Control of a non-stationary tandem queue model of the airport surface

Sandeep Badrinath[1] and Hamsa Balakrishnan[2]

***Abstract*— Tandem queues have been used to model congestion in a wide variety of systems such as communication networks, manufacturing systems, supply chains and traffic flows. This paper considers the optimal control of tandem queues in order to mitigate surface congestion at large airports.**

The taxi-out process is modeled by two queues in tandem: the first one represents aircraft in a congested ramp or apron area, and the second one reflects aircraft waiting in the departure runway queue. The evolution of the mean queue lengths are described using ordinary differential equations. The resulting model is used to determine the optimal gate release rate for departure flights, in order to control the lengths of queues on the airport surface. Simulations of the optimal control policy show a reduction in queue lengths, resulting in lower taxi-out times.

I. INTRODUCTION

The lack of sufficient capacity at major airports worldwide has resulted in congestion, especially with the growth in air traffic demand over the past several years. This congestion, in turn, has resulted in increased taxi times, flight delays, fuel burn, and emissions. Increasing airport capacity typically requires substantial investments in infrastructure (such as new runways), and may not even be feasible due to tight operating budgets and limited availability of land. These factors have motivated the development of airport congestion control algorithms that better utilize available capacity.

An approach that has been demonstrated to mitigate congestion is to hold the departures at their gates by controlling the pushback rate during periods of congestion [1]. The pushback rate refers to number of aircraft that are allowed to start the taxi-out process from their gates per unit time. The control of pushback rate has been determined either using heuristics [2], or using dynamic programming [3]. Several previous studies have developed queuing models of the airport surface along with corresponding control algorithms [3], [4]. However, these approaches have generally assumed that the formation of queues is restricted to the runways, and that queues do not form in other locations on the airport surface, such as the ramp area. Moreover, time-varying demand and service rates have not been explicitly considered in prior work.

While some airports (for example, Boston Logan International Airport (BOS)) primarily experience queues only at the runways [3], others present an exception to this assumption. At these airports, additional queuing can occur in the ramp or apron area, namely, the region near the terminal buildings. One such prominent example is Charlotte Douglas International airport (CLT), a large hub airport that handles more than 1,400 aircraft operations each day [5]. The average taxi-out time (i.e., the time taken for an aircraft to travel from the gate to the runway) during peak hours can be as high as 30 minutes, even though the unimpeded taxi-out time (i.e., during periods of low traffic) is only around 12 minutes.

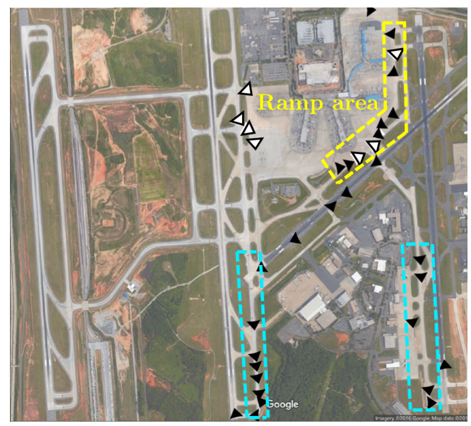


Fig. 1. Airport layout for Charlotte Douglas International airport (CLT). The flights taxiing-in and those taxiing-out are represented by gray and black triangles, respectively. The ramp queue and runway queue are shown in yellow and cyan, respectively.

Fig. 1 shows the physical layout of CLT, along with a snapshot of aircraft locations. The figure shows the formation of queues near the runway as well as on the ramp. As a result of the ramp queue, flights spend around 10 minutes on an average in the ramp area, which is nearly half of the mean taxi-out time. These queues are formed because the throughput of the runway and the spot (the exits from the ramp area into the active movement area) are both restricted due to operational constraints.

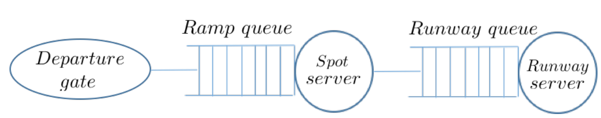


Fig. 2. Tandem queue model for the taxi-out process.

Observations of complex airports such as CLT suggest that such congestion in the ramp and runway is best represented using two separate queues in tandem, as shown in Fig. 2. In particular, this paper assumes that there are two different

*This work was partially supported by NSF under CPS: Frontier: FORCES (grant number 1239054).

[1]Sandeep Badrinath is a graduate student in the Department of Aeronautics and Astronautics at the Massachussetts Institute of Technology sandeepb@mit.edu

[2]Hamsa Balakrishnan is an associate professor in the Department of Aeronautics and Astronautics at the Massachussetts Institute of Technology hamsa@mit.edu

servers: a spot server that serves the ramp queue, and a runway server that serves the runway queues. Customers (aircraft) first enter the ramp queue and wait for service by the spot server, after which they enter the runway queue (potentially with some delay) and wait until they are served by the runway server. Such a tandem queue model would be capable of reflecting congestion at multiple regions, which is a critical need for airports like CLT.

