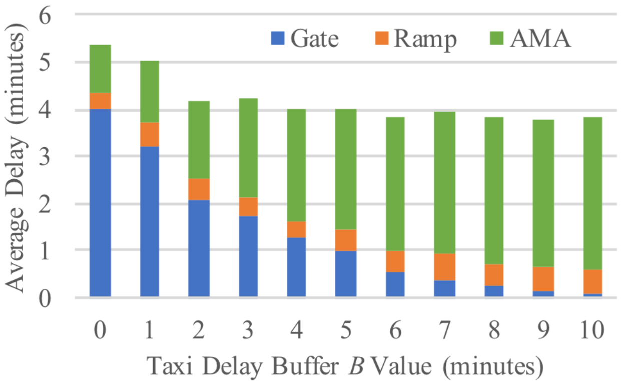

Figure 3. Gate, Ramp, and AMA Departure Delay

Fig. 4 shows a stacked plot of average arrival d_{AMA} and d_{ramp} for each simulation as B was increased. Total arrival delay d_{arr} can be viewed as the sum of the stacked bars. Although average d_{arr} is consistently much larger than d_{dep}, it did not appear to be sensitive to departure taxi time delay buffers. Although extra arrival delay is expected due to additional gate conflicts as arrivals wait longer for departures held at their gates, much of this delay was absorbed as extra taxi path distance for arrivals using hardstands, which is not captured in the presented arrival delay calculations. Whereas departures absorbed most of their active taxi delay in the AMA, arrivals absorbed most of their delay in the ramp. In general, the arrival taxi-in routes are much longer than departure taxi-our routes. Also, the areas where arrivals tended to stack up waiting for use of common taxi route segments happened to be in the ramp area.

B. *Runway Usage Time Prediction Error*

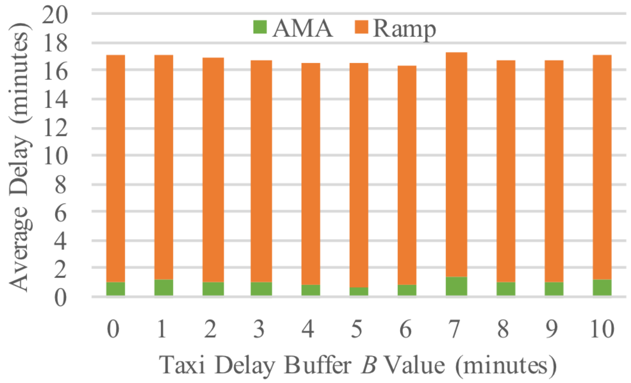

Figure 4. AMA and Ramp Arrival Delay

Fig. 5 shows the average and standard deviation of runway usage time prediction error, e, across all departures at their ready times for each simulation of varying B value. On average, departures use the runway later than predicted as indicated by the positive average e for each simulation. This bias may be because ERUTs used to derive TRUTs are based on unimpeded transit time. This bias could be reduced by subtracting some expected congestion delay from the ERUT calculations shown in Table I.

The average e increases with B and begins to level off as B exceeds 6 minutes. Despite releasing departures earlier from the gates (higher B values), these departures encounter so much additional congestion delay as to cause their actual use of the runway to be even later. Standard deviation of e also increases with B. This indicates that it is more difficult to predict the departure sequence in the presence of additional congestion caused by releasing departures from their gates earlier.

