Dynamic Control of Airport Departures: Algorithm Development and Field Evaluation

Ioannis Simaiakis and Hamsa Balakrishnan

***Abstract*— Surface congestion leads to significant increases in taxi times and fuel burn at major airports. In this paper, we formulate the airport surface congestion management problem as a dynamic control problem. We address two main challenges: the random delay between actuation (at the gate) and the server being controlled (the runway), and the need to develop control strategies that can be implemented in practice by human air traffic controllers. The second requirement necessitates a strategy that periodically updates the rate that departures pushback from their gates.**

We model the runway system as a semi-Markov process using surface surveillance data. We use this modeling framework to derive optimal pushback policies to control congestion. Finally, we present the results of the real-world implementation and field testing of this control protocol at Boston Logan International Airport.

I. Introduction

Airport surface congestion contributes significantly to taxi times, fuel burn and emissions at airports. Annually, taxi-out delays at major US airports exceed 32 million minutes, while taxi-in delays exceed 13 million minutes [1]. The objective of this paper is to develop a control policy that can reduce surface congestion and its impacts and is amenable to implementation in practice.

A. Related work

An airport congestion control strategy in its simplest form would be a state-dependent pushback policy aimed at reducing surface congestion. One such approach is the *N-Control* strategy, which was initially considered in the Departure Planner [2], and has been extensively studied since [3], [4], [5]. The N-Control policy is effectively a simple threshold heuristic: If the total number of departing aircraft on the ground exceeds a certain threshold, further pushbacks are stopped until the number of aircraft on the ground drops below the threshold. A similar heuristic, based on the concept of an Acceptable Level of Traffic (ALOT), is used by Air Traffic Controllers at BOS during extreme congested situations [6]. The N-Control policy is also closely related to *constant work-in-process* (CONWIP) policies in manufacturing systems, which are used because of their simplicity, implementability and controllability [7].

More complex policies which attempt to attain some optimization objective have also been considered for surface traffic recently. In 2009, Burgain et al. used more advanced modeling and optimization tools for the characterization of optimal pushback policies. They modeled the airport surface as a Markov chain, and characterized optimal pushback policies as a function of the state of the system, and not just the total number of aircraft on the ground [8]. However, the optimal policies considered were still restricted to threshold policies, and challenges remained regarding the state space modeling, and the dimensionality of the problem[5]. All the above policies (N-Control, CONWIP systems, and Burgain et al.'s refinements) can be classified as token-based, or surplus-based policies [9]. In these approaches, every state transition generates a token, an action or a signal, which is applied at the input to the system (the pushback process). Equivalently, every state transition translates to a new surplus level (or lack thereof) at different buffers of the system, which implies a different flow of input into the system.

There has been much prior research on the optimal control of a variety of queuing systems, considering different decision variables and control objectives [10], [11], [12]. However, several challenges remain when attempting to apply results from queuing, manufacturing and inventory control in the context of controlling the departure process. Firstly, on-off or event-driven control policies for controlling the pushback process are difficult to implement in practice. Both the air traffic controllers and the airlines would prefer a state-dependent dispatch rate that would be valid for a predefined time period, after which it would be updated. Air traffic controllers prefer such periodically updated pushback rate recommendations for workload and procedural reasons, and airlines prefer them because of their predictability. Secondly, the control input is applied at the gates during pushback, whereas the main bottleneck is the runway. The control strategy cannot be applied directly at the runway queue, but instead has to accommodate stochastic taxi-out times between the gate and the runway. Reasons behind the stochasticity of taxi-out times include the pushback process, flight checklists, communication delays, and variable taxi speeds. Finally, Eulerian models that are concerned with controlling aircraft flows rather individual aircraft trajectories, have been used in the context of air traffic flow management [13], [14]. However, these dynamic control approaches have not been previously applied to surface operations.

In recent work [15] we developed and tested a Pushback Rate Control protocol (henceforth referred to as PRC.v1.0), which was an adaptation of the N-Control policy. PRC_v1.0 suggested a rate at which aircraft push back from their gates, so as to keep the airport from reaching highly congested

This work was supported by the FAA through MIT Lincoln Laboratory under Air Force Contract No. FA8721-05-C-0002.

I. Simaiakis is a PhD student in the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology ioa_sim@mit.edu

H. Balakrishnan is Assistant Professor of Aeronautics and Astronautics at the Massachusetts Institute of Technology hamsa@mit.edu

states. The rate was periodically updated based on the operating conditions (weather, configuration and arrival demand) and the total number of active departures on the surface. In this paper, we develop, using dynamic programing, a refined version of PRC_v1.0, which we call PRC_v2.0. We propose a semi-Markov process model of the runway queuing system, and derive optimal control policies (the rate at which to release aircraft from their gates) to balance the tradeoff between congestion and the risk of low runway utilization.

II. Control strategy requirements

The strategy must be compatible with current levels of information and automation in the airport tower. In addition, it must be incorporated into current operational procedures with minimal controller workload and procedural modifications. As mentioned in Section I, the preferred form of a congestion control strategy is one that recommends a pushback rate for departures to air traffic controllers. This recommended pushback rate is updated periodically based on conditions on the airport surface.