Tandem queues, also known as queues in series, have been used to model a wide variety of systems, including communication networks, manufacturing systems, supply chains and urban traffic flows [6], [7]. Two aspects have been considered: the control of service rates in tandem queues [8], [6], and the control of arrival rates [9]. The latter problem has received relatively less attention [10], with the focus remaining on regulating arrivals into the second queue [11]. These works have assumed that the service rates and arrival rates are stationary. In general, previous research on such queuing models has been based on steady-state analysis, and few methods exist for non-stationary queues [12]. However, real world airports experience time-varying departure demand as well as service rates, due to factors such as the number of aircraft arriving from other airports, airline scheduling practices, weather phenomena, wind and visibility patterns, air traffic controller workloads, etc. There is therefore a need to develop control strategies that can account for non-stationary queues, especially for airports such as CLT that exhibit queues at multiple locations. This paper fills this gap in the literature by developing a non-stationary tandem queue model of the airport surface, and then determining an appropriate control policy (for the pushback rate) that regulates the queue lengths on the surface. Specifically, the pointwise stationary fluid flow approximation (PSFFA) model [13] is used to construct a tandem queue model of the airport surface. The PSFFA model yields a differential equation that governs the dynamics of the mean queue length, making it computationally efficient compared to other methods.

II. Model of non-stationary tandem queues

The pointwise stationary fluid flow approximation (PSFFA) model [13] is used to obtain the mean queue length of a non-stationary tandem queue. The model is a continuum approximation to the discrete queuing problem. The model is derived by combining results from the steady state queuing theory with the flow conservation principle. In this section, the model for a single queue is presented first, followed by a model for queues in tandem.

A. Single queue model

Let $x(t)$ represent the average number of customers in the queue. Let $f_i(t)$ and $f_o(t)$ represent the in-flow and out-flow from the queue at time t. All the quantities are ensemble averages at a particular time instant. From the flow conservation principle, we have:

$$\dot{x}(t) = -f_o(t) + f_i(t). \tag{1}$$

Let $\lambda(t)$ and $\mu(t)$ denote the average arrival rate and service rate, respectively, at time t. Assuming that there are no restrictions on the queue length, we have $f_i(t) = \lambda(t)$. For the out-flow, we can write $f_o(t) = \mu(t)\rho(t)$, where $\rho(t)$ is the average utilization of the server. The queue dynamics therefore takes the form:

$$\dot{x}(t) = -\mu(t)\rho(t) + \lambda(t). \tag{2}$$

The average utilization, $\rho(t)$, is approximated by a function, $G(x(t))$. This function $G(x(t))$ needs to satisfy the following properties: (a) $G(0) = 0$ and $G(\infty) = 1$; (b) $G(x)$ is strictly concave and non-negative $\forall x \in [0, \infty)$, in order to represent congestion. The dynamics for $x(t)$ can then be rewritten in terms of $G(x)$ as:

$$\dot{x}(t) = -\mu(t)G(x) + \lambda(t), \quad x(0) = x_0. \tag{3}$$

The expression for $G(x)$ is obtained by matching the steady-state number of customers in the system. If the arrival process is assumed to be Poisson, the Pollaczek-Khinchine formula ((4)) provides an expression for the mean number of customers (x_s) at steady state [14], namely,

$$x_s = \rho + \frac{\rho^2(1 + C_v^2)}{2(1 - \rho)}. \tag{4}$$

Here, C_v is the coefficient of variation of the service time distribution. Expressing ρ in terms of x_s, we get:

$$\rho = \frac{x_s + 1 - \sqrt{x_s^2 + 2C_v^2 x_s + 1}}{1 - C_v^2}. \tag{5}$$

Using the fact that $G(x)$ is an approximation for ρ, we obtain $G(x)$ as follows,

$$G(x) \approx \rho = \frac{x + 1 - \sqrt{x^2 + 2C_v^2 x + 1}}{1 - C_v^2}. \tag{6}$$

For example, since $C_v = 1$ for a M/M/1 queue, we can show that $G(x) = x/(1 + x)$ (using L'Hospital's rule).