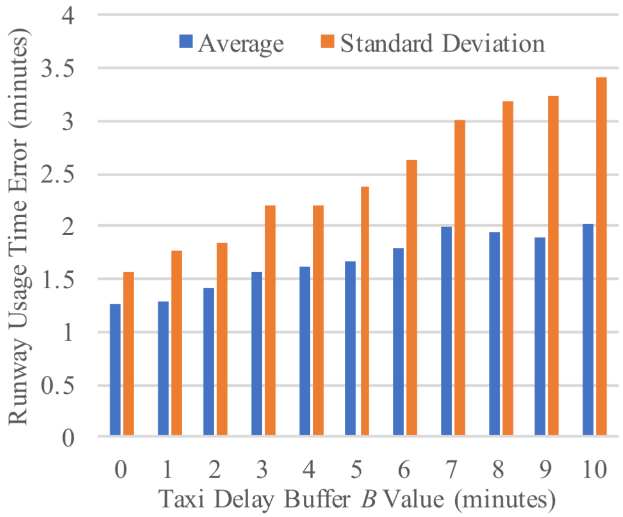

Figure 5. Runway Usage Time Prediction Error

C. *Throughput Prediction Error*

Fig. 6 shows the average and standard deviation error in throughput prediction made at each departure's ready time $E(\text{ARUT}, t_{\text{ready}})$, referred to as simply E below. For $B < 2$, runway throughput is over predicted (E is negative) on average. For these simulations, departures were not released from the gate early enough to provide sufficient pressure on the runways allowing instances of excess separation that decreased actual throughput. Like e, the average E increases with B and begins to level off as B exceeds 6 minutes. Unlike e, the standard deviation of E is similar for all values of B as this metric is less sensitive to departure sequence.

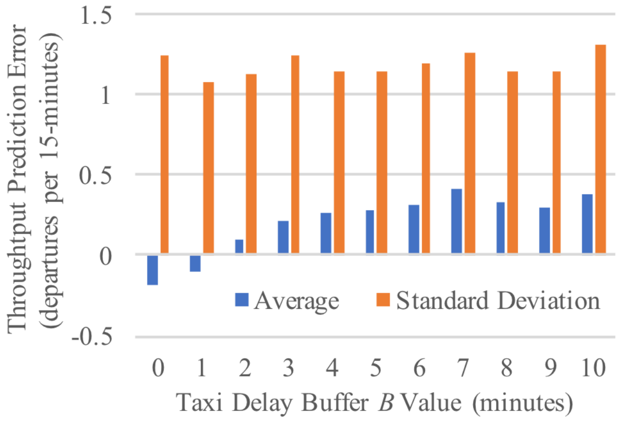

Figure 6. Throughput Prediction Error

D. *Departure Queue*

Departure queues were analyzed by calculating the maximum and average values separately for each runway and two hours of simulation time capturing a single demand cycle. Results are presented for only the first two hours of simulation time encompassing the larger of the two demand cycles as can be seen in Fig. 2. Figs. 7 and 8 show maximum and average queue sizes for 18L and 18C, respectively.

For both runways, the q_{AMA} results are similar to the q_{line} results. The average q_{AMA} is always slightly higher than the average q_{line} because most of the line is contained within the AMA. At $B = 10$, maximum q_{line} exceeds maximum q_{AMA} for 18L when the line extends out into the ramp. Fig. 9 shows how the last departure counted as part of q_{line} is in the ramp (blue nodes and links).