In general, the length of the time period, Δ, should equal the delay between the application of the control input (that is, setting an arrival rate for the runway server by controlling the pushback rate) and the time that the runway sees that rate. For the departure process, this time delay is given by the travel time from the gates to the departure queue. By choosing a time horizon that is approximately equal to the expected travel time from the gates to the departure queue, the flights released from the gate during a given time period are expected to reach the departure queue in the next time period.

III. Departure process model

At any time t, the state N_t of the departure process consists of the number of aircraft traveling from the gates to the departure queue (T_t) and the number of aircraft in the departure queue (D_t):

$$N_t = (T_t, D_t) \tag{1}$$

$$W_t = T_t + D_t \tag{2}$$

T_t and D_t can be observed using surface surveillance data, or by counting flight strips in the airport tower. W_t is the total number of aircraft taxiing out, or the total work-in-process of the departure process.

A. Pushback process

At the beginning of each time period the decision maker chooses a pushback rate (arrival rate into the surface system) of $\lambda \in \Lambda = [0, \hat{\lambda}]$. λ is expressed as the number of pushbacks per Δ minutes. The time instances at which the pushback rate is updated are called *epochs*. In contrast to typical dynamic queuing control problems in which the decision maker sets the arrival rate into a facility, in our case, when setting a pushback rate at epoch τ, the decision maker authorizes λ aircraft to push back in that time period. In other words, λ pushbacks will occur in the time period $(\tau, \tau+\Delta]$ with probability 1 (w.p. 1). Furthermore, λ is an integer: $\lambda \in [0, 1, \ldots, \hat{\lambda}]$.

B. Runway service process

The model treats the departure runways as a single server where aircraft line up (queue) to await takeoff. The queuing system has finite queuing space C, which depends on the airport layout and operational procedures. At each airport, there is an upper bound on the number of aircraft that can queue up, which is the queuing space C of the queuing system. The runway service times are modeled as being Erlang distributed. The shape and rate $(k, k\mu)$ of the distribution are extracted from surveillance (ASDE-X) data, as will be explained in Section IV. The arrival times at the queuing system are modeled to be random and independent from each other. However, at each epoch, the total number or aircraft taxiing out (traveling from the gate to the departure queue) is known (denoted R_τ). We assume that by the next epoch, all of them (R_τ) will have reached the runway server. We show later how this assumption can be relaxed. In summary, the arrival process at the runway is modeled as a non-stationary Poisson process, in which the rate is updated every Δ minutes, and the process is conditioned on a given number of arrivals at the runway between two epochs.

This departure runway queuing system resembles a $M(t)/E_k/1$ system of queuing space C, with the additional constraint of R_τ arrivals during the $(\tau, \tau+\Delta]$ time interval. We denote it $(M(t)|R_\tau)/E_k/1$. Assuming that at epoch τ, R_τ aircraft are taxiing out, the probability density function g of the rth arrival at the departure runway at time t is:

$$\begin{aligned} g(r,t) &= \frac{R_\tau - (r-1)}{(\tau+\Delta)-t},\ t \in (\tau, \tau+\Delta],\ r = 0, 1, \ldots R_\tau \\ &= \frac{R_0 - (r-1)}{\Delta - t},\ \text{for } \tau = 0,\ t \in (0, \Delta],\ r = 0, \ldots R_0 \end{aligned} \tag{3}$$

To derive Equation (3), we consider $R_0 - (r-1)$ uniformly distributed random variables in the time interval $(\Delta - t]$. The probability that one of these lies in the interval $(t, t+dt]$ is $\big(R_0 - (r-1)\big)dt/(\Delta - t)$.

The state of the queuing system at time t is denoted by $S_t = (R_t, Q_t)$, where R_t is the number of aircraft that were taxiing out at the start of that epoch but have not reached the departure queue yet, and $Q_t \in \{0, 1, \ldots, kC\}$ is the state of the embedded chain of the semi-Markov process. An example of the chain for $k = 2$ and $C = 4$ is shown in Figure 1.

A service completion of an Erlang process with shape k and rate $k\mu$ is represented with k stages of exponentially distributed random variables with rate $k\mu$. We call each such stage "stage of work". Each state of the Markov chain (r, q) denotes that there are r aircraft that have been taxiing to the runway since the start of that epoch, and there are q *stages of work* to be completed at the departure runway server, i.e., there are $\min(1, q)$ aircraft in service and $\max(\lfloor (q-1)/k \rfloor, 0)$ aircraft in the departure queue.

At epoch 0, the Markov chain is in state (R_0, Q_0). With reference to Figure 1, the chain is in the bottom level of the chain (R_0 aircraft taxiing) with Q_0 stages of work to be completed. By the end of the time interval Δ, all of R_0 aircraft will have reached the departure queue, and the Markov chain will be at the top level (0 aircraft taxiing).