B. Tandem queue model

Consider two servers with mean service rates μ_1 and μ_2 to be in tandem. Let x_1 and x_2 represent the mean number of customers in each of the two queues. Let λ_1 and λ_2 represent the arrival rates into the two queues. The arrival rate at the second server is approximated to be Poisson. This assumption is reasonable (and will be validated in Section IV-B), since the coefficients of variation of the service time distributions of both servers are high. Combining the dynamics of a single queue ((3)) along with the flow conservation principle, we get:

$$\lambda_2(t) = \mu_1(t)G_1(x_1). \tag{7}$$

The dynamics for the mean length of tandem queue is as follows:

$$\dot{x}_1(t) = -\mu_1(t)G_1(x_1) + \lambda_1(t) \tag{8}$$
$$\dot{x}_2(t) = -\mu_2(t)G_2(x_2) + \mu_1(t)G_1(x_1) \tag{9}$$

The functions $G_i(x)$ are further approximated by $C_i x/(1 + C_i x)$, in order to obtain a simpler expression for the dynamics. Note that when $C_i = 1$, it corresponds to a M/M/1 queue. The parameter C_i is determined through the following minimization:

$$\min_{C_i} \int_0^{x_m} (G_i(x) - C_i x/(1 + C_i x))dx. \tag{10}$$

A comparison between the approximation for $G(x)$ and its actual value is shown in Fig. 3. The value of C_v chosen for the comparison is based on the runway service time distribution which will be mentioned in Section IV-B. The overlap between the two curves in the figure shows that $Cx/(1 + Cx)$ is indeed a good approximation for $G(x)$.

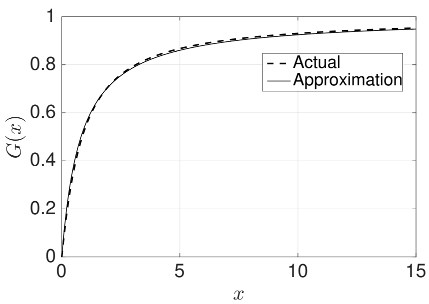


Fig. 3. Comparison between the approximation for $G(x)$ and the actual value ($C_v = 0.67, x_m = 15, C = 1.23$)

Using the above approximation for $G(x)$, the dynamics for the tandem queue is given by

$$\begin{aligned} \dot{x}_1(t) &= -\mu_1(t)\frac{C_1 x_1(t)}{1 + C_1 x_1(t)} + \lambda_1(t) && (11)\\ \dot{x}_2(t) &= -\mu_2(t)\frac{C_2 x_2(t)}{1 + C_2 x_2(t)} + \mu_1(t)\frac{C_1 x_1(t)}{1 + C_1 x_1(t)} && (12)\\ x_1(0) &= x_{1,0},\ x_2(0) = x_{2,0}. && (13) \end{aligned}$$

C. Tandem queues with delays

Consider that the customers arriving at a tandem queue are from a particular source. Let $u(t)$ correspond to the departure rate at the source. Let t_1 denote the travel time from the source to the first queue. Similarly, let t_2 denote the travel time from the first server to the second queue. Considering the travel time in the system, the arrival rates are given by,

$$\begin{aligned} \lambda_1(t) &= u(t - t_1) && (14)\\ \lambda_2(t) &= \mu_1(t - t_2)\frac{C_1 x_1(t - t_2)}{1 + C_1 x_1(t - t_2)}. && (15) \end{aligned}$$

Using the above expressions, we obtain modified governing equations for the queuing dynamics,

$$\dot{x}_1(t) = -\mu_1(t)\frac{C_1 x_1(t)}{1 + C_1 x_1(t)} + u(t - t_1) \tag{16}$$

$$\dot{x}_2(t) = -\mu_2(t)\frac{C_2 x_2(t)}{1 + C_2 x_2(t)} + \mu_1(t - t_2)\frac{C_1 x_1(t - t_2)}{1 + C_1 x_1(t - t_2)} \tag{17}$$

Since the dynamics has delay in its state variables, we require the following initial conditions to find the state of the queue at any time $t > 0$,

$$\begin{aligned} u(t) &= g(t), t \in [-t_1, 0) && (18)\\ x_1(t) &= \phi(t), t \in [-t_2, 0] && (19)\\ x_2(0) &= x_{2,0} && (20) \end{aligned}$$

Here, $g(t)$ and $\phi(t)$ are initial profiles of u and x_1.

III. Control of Tandem Queues

The control problem is motivated by the problem of airport surface congestion control. The aim of the control algorithm is to release flights at an optimal rate at the departure gate so that smaller queues are formed on the airport surface, resulting in reduced taxi-out times.

Consider the delayed tandem queue system described in Section II-C. Let $d(t)$ denote the demand rate at the source. The demand rate is the number of customers who are ready to leave the source per unit time. The control variable is the release rate at the source, which is denoted by $u(t)$. The number of customers who are held back by the controller is denoted by $h(t)$. The dynamics for $h(t)$ is given by

$$\dot{h}(t) = \begin{cases} (d(t) - u(t))^+ & h(t) = 0 \\ (d(t) - u(t)) & h(t) > 0 \end{cases} \tag{21}$$

Here, $(d(t) - u(t))^+ = \max\Big((d(t) - u(t)), 0\Big)$. The objective is to control the release rate $u(t)$ to minimize the queue length while achieving maximum throughput.