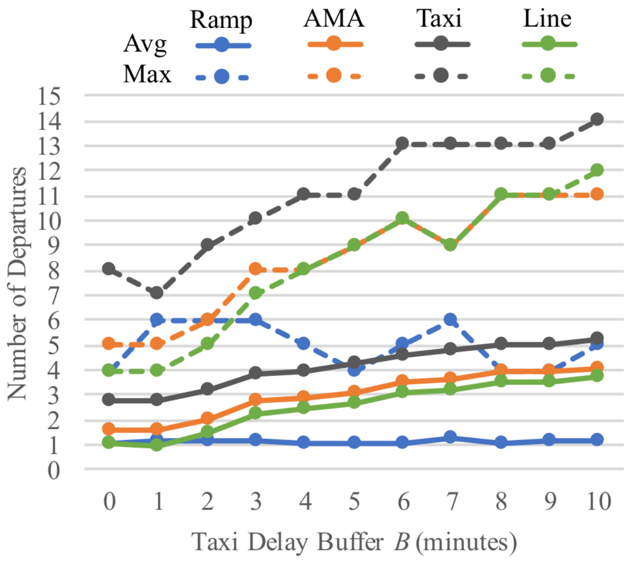

Figure 7. 18L Departure Queues

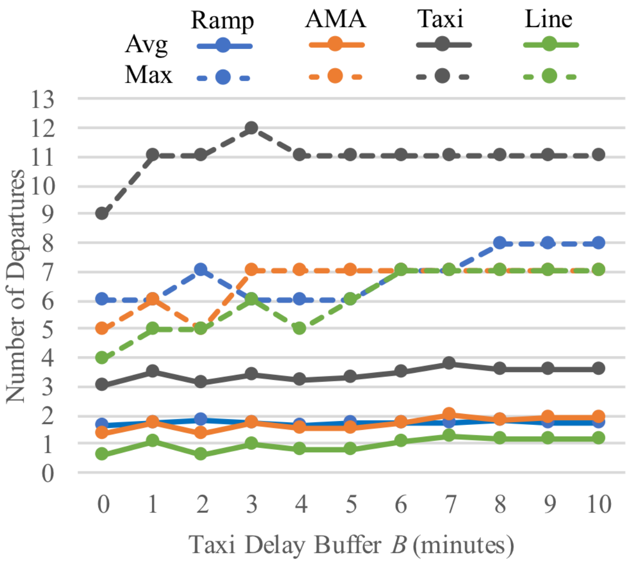

Figure 8. 18C Departure Queues

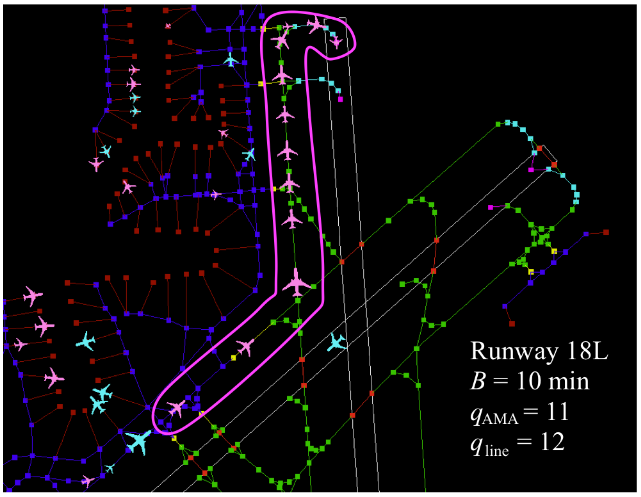

Figure 9. 18L Maximum Queue Line

For 18L, the average and maximum q_{AMA} and q_{line} increase with B, whereas the q_{ramp} metrics do not. This is probably because there is very little room in CLT's ramp area near 18L, and so the q_{ramp} saturates quickly. The fluctuations in maximum q_{ramp} between simulations are due to subtle differences in the traffic jams encountered in each simulation, but the ramp area near 18L appears to saturate at q_{ramp} = 6 departures. The max q_{taxi} shows signs of saturation for $B > 5$ when it levels off to 13 departures. It is interesting to note that the maximum q_{ramp} jumps higher and the q_{AMA} dips lower for $B = 7$, suggesting that a ramp traffic jam gave some relief to the AMA in this run.

For 18C, the average queue results show minimal increases between simulations. There is more room in both the ramp area and AMA near 18C than near 18L. Whereas areas near 18C did not saturate, traffic jams near 18L may have restricted the flow of departures to 18C. The maximum queues for 18C actually occur during relatively light demand periods when departures to 18C are allowed to flow more freely.

In actual operations, interviews with CLT ramp managers reveal that they try to maintain a q_{taxi} of ~10 departures. The maximum q_{taxi} for 18L is within one departure of this desired queue value for B values between 2 and 5. However, for $B > 5$, the maximum q_{taxi} values for 18L are assumed to be undesirable.

VI. Summary and Conclusions

A model of the ATD-2 tactical scheduler was implemented in fast time simulation and evaluated for varying values of taxi time delay buffer, B from (1).

The two most interesting shifts in scheduler evaluation metrics occurred when B was increased from 1 to 2 minutes. Average total delay decreases began to level off and the average throughput prediction at flight ready time shifted from over prediction (negative error) to under prediction (positive error). The goal of surface metering is to move as much delay as possible from taxiing to the gate without negatively impacting total delay. Because throughout is lost when the departures are over metered, it is better to under predict throughput and allow the queues to temporarily increase than to over predict and allow the queues to reduce to 0 and let the runway go dry. In actual operations, CLT ramp managers prefer maintain a count of departure taxiing to a particular runway ~10, which runway 18L exceeds for $B > 5$. For these reasons, B values in the range of 2 to 5 minutes are recommended for future SOSS simulation studies for the ATD-2 tactical scheduler implemented at CLT. Other factors, such as additional uncertainties and human factors, may influence the desired delay buffer, requiring it to be calibrated separately in the field.