Let $P_{r,q}(t)$ denote the probability that the queuing system is in state (r,q) at time t, where $0 < t \leq \Delta$. The state probabilities $P_{0,0}(\Delta), P_{0,1}(\Delta), \cdots P_{0,kc}(\Delta)$ describe fully the state of the queuing system at the end of the time interval Δ. They are calculated by deriving the first-order differential equations (*Chapman-Kolmogorov equations*) that describe the evolution over the time $(0,\Delta]$, given R_0 arrivals in this interval: For $0 < t \leq \Delta$, and $1 \leq r < R_0$:

$$\frac{dP_{0,0}}{dt} = k\mu P_{0,1} \tag{4}$$

$$\frac{dP_{0,q}}{dt} = k\mu P_{0,q+1} - k\mu P_{0,q}, \quad 1 \leq q < k \tag{5}$$

$$\frac{dP_{0,q}}{dt} = k\mu P_{0,q+1} + \frac{1}{\Delta - t} P_{1,q-k} - k\mu P_{0,q}, \quad k \leq q < kC \tag{6}$$

$$\frac{dP_{0,kC}}{dt} = \frac{1}{\Delta - t} P_{1,k(C-1)} - k\mu P_{0,kC} \tag{7}$$

$$\frac{dP_{r,0}}{dt} = k\mu P_{r,1} - \frac{r}{\Delta - t} P_{r,0} \tag{8}$$

$$\frac{dP_{r,q}}{dt} = k\mu P_{r,q+1} - k\mu P_{r,q} - \frac{r}{\Delta - t} P_{r,q}, \quad 1 \leq q < k \tag{9}$$

$$\frac{dP_{r,q}}{dt} = k\mu P_{r,q+1} + \frac{r+1}{\Delta - t} P_{r+1,q-k} - \frac{r}{\Delta - t} P_{r,q} - k\mu P_{r,q}, \quad k \leq q \leq k(C-1) \tag{10}$$

$$\frac{dP_{r,q}}{dt} = k\mu P_{r,q+1} + \frac{r+1}{\Delta - t} P_{r+1,q-k} - k\mu P_{r,q}, \quad k(C-1) < q < kC \tag{11}$$

$$\frac{dP_{r,kC}}{dt} = \frac{r+1}{\Delta - t} P_{r+1,k(C-1))} - k\mu P_{r,kC} \tag{12}$$

$$\frac{dP_{R_0,0}}{dt} = k\mu P_{R_0,1} - \frac{R_0}{\Delta - t} P_{R_0,0} \tag{13}$$

$$\frac{dP_{R_0,q}}{dt} = k\mu P_{R_0,q+1} - \left(\frac{R_0}{\Delta - t} - k\mu\right) P_{R_0,q}, \; 1 \leq q \leq k(C-1) \tag{14}$$

$$\frac{dP_{R_0,q}}{dt} = k\mu P_{R_0,q+1} - k\mu P_{R_0,q}, \; k(C-1) < q < kC \tag{15}$$

$$\frac{dP_{R_0,kC}}{dt} = -k\mu P_{R_0,kC} \tag{16}$$

Solving Equations (4)-(16) numerically for time $t = \Delta$ with initial value (R_0, Q_0), we obtain the state probabilities $P_{0,0}(\Delta), P_{0,1}(\Delta), ...P_{0,kC}(\Delta)$. The state of the queuing system at time Δ, Q_Δ, is a probabilistic function f of the initial value (R_0, Q_0), and the probabilities $p_{q(i)}$ of each state i are the calculated probabilities $P_{0,i}(\Delta)$:

$$Q_\Delta = f(R_0, Q_0) \tag{17}$$

$$\text{with} \quad p_{q(i)}(R_0, Q_0) = P_{0,i}(\Delta) \quad \text{for } 0 \leq i \leq kC \tag{18}$$

$$\implies \quad \vec{p}_q(R_0, Q_0) = \vec{P}_0(\Delta) \tag{19}$$

where $\vec{P}_0(\Delta) = [P_{0,0}(\Delta), P_{0,1}(\Delta), ...P_{0,kC}(\Delta)]$.

C. System dynamics

Suppose, at epoch τ, that R_τ aircraft are taxiing, Q_τ stages of work are left to be completed in the queue, and the decision maker selects a pushback rate λ_τ. At $\tau + \Delta$, R_τ aircraft will have reached the departure queue, λ_τ aircraft will be taxiing, and $Q_{\tau+\Delta} = f(R_\tau, Q_\tau)$ stages of work will remain to be completed. The queuing system therefore evolves according to the following equation:

$$(R_{\tau+\Delta}, Q_{\tau+\Delta}) = (\lambda_\tau, f(R_\tau, Q_\tau)) \tag{20}$$

The probabilities $\mathbb{P}_{(r,q)\to(i,j)}(\lambda)$ that the chain is in state (i,j) at the next epoch $\tau + \Delta$ given it is in state (r,q) at the epoch τ and the pushback rate λ is chosen are:

$$\mathbb{P}_{(r,q)\to(i,j)}(\lambda) = \begin{cases} p_{q(j)}(r,q) & \text{if } i = \lambda \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