A. Problem formulation

The problem is formulated as an optimal control problem. The state variables are the number of customers in two queues $(x_1(t), x_2(t))$ and the number of customers held at the source $(h(t))$. The problem formulation is as follows,

$$\begin{aligned} &\min_u \int_0^T \Big(x_1^2 + x_2^2 + h^2/4 + (d - u)^2\Big)dt && (22)\\ &\text{Subject to:} && (23)\\ &\dot{x}_1 = f_1\Big(x_1(t), u(t - t_1), t\Big) && (24)\\ &\dot{x}_2 = f_2\Big(x_1(t - t_2), x_2(t), t\Big) && (25)\\ &\dot{h} = f_3\Big(d(t), u(t)\Big) && (26)\\ &0 \le x_1, x_2, h,\ 0 \le u \le u_m && (27)\\ &u(t) = g(t), t \in [-t_1, 0) && (28)\\ &x_1(t) = \phi(t), t \in [-t_2, 0] && (29)\\ &x_2(0) = x_{2o},\ h(0) = h_o && (30) \end{aligned}$$

In the above equations, f_1, f_2, f_3 represent the dynamics of the state variables and are given by (16)-(17) and (21). The quadratic cost function penalizes the queue length, the number of customers held at the source and the instantaneous deviation from the planned departure rate. The holding term in the cost function ensures maximum utilization of the server. The deviation from the planned departure rate is included in the cost function to obtain a smooth solution.

B. Solution methodology

The solution to the optimal control problem is obtained by discretizing the state and control variables in time. The equations governing the dynamics are discretized using first order Euler method. Higher order discretization schemes like the Runge-Kutta method could also be used. The discretized control problem is then transformed into a non-linear programming problem (NLP). The detailed procedure is similar to that shown in [15]. The resulting NLP is solved using a standard MATLAB® solver.

IV. Taxi-out process model

A. Description of the data

The tandem queue model for the taxi-out process is constructed using the data for Charlotte Douglas International airport (CLT) as an example. Data for the analysis was extracted from multiple datasets. The flight tracks were obtained from airport surface surveillance data (ASDE-X) [16]. These tracks were used to determine the time at which the aircraft reached the spot or runway. This information was used to compute the service time distributions of the spot and runway servers. The flight schedule that includes the actual pushback time for the departure flights and the landing time for the arrival flights was obtained from OAG data [17].

B. Representation of surface queues

The taxi-out process is modeled as two queues in tandem, as shown earlier in Fig 2. The first one represents the congestion at the ramp, and the second one represents the congestion near the runway. Note that a single queue is considered for the ramp even though flights might exit the ramp at different spots. Similarly, a single queue is considered for representing multiple departure runways (two, in the case of CLT).

A flight is defined as being in the ramp queue if it is yet to reach the spot, and its travel time has exceeded the unimpeded gate to spot time. Similarly, a flight is said to be in the runway queue if it is yet to takeoff and its travel time from the spot has exceeded the unimpeded spot to runway time. A comparison of the runway queue lengths obtained using the above definition, with the queue observed using the physical locations of flights, is shown in Fig. 4. The figure shows a good match between the queue lengths obtained using the two methods. This shows that the above definition of queue length obtained from using unimpeded time is equivalent to the physical queue length seen at an airport.

As mentioned earlier, a key feature of airport operations are that the service times of the spot and runway servers are stochastic and time-varying. The time variability is primarily due to fluctuations in the number of landings and flights taxiing-in at the airport. Other factors such as weather and aircraft type could also influence the service time. However, these factors were found not to play an important role for the periods considered. The mean service time for the spot server as a function of number of flights taxiing-in on the ramp is shown in Fig. 5(a). The mean service time shows an increasing trend with increase in number of flights taxiing-in. Similarly, the mean service time at the departure runway server increases with an increase in the number of landings, as seen in Fig. 5(b). The distributions of the service times for the two servers is shown in Fig. 6. The service time distribution is conditioned on the taxi-in traffic for the spot server, and the number of landings for the runway server. A large variability in the service times can also be seen from these plots. The coefficient of variation is around 0.86 for the ramp queue, and 0.67 for the runway queue.