This study provides a benchmark for fast-time exploration of design options for other ATD-2 tactical scheduler features. Future research building off of this benchmark will include additional uncertainties, external traffic management initiatives, and airline priority flights.

Acknowledgment

This work was sponsored by NASA Airspace Operations and Safety Program's Aviation Technology Demonstration 2 (ATD-2) subproject.

References

[1] S. Zelinski and R. Coppenbarger, "A Concept for Integrated Arrival/Departure/Surface (IADS) Traffic Management for the Metroplex," Airspace Technology Demonstration 2 (ATD-2) ConOps Synopsis, NASA Ames Research Center, August 2015, unpublished.

[2] NASA ATD-2 ConUse Team, "ATD-2 Concept of Use (ConUse) for Charlotte-Douglas International Airport for the Phase 1 Baseline Integrated Arrival, Departure, Surface (IADS) Demonstration," Working Paper Series: Informal Publications of the ATD-2 Sub Project, June 2016, unpublished.

[3] 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," Eleventh USA/Europe Air Traffic Management Research and Development Seminar (ATM2015), Lisbon, Portugal, June 2015.

[4] FAA Air Traffic Organization Surface Operations Office Directorate "U.S. Airport Surface Collaborative Decision Making (CDM) Concept of Operations (ConOps) in the Near-Term: Application of the Surface Concept at United States Airport," June 2014.

[5] C. Brinton, S. Lent, C. Provan, T. Prevost, and S. Passmore, "Collaborative Departure Queue Management: An Example of Airport Collaborative Decision Making in the United States," Ninth USA/Europe Air Traffic Management Research and Development Seminar (ATM2011), Berlin, Germany, June 2011.

[6] A. Nakahara, T. G. Reynolds, T. White, C. Maccarone, and R. Dunsky, "Analysis of a Surface Management Technique at New York JFK Airport," 11th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, Virginia Beach, Virginia, September 2011.

[7] S. Stroiney, B. Levy, H. Khadilkar, and H. Balakrishnan, "Assessing the Impacts of the JFK Ground Management Program," 32nd Digital Avionics Systems Conference (DASC), Syracuse, New York, October 2013.

[8] I. Simaiakis, H. Khadilkar, H. Balakrishnan, T. G. Reynolds, R. J. Hansman, B. Reilly and S. Urlass, "Demonstration of Reduced Airport Congestion Through Pushback Rate Control," Ninth USA/Europe Air Traffic Management Research and Development Seminar (ATM2011), Berlin, Germany, June 2011.

[9] I. Simaiakis, and H. Balakrishnan, "Dynamic Control of Airport Departures: Algorithm Development and Field Evaluation," American Control Conference, Montreal, Canada, July 2012.

[10] D. Böhme, "Tactical Departure Management with the Eurocontrol / DLR DMAN," Sixth USA/Europe Air Traffic Management Research and Development Seminar (ATM2005), Baltimore, Maryland, June 2005.

[11] M. Schaper, G. Tsoukala, R. Stravrati, and N. Papadopoulos, "Departure Flow Control Through Takeoff Sequence Optimization: Setup and Results of Trials at Athens Airport," 30th Digital Avionics Systems Conference (DASC), Seattle, Washington, October 2011.

[12] M. Schaper and I. Gerdes, "Trajectory Based Ground Movements and Their Coordination with Departure Management," 32nd Digital Avionics Systems Conference (DASC), Syracuse, New York, October 2013.

[13] NextGen Air Traffic Decision Support Systems. (n.d.) http://www.faa.gov/nextgen/update/progress_and_plans/decision_support_systems/ [cited on July 10, 2017].

[14] R. D. Windhorst, J. V. Montoya, Z. Zhu, S. Gridnev, K. Griffin, A. Saraf, and S. Stroiney, "Validation of Simulations of Airport Surface Traffic with the Surface Operations Simulator and Scheduler," 13th Aviation Technology, Integration, and Operations (ATIO) Conference, Los Angeles, California, August 2013.

[15] S. Zelinski and R. Windhorst, "Modeling and Simulating Airport Surface Operations with Gate Conflicts," Royal Aeronautical Society Flight Simulation Conference, London, United Kingdom, November 2017.