The state S of the queuing system maps to the state of the departure process (N) as follows:

$$N_t = \begin{cases} \left(\lambda_{t-\Delta}, \max(\lfloor (Q_t - 1)/k \rfloor, 0)\right), & t \in \{0, \Delta, \dots\} \\ \left(V_t + R_t, \max(\lfloor (Q_t - 1)/k \rfloor, 0)\right), & \text{otherwise} \end{cases} \tag{22}$$

where V_t is the number of aircraft that pushed back between the start of the epoch within which t lies, and the time t. We note that by sampling the system every Δ time intervals, we decouple the departure process into two processes that are independent within each time period, namely, the pushback process and the runway service process.

D. Choice of cost function

The control strategy sets the arrival rate to balance two objectives, namely, to minimize the expected departure queue length and to maximize the runway utilization. These requirements are captured in a cost function, $c(q)$ for a state (r,q) of the queuing system. This cost is a combination of the queuing cost and the cost of non-utilization of the runway. The runway is unutilized when $q = 0$. If $q \in \{1, 2, \dots k\}$ both the queuing and non-utilization costs are zero. For all higher states, $q > k$, there is a queuing cost $c(q)$, which is usually assumed to be a monotonically non-decreasing function of q with increasing marginal costs [16], [17]: A candidate cost function with these properties is:

$$c(q) = \begin{cases} l, & q = 0 \\ (\lfloor (q-1)/k \rfloor)^2 & q = 1, \dots, kC \end{cases} \tag{23}$$

where l^2 is the cost of a loss of runway utilization.

We solve Equations (4)-(16) numerically to calculate $\vec{p}_q(R_0, Q_0, t) = \left[\sum_{r=0}^{R_0} P_{r,0}(t), \sum_{r=0}^{R_0} P_{r,1}(t), \dots, \sum_{r=0}^{R_0} P_{r,kC}(t)\right]$ at time t. Numerical experiments showed that sampling every 0.1 min is sufficiently accurate for calculating the expected cost of each state, $\bar{c}$ over the time interval Δ:

$$\bar{c}(R_0, Q_0) = \sum_{i=0}^{10\Delta - 1} \frac{1}{10} \vec{p}_q(R, Q, i/10) \cdot \vec{c} \tag{24}$$

E. Dynamic control of the departure process

The Bellman equation for the infinite horizon problem with discount factor α is:

$$\begin{aligned} J^*(r,q) &= \min_{\lambda \in \Lambda} \{\bar{c}(r,q) + \alpha \sum_{j=0}^{kC} \mathbb{P}_{(r,q)\to(\lambda,j)} J^*(\lambda, j)\} \\ \implies J^*(r,q) &= \min_{\lambda \in \Lambda} \{\bar{c}(r,q) + \alpha \vec{p}_q(r,q) \cdot \vec{J}^*(\lambda) \end{aligned} \tag{25}$$

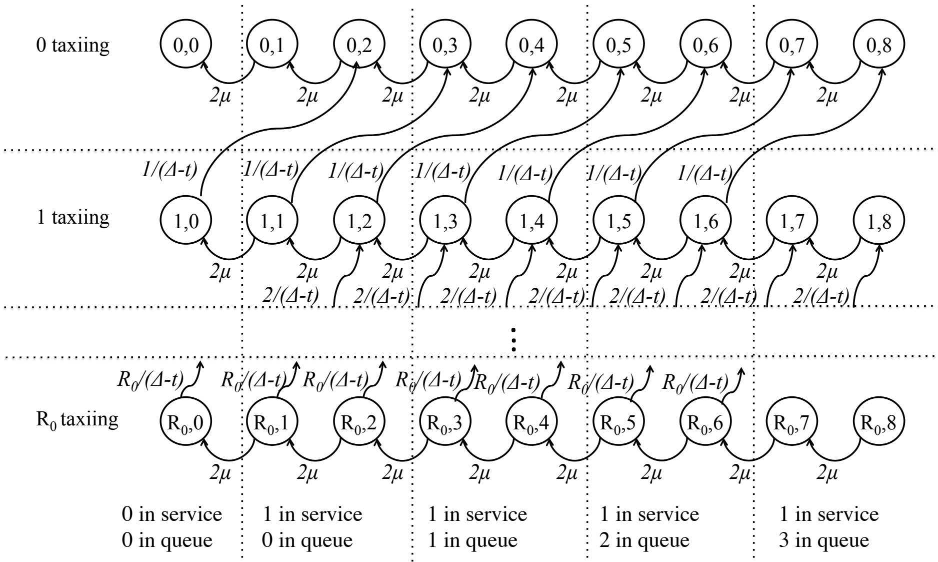


Fig. 1: State transition diagram for an $(M(t)|R_0)/E_2/1$ system with queuing space of 4 customers in the system.

where $\vec{J}^*(\lambda) = [J^*(\lambda,0), J^*(\lambda,1), \ldots, J^*(\lambda,kC)]$ for $r \in \{0,1,\ldots,\hat{\lambda}\}$ and $q \in \{0,1,\ldots,kC\}$.