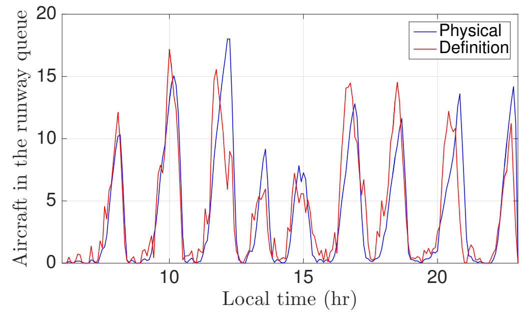
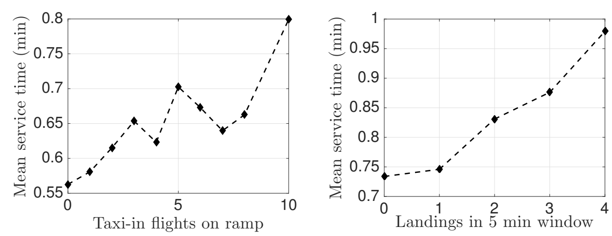
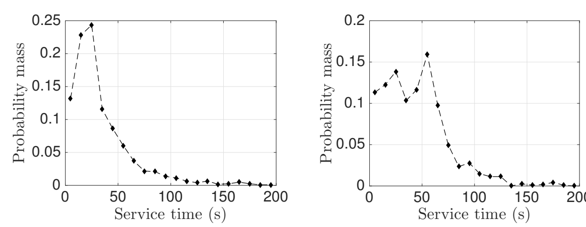


Fig. 4. Comparison of the total queue length obtained from the definition of the runway queue, and the physical queue observed at the airport for a typical day.



(a) Spot server (b) Runway server

Fig. 5. Mean service times of the spot and runway servers, as functions of the number of aircraft taxiing-in.



(a) Spot server (Flights taxiing-in on ramp = 0) (b) Runway server (Landings in 5 min window = 0)

Fig. 6. Service time distributions of the spot and runway servers, when the number of aircraft taxiing-in and the number of landings are zero, respectively.

C. Simulations

The queuing system is simulated in MATLAB® to validate the assumption that the service time distributions depend only on the taxi-in traffic. The inputs to the simulator are the actual push back times, landing schedules, and number of flights taxiing-in on the ramp. In the simulation, flights are added into the system at the actual pushback times (t_p), and each aircraft reaches the ramp queue at $t_p + t_{u,gs}$.

Here, $t_{u,gs}$ is the unimpeded gate-to-spot time, averaged over all gate-spot combinations. Each aircraft waits in the ramp queue for a time W_{rp}, depending on the current state of the ramp queue. The service times of the spot are sampled from an empirical distribution conditioned on the taxi-in traffic. Once the aircraft is served by the spot server, it enters the runway queue at $t_p + t_{u,gs} + W_{rp} + t_{u,sr}$, where $t_{u,sr}$ is the unimpeded spot-to-runway time, averaged over all spot-runway combinations. The aircraft then waits in the runway queue for a time W_{rw}, depending on the state of the runway queue, before being served. The taxi-out time (t_{out}) is therefore the sum of the unimpeded times and the wait time at the queues, that is:

$$t_{out} = t_{u,gs} + W_{rp} + t_{u,sr} + W_{rw}. \tag{31}$$

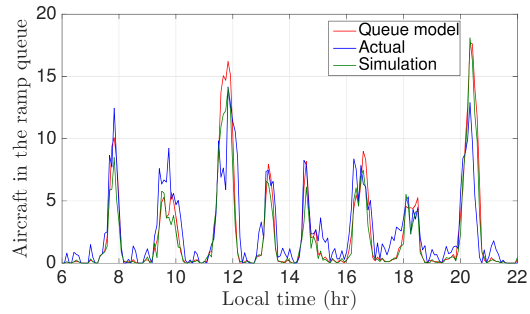
A comparison of the mean number of flights in the queue from the simulation, with actual data, is shown in Fig. 7. The simulation is repeated 1,000 times to obtain the mean queue length. The figure shows that the mean queue length from the simulation matches well with the actual queue length, validating our assumption on the service time distributions.

D. Tandem queue model of the taxi-out process

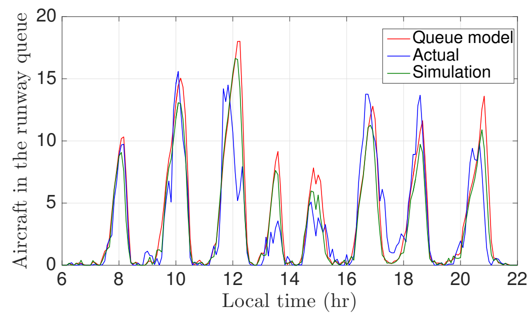
The tandem queue model developed in Section II-C is applied to the taxi-out process. In this case, x_1 and x_2 represent the queue length at the ramp and runway, respectively, $u(t)$ corresponds to the pushback rate at the gate, and the time delays in the dynamics correspond to the unimpeded travel times ($t_1 = t_{u,gs}$, $t_2 = t_{u,sr}$). The queue length at any time instant is obtained by integrating (16)-(17) forward in time. The model is validated by comparing the results with the actual data and simulation. The comparisons of the number of flights in the ramp and runway queues are shown in Fig. 7. The figure shows that the number of flights in the two queues obtained from the proposed tandem queue model matches both the actual data and the simulations well.