We relax the assumption of Equation (20) that R_τ aircraft taxiing out at epoch τ will reach the queue during the time interval $(\tau, \tau+\Delta]$ and a pushback rate (λ_τ) set at epoch τ will arrive at the runway at $t > \tau+\Delta$ w.p. 1, as follows. For each λ_τ and R_τ, i out of λ_τ aircraft reach the runway during the time interval $(\tau, \tau+\Delta]$ with probability β_i. Similarly, i out of R_τ aircraft reach the runway at $t > \tau+\Delta$ with probability γ_i. Finally, R_τ aircraft reach the runway during the time interval $(\tau, \tau+\Delta]$, and λ_τ aircraft at $t > \tau+\Delta$ only with probability $1-\sum q_i^a - \sum q_i^b$.

Equation (20) becomes:

$$(R_{\tau+\Delta}, Q_{\tau+\Delta}) = \begin{cases} \big(\lambda_\tau, f(R_\tau, Q_\tau)\big), & \text{w.p. } 1-\sum\beta_i-\sum\gamma_i \\ \big(\lambda_\tau - i, f(R_\tau+i, D_\tau)\big), & \text{w.p. } \beta_i, i=1,\ldots,\lambda_\tau \\ \big(\lambda_\tau + i, f(R_\tau-i, D_\tau)\big), & \text{w.p. } \gamma_i, i=1,\ldots,R_\tau \end{cases}$$

In the most general case, β_i and γ_i are a function of both R and λ. For these system dynamics, the Bellman equation for the infinite horizon problem with discount factor α is:

$$\begin{aligned} J^*(r,q) = \quad & \min_{\lambda\in\Lambda}\Big\{(1-\textstyle\sum\beta_i-\sum\gamma_i)[\bar{c}(r,q)+\alpha\vec{p}_q(r,q)\cdot\vec{J}^*(\lambda)] \\ & +\textstyle\sum\beta_i[\bar{c}(r+i,q)+\alpha\vec{p}_q(r+i,q)\cdot\vec{J}^*(\lambda-i)] \\ & +\textstyle\sum\gamma_i[\bar{c}(r-i,q)+\alpha\vec{p}_q(r-i,q)\cdot\vec{J}^*(\lambda+i)]\Big\} \end{aligned} \tag{26}$$

Equation (26) illustrates the tradeoffs involved with the choice of appropriate time period, Δ. If the time period is large, a large fraction of the pushbacks will be likely to reach the runway in the current time period (large $\beta_i' s$). This will cause excessive congestion and might eventually lead to large traffic oscillations. If Δ is too small (large $\gamma_i' s$), finer control will be possible. However, as Δ decreases, the control strategy tends toward a *token-based* or *surplus-based* strategy, increasing controller workload. We also note that the portions β, γ and the probabilities p_β and p_γ do not need not be constants and can be a function of λ_τ.

Finally, we note that this problem satisfies the property of weak accessibility: Suppose that at the beginning of epoch 0, the embedded chain is at state (r_0, q_0). At the beginning of the next epoch the chain will be at any of the states $(\lambda_0, 0), (\lambda_0, 1), \ldots (\lambda_0, \min(r_0+q_0, kC))$ with non-zero probability. Suppose that the following control law is applied: For all (r_0, q_0), $\lambda_0 = \hat{\lambda}$, where $\hat{\lambda} > \mu$. Then, the queuing system will reach the state $(\hat{\lambda}, kC)$ within a finite number of epochs with nonzero probability. Also, at the next epoch, the state will be in any of the states $(\hat{\lambda}, 0), (\hat{\lambda}, 1), \ldots (\hat{\lambda}, kC))$ with nonzero probability. As before, from any of these states, the chain will reach the state $(\hat{\lambda}, kC)$ within a finite number of epochs with nonzero probability. Therefore, the state $(\hat{\lambda}, kC)$ is recurrent under this control law, and weak accessibility is satisfied.

Using a discount factor as in Equation (26) may not be appropriate, since the cost of an unutilized runway remains constant in time. An alternate formulation is to determine the average optimal cost per stage, c^*:

$$\begin{aligned} & c^* + h^*(r,q) = \\ & \min_{\lambda\in\Lambda}\Big\{(1-\textstyle\sum\beta_i-\sum\gamma_i)[\bar{c}(r,q)+\vec{p}_q(r,q)\cdot\vec{h}^*(\lambda)] \\ & \quad +\textstyle\sum\beta_i[\bar{c}(r+i,q)+\vec{p}_q(r+i,q)\cdot\vec{h}^*(\lambda-i)] \\ & \quad +\textstyle\sum\gamma_i[\bar{c}(r-i,q)+\vec{p}_q(r-i,q)\cdot\vec{h}^*(\lambda+i)]\Big\} \end{aligned} \tag{27}$$

IV. Application of the control policy at BOS

This section describes the application of PRC_v2.0, as derived in Equation (27) to the departure process at BOS. We focus on runway configuration (22L, 27 | 22L, 22R) under visual meteorological conditions (VMC) during the evening departure push. The control strategy is restricted to jet aircraft at BOS, for reasons explained in prior work [15].

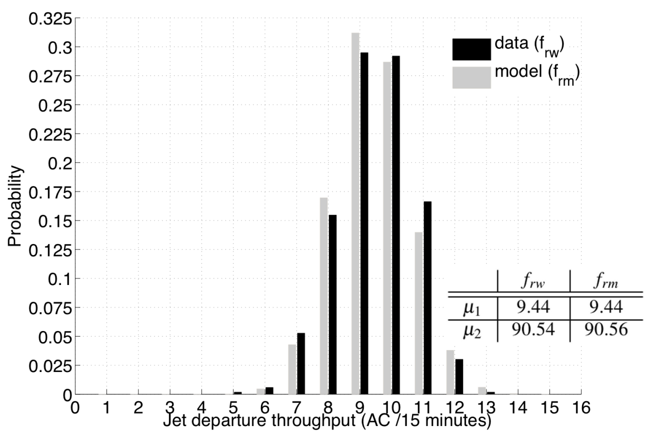
A. Selection of time period

The average unimpeded taxi-out time at BOS is 12.6 minutes under VMC [18]. There is an added delay due to taxiway congestion, which is proportional to the number of aircraft taxiing out [18], [19]. For non-excessive traffic levels, the additional average delay in the case of the BOS airport is 1-2 minutes. This makes 15 minutes a suitable choice of time-window for BOS. Furthermore, because of lack of accurate measurements [20], we assume that $\beta_i = \gamma_i = 0$ for all $i's$. Equation (27) then becomes:

$$c^* + h^*(r,q) = \min_{\lambda \in \Lambda} \{ (\bar{c}(r,q) + \vec{p}_q(r,q) \cdot \vec{h}^*(\lambda) \} \quad (28)$$

B. Estimation of the runway service process parameters

We are interested in estimating the parameters of the runway service process of the BOS airport during peak evening times. For this reason, we perform the analysis outlined in recent work [21] using ASDE-X data from November 2010-June 2011, and isolate 15-minute intervals during which the runway was under *continuous demand*. We obtain 1726 measurements of the runway throughput (departures/15 min) that provide an empirical distribution of the departure capacity. Figure 2 shows the resulting empirical distribution f_{rw} in black.



	f_{rw}	f_{rm}
μ_1	9.44	9.44
μ_2	90.54	90.56

Fig. 2: Empirical (f_{rw}) and modeled (f_{rm}) probability distributions (and first two moments) of the departure capacity of runway configuration 22L, 27|22L,22R under visual meteorological conditions during evening times.

We assume that the service times are generated from an Erlang distribution with parameters $(k, k\mu)$. We estimate these parameters using an approximation based on the method of moments. The output is the Poisson distribution satisfied by the kth arrivals of the exponential distribution with service rate $(k\mu)$ in a Δ time period, and that matches the first moment and has the smallest absolute error of the second moment of f_{rw}.

For the empirical distribution of Figure 2, we obtain the parameters of Erlang distribution (7, 4.6). The mean service time μ_2 is $7/4.6 = 1.5$ min. The variance of the service time is 0.3 min^2. The corresponding distribution, f_{rm}, of the number of takeoffs in Δ min is depicted in Figure 2 in grey. The empirical and modeled distributions are similar, as is also seen in the inset table.

C. Maximum pushback rate and cost function

The set of permissible policies is defined as $0, 1, \ldots, \hat{\lambda}$. At BOS, as in most airports, there is a natural threshold for the maximum admissible rate of arrivals into the departure process (pushbacks). At BOS, $\hat{\lambda}$ is calculated to be 15 jet aircraft/15 minutes, that is, $\Lambda = \{0, 1, \ldots, 15\}$. The space of the queuing system (C) is estimated to be 30, and the cost of underutilizing the runway, c(0), is chosen to be equal to the cost of a queue of 25 departures. $c(0)$ is chosen to reflect the fact that at BOS, a very long queue can lead to surface gridlock, and consequently, non-utilization of the runway.

D. Derivation of optimal policies

Given the service time distribution $(k, k\mu)$, the time period Δ, the queuing space C, the set Λ and the costs c, Equation (28) can be solved to obtain the optimal pushback policies. The efficient solution of Equation (28) is possible using the policy iteration method with a suitable choice of initial policy. In selecting initial policies, we use the insights that (1) For given q, the pushback policy is expected to be a non-decreasing function of r; (2) For given r, the pushback policy is expected to be a non-decreasing function of q; (3) The pushback policy is expected to target for a specific level of inventory (number of aircraft in the queue). We used a target inventory, $b_f = 5$ aircraft in the queue. For each state (r,q), the initial policy $\lambda_0(r,q)$ is calculated as:

$$\lceil \min(\mu_2 + b_f - \max(\max(r + \max(\lfloor (q-1)/k \rfloor, 0) - \mu_2), 0), \hat{\lambda}) \rceil$$