The wait time for each flight depends on the number of flights ahead of it when it enters the queue as well as the mean service rate. The average taxi-out time obtained from the tandem queue model is compared with the actual data and simulation in Fig. 8. A good agreement can be seen between the average taxi-out times obtained from the model, simulations, and the actual data.

Table I shows the error statistics for the travel times obtained from the queue model when compared with the actual data. These statistics are based on a total of 10,454 departure flights. The mean error for taxi-out time is 1.15 min. This value is small when compared to the average taxi-out time during peak traffic periods, which can be as high as 30 min. Taxi-out time obtained using standard machine learning techniques [18] have similar error statistics. However, the main advantage of using the queuing model is that it provides a simple equation governing the queue dynamics in terms of the pushback rate, which can be used to control the queue length.



(a) Ramp queue



(b) Runway queue

Fig. 7. Comparison of the number of aircraft in the queue (averaged over a 5 min window) obtained using the tandem queue model, actual data and simulations for a typical day.

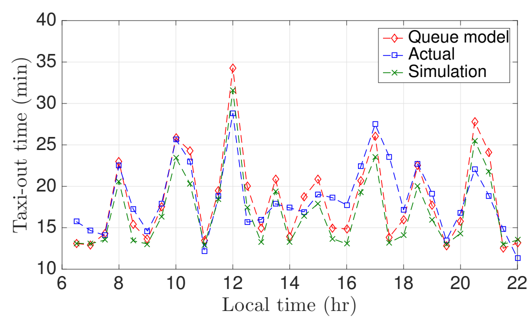


Fig. 8. Comparison of the taxi-out times (averaged over a 30-min window) obtained from the tandem queue model, actual data and simulations for a typical day.

V. Pushback rate control

During periods of congestion at an airport, it is better for the aircraft to wait at the departure gate with their engines off, than to wait in a queue with their engines on. The strategy is therefore to control the pushback rate, in order to achieve smaller queue lengths on the airport surface. The problem formulation and solution methodology to control a tandem queue were discussed in Section III. In this section, a solution is obtained to control the ramp and runway queues. Here, the demand rate ($d(t)$) refers to the number of flights that are ready to pushback from their departure gate per unit time, while the number of holds ($h(t)$) refers to the number of aircraft that are ready to pushback, but are held at the gate by the controller at time t.

TABLE I

ERROR STATISTICS FOR THE TAXI-OUT TIME BASED ON 10,454 FLIGHTS.

Statistic (min)	Gate to spot	Spot to runway	Taxi-out
Mean error	-0.04	1.16	1.15
Mean |error|	3.15	3.48	5.15
RMSE	4.5	4.3	6.8

The entire day is divided into 5-min long time-windows, and the pushback rate is determined for each time-window. An additional constraint that ensures a constant value of pushback rate over each time window is imposed in the NLP. The actual pushback rate from the data is considered as the demand rate ($d(t)$). The number of landings and number of flights taxiing-in on the ramp are assumed to be known ahead in time, and are used to determine the service rates ($\mu_1(t), \mu_2(t)$) of the spot and runway servers.

A. Optimal pushback rate

The optimal pushback rate profile is computed for a typical day, and benefits of the policy are discussed. Fig. 9 shows the optimal pushback rate along with the demand rate, over the course of the day. The resulting queue lengths and the number of gate holds are shown in Fig. 10. Initially, as the queue builds up, the pushback rate is equal to the demand rate for small queue lengths. When the queue size is large, the pushback rate is smaller than the demand rate and flights are held back at the gate. This policy ensures that the taxi-out flights do not have to go through a larger queue. Note that the pushback rate has a wider peak to accommodate the spill in the demand. This results in sustained runway utilization with smaller queue lengths. One can also notice that the runway queue length is typically larger than the ramp queue length. This is due to the fact the runway server has a higher mean service time when compared to the spot server. The number of holds is seen to be considerably large. This is due to the fact that the cost function gives the holding cost a smaller weight relative to the queuing cost.

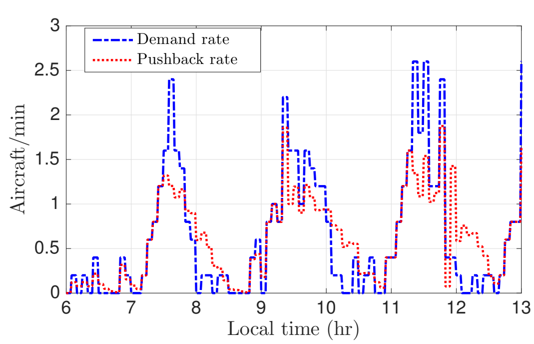


Fig. 9. Variation of the optimal pushback rate (aircraft/min) for a given demand profile on a typical day.