The policy iteration algorithm converges in fewer than 10 iterations. The optimal policies λ^* are a function of the state of the embedded chain (r,q), which is not observable. However, each state of the chain is mapped to an observed state of the process, N (Equation 22). For $0 \le T \le \hat{\lambda}$, the optimal pushback rate is approximated by:

$$\bar{\lambda}(T,0) = \lfloor \frac{\sum_{j=0}^{k} \lambda^*(T,j)}{k+1} + 0.5 \rfloor \quad (29)$$

$$\bar{\lambda}(T,D) = \lfloor \frac{\sum_{j=dk+1}^{(d+1)k} \lambda^*(T,j)}{k} + 0.5 \rfloor \quad \text{for } 1 \le D < C \quad (30)$$

Figure 3 shows the contours of the optimal pushback policy $\bar{\lambda}$ as a function of the number of aircraft in the departure queue (D) and the number of aircraft taxiing (T). As expected, the optimal pushback rates decrease for increasing D and T. A different way to characterize the optimal policies is to plot the expected work-in-process at the next epoch, $W_{\tau+\Delta} = T_{\tau+\Delta} + D_{\tau+\Delta}$, as a function of the current state (T_τ, D_τ), as shown in Figure 4. When $W_\tau = T_\tau + D_\tau \ge 12$, the policy attempts to control the expected work-in-progress at the next epoch to 13. When $W_\tau \ge 23$, the optimal pushback rate is 0, but it is not sufficient to reduce the expected $W_{\tau+\Delta}$ to 13. We also note that when $W_\tau \le 12$, the optimal pushback policy increases the expected $W_{\tau+\Delta}$ to values higher than 13.

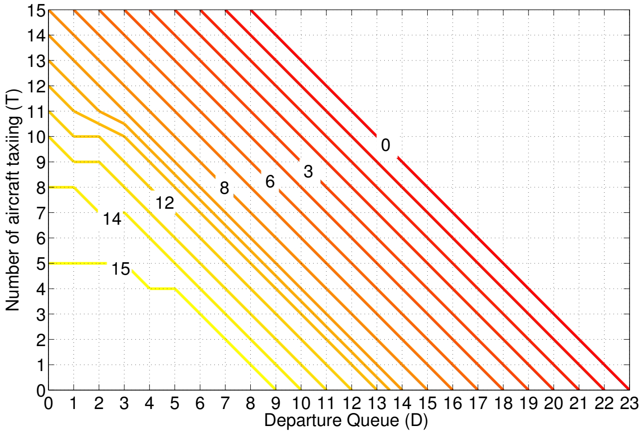


Fig. 3: Optimal pushback policy $\bar{\lambda}$ as a function of the number of aircraft in the departure queue (D) and the number of aircraft taxiing (T).

Figure 4 suggests that the algorithm aims at controlling the process to a desired value of W_τ. The expected $W_{\tau+\Delta}$ consists of the expected queue length at $\tau+\Delta$, $D_{\tau+\Delta}$ and the pushback rate $\bar{\lambda}_\tau$ set at at time τ (Equation 22). This implies that the optimal pushback policy at time τ, is a function of the expected queue length at time $\tau+\Delta$.

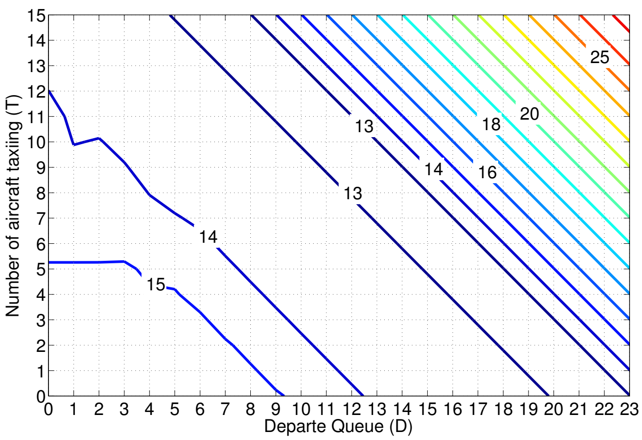


Fig. 4: Expected work-in-process, W, at the next epoch (τ_Δ) as a function of the number of aircraft in the departure queue (D_τ) and the number of aircraft taxiing (T_τ).

Figure 5 shows the scatterplot between the optimal pushback rate $\bar{\lambda}_\tau(T_\tau, D_\tau)$ and the expected $D_{\tau+\Delta}(T_\tau, D_\tau)$, for all $0 \leq T \leq \hat{\lambda}$ and $0 \leq D < kC$, along with a fitted convex non-increasing function that minimizes absolute deviations from the data. The equivalent PRC_v1.0 strategy, which aims at keeping $W_{\tau+\Delta}$ always at 13 irrespective of the state N_τ, is also shown. For the most part, the two strategies are the same after rounding to the closest integer. However, when the expected queue length at $\tau+\Delta$ is less than 4, the optimal pushback policy increases $W_{\tau+\Delta}$ to 14 or 15. In this region, the departure throughput can be increased with a high pushback rate at a very low congestion cost. Figure 5 also shows the benefit of the PRC_v1.0 strategy [15]. By simply aiming at a target $W_{\tau+\Delta}$ at the next epoch, the strategy is suboptimal only when the expected value of $W_{\tau+\Delta}$ is 1, 2 or 3. However, these are instances of high risk of runway non-utilization, and PRC_v2.0 accounts better for this risk.