Simulations are performed with the optimal pushback rate to test the efficacy of the policy in a stochastic environment. The simulation environment used here is same as the one mentioned earlier in Section IV-C. A comparison of the ramp and runway queue lengths between the controlled and uncontrolled cases is shown in Fig. 11. The queue lengths

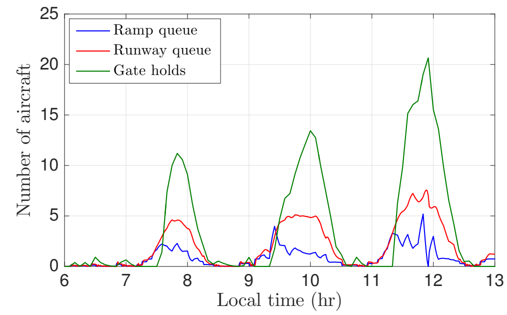


Fig. 10. The number of aircraft in the ramp queue (x_1), runway queue (x_2), and gate holds (h) with the optimal push back policy.

obtained with the optimal pushback policy are found to be significantly smaller than the queue lengths without any gate-holds. Fig. 12 shows a comparison of the the taxi-out times between the two cases. A significant reduction in the taxi-out times can be seen, which is a result of smaller queue lengths.

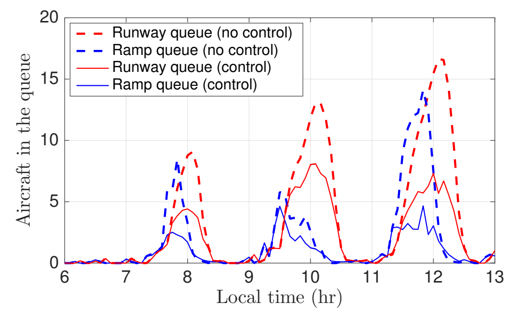


Fig. 11. Comparison of the number of flights in the ramp and runway queues with, and without, the optimal pushback policy.

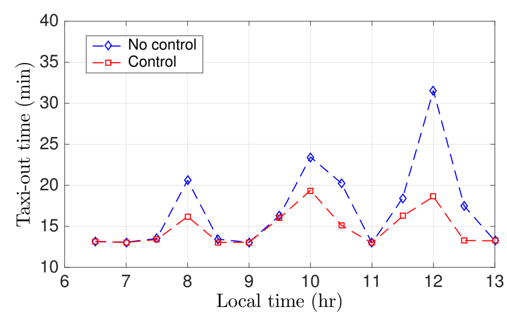


Fig. 12. A comparison between the mean taxi-out time (30-min average) obtained using the pushback control policy, and when aircraft pushback without any gate-holds.

B. Receding horizon control

The control algorithm described earlier has two major drawbacks: (a) It assumes that the demand rate and service rate are known for long periods in advance, when in reality, this information might be available only 30 min into the future; and (b) The algorithm does not utilize the information about the current state of the airport. To overcome these drawbacks, a receding horizon control approach is applied.

The approach uses the same formulation as used earlier. At the beginning of every time-window (5-min), the control policy is determined for the next 30 min, utilizing the current state of the airport as the initial condition. However, only the first 5 min of the control policy is implemented by the controller. The process is repeated every 5 min.

The proposed approach is tested using simulation. The results from 40 simulation runs are used to obtain the mean queue length. Fig. 13 shows the mean number of flights in the ramp and runway queue obtained using the receding horizon control policy. The queue length obtained from the new policy yields smaller queue length when compared to the case with no pushback control. This is also reflected in the taxi-out time, which is shown in Fig. 14. Moreover, one can see that the receding horizon method performs slightly better than the earlier method. The improvement can be attributed to the fact that the state of the airport is being used to compute the pushback rate. The performance of the control policy also depends on the time horizon. A longer time horizon is expected to yield a better performance, but with additional computational cost.

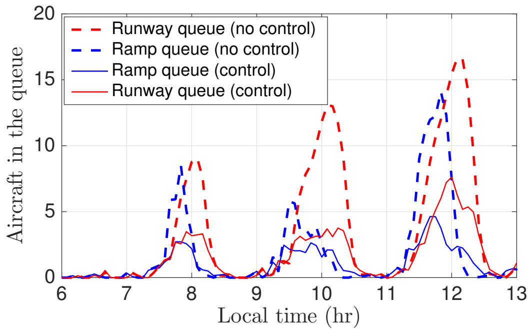


Fig. 13. Comparison of the number of flights in the queue obtained using the receding horizon control policy.

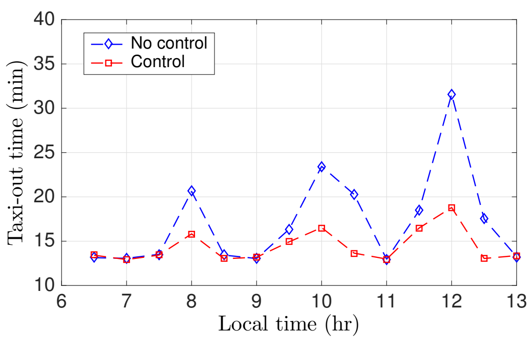


Fig. 14. Comparison of the averaged taxi-out time obtained using the receding horizon control policy

VI. Discussion and Future Work

This paper proposed an optimal control methodology to control a non-stationary tandem queue. As an illustrative example, the control methodology was applied to determine the optimal pushback rate for the taxi-out process in an airport. Simulations with the optimal control policy indicated a significant reduction in the queue sizes at the airport, resulting in lower taxi-out times.

The model can be extended to control a network of non-stationary queues, since the dynamics of each queue is governed by a simple differential equation. In addition to pushback rate control, the fraction of flights routed to each runway can be controlled in order to have more balanced runway operations and a better utilization of airport capacity. This can be achieved by modeling the runway as two queues instead of one. Another promising extension is the incorporation of the the taxi-in process into the model, which would be particularly useful to understand the effects of active runway crossings. Such a model would enable an integrated control of airport surface operations.

Acknowledgment

We would like to thank Tom Reynolds and Melanie Sandberg of MIT Lincoln Labs for providing the airport surface surveillance data (ASDE-X) and ASPM data.

References

[1] I. Simaiakis, H. Khadilkar, H. Balakrishnan, T. G. Reynolds, and R. J. Hansman, “Demonstration of reduced airport congestion through pushback rate control,” *Transportation Research Part A: Policy and Practice*, vol. 66, pp. 251–267, 2014.

[2] N. Pujet, B. Delcaire, and E. Feron, “Input-output modeling and control of the departure process of congested airports,” *Air Traffic Control Quarterly Journal*, vol. 8, no. 1, p. 1, 2000.

[3] I. Simaiakis, M. Sandberg, and H. Balakrishnan, “Dynamic control of airport departures: Algorithm development and field evaluation,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 1, pp. 285–295, 2014.

[4] P. McFarlane and H. Balakrishnan, “Optimal control of airport pushbacks in the presence of uncertainties,” in *2016 American Control Conference (ACC)*. IEEE, 2016, pp. 233–239.

[5] Federal Aviation Administration (FAA), “Aviation System Performance Metrics (ASPM) website,” 2016, http://aspm.faa.gov/.

[6] M. H. Veatch and L. M. Wein, “Optimal control of a two-station tandem production/inventory system,” *Operations Research*, vol. 42, no. 2, pp. 337–350, 1994.

[7] C. Wang, “Urban transportation networks: Analytical modeling of spatial dependencies and calibration techniques for stochastic traffic simulators,” Ph.D. dissertation, Massachusetts Institute of Technology, 2013.

[8] Z. Rosberg, P. Varaiya, and J. Walrand, “Optimal control of service in tandem queues,” *IEEE Transactions on Automatic Control*, vol. 27, no. 3, pp. 600–610, 1982.

[9] S. Stidham, “Optimal control of admission to a queueing system,” *IEEE Transactions on Automatic Control*, vol. 30, no. 8, pp. 705–713, 1985.

[10] H. A. Ghoneim and S. Stidham, “Control of arrivals to two queues in series,” *European Journal of Operational Research*, vol. 21, no. 3, pp. 399–409, 1985.

[11] B. Zhang and H. Ayhan, “Optimal admission control for tandem queues with loss,” *IEEE Transactions on Automatic Control*, vol. 58, no. 1, pp. 163–167, 2013.

[12] K. M. Malone, “Modeling a network of queues under nonstationary and stochastic conditions,” Ph.D. dissertation, Massachusetts Institute of Technology, 1993.

[13] D. Tipper and M. K. Sundareshan, “Numerical methods for modeling computer networks under nonstationary conditions,” *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 9, pp. 1682–1695, 1990.

[14] W.-P. Wang, D. Tipper, and S. Banerjee, “A simple approximation for modeling nonstationary queues,” in *INFOCOM’96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proceedings IEEE*, vol. 1. IEEE, 1996, pp. 255–262.

[15] L. Gollmann, D. Kern, and H. Maurer, “Optimal control problems with delays in state and control variables subject to mixed control-state constraints,” *Optimal Control Applications and Methods*, vol. 30, no. 4, pp. 341–365, 2009.

[16] *ASDE-X brochure - ASDE-X 10/05.qxd*, Sensis Corporation, East Syracuse, NY, 2008.

[17] OAG.

[18] H. Lee, W. Malik, and Y. C. Jung, “Taxi-out time prediction for departures at Charlotte airport using machine learning techniques,” in *16th AIAA Aviation Technology, Integration, and Operations Conference*, 2016, p. 3910.