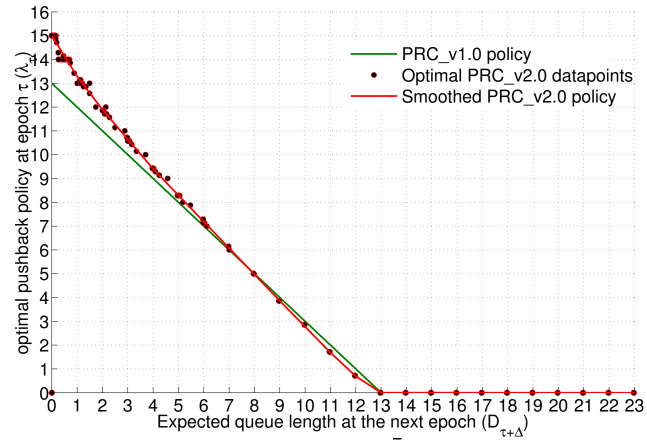


Fig. 5: Optimal pushback policy $\bar{\lambda}_\tau$ as a function of the expected *queue* D at the next epoch ($\tau+\Delta$).

To illustrate how the control algorithm would work in conjunction with the system dynamics described in Equation 20, we consider a sample path of the certainty equivalent system: At the first epoch ($t=0$), the state is $(0,0)$, that is, there are no aircraft on the ground. At the next epoch ($t=15$), the expected queue will be zero, and the curve of Figure 5 recommends that 15 aircraft pushback in the next 15 minutes (or a pushback rate of 1/ minute). Thus, $S_{15} = (15,0)$. Solving the Chapman-Kolmogorov equations numerically for the queuing model $(M(t)|R_\tau)/E_k/1$, we find that at the third epoch ($t=30$), the expected queue is 6. As a result, Figure 5 recommends a pushback rate of (7/15 minutes), so $S_{30} = (7,6)$. Similarly, $S_{45} = (10,4)$, $S_{60} = (5,8)$, $S_{75} = (10,4)$, etc. Therefore, after two cycles, the system stabilizes at a traffic level of 13-14 aircraft. We also note that the expected queue length at each epoch is at least 4. Finally, since the pushback rate is bounded at 15 aircraft/15 minutes, the traffic level can reach at most 24 aircraft: This happens in the extreme case in which the state is $(0,10)$, which implies $\bar{\lambda} = 14$, and no aircraft manages to takeoff. If this happens, due to an unpredicted runway closure for example, the next state is $(14,10)$ and the pushback rate is set to 0, as can be seen from Figure 3.

V. Field tests at BOS

The average cost-per-stage control algorithm PRC_v2.0 was adapted as follows, and tested at BOS.

A. Conditional capacity forecasts

Parameters such as the fleet mix and the expected number of landings in the next time period $(\tau, \tau+\Delta]$ can provide a conditional forecast for the runway service time distribution derived in Section IV-B [21]. These parameters explain some of the variance of the departure throughput and provide a more accurate estimate of the expected departure capacity. These conditional forecasts are incorporated into the algorithm in an approximate fashion:

- At epoch τ, use the conditional service time distributions for the time period $(\tau, \tau+\Delta]$ to calculate the expected queue length at the next epoch, $\tau+\Delta$.
- Use the PRC_v2.0 curve of Figure 5 to calculate the optimal pushback policy for this expected queue length.

This is a heuristic modification of PRC_v2.0 to incorporate the conditional forecasts given fleet mix and expected number of landings. We call this control protocol PRC_v2.1. This heuristic was chosen because of its simplicity and intuitiveness. An alternative would be to augment the state and include the service time forecast as a state variable.

The conditional service time distributions are characterized by different $(k, k\mu)$ parameters than the unconditional one. Figure 5 implies that the expected queue length at the next epoch ($Q_{\tau+\Delta}$) can be used as a quasi state for this system: Essentially, it is the post-decision state variable. Post-decision, from epoch $\tau+\Delta$ onwards, the system dynamics are accurately accounted for in the curve of Figure 5. Also, the expected queue, $Q_{\tau+\Delta}$ can be calculated solving the Chapman-Kolmogorov equations with the conditional service time distribution. Thus, the heuristic algorithm is expected to have near-optimal performance.

B. Results of field testing

During 8 four-hour test periods in 2011, fuel use was reduced by an estimated 9 US tons (2,650 US gallons), and aircraft taxi times decreased by an average of 5.3 min for the 144 flights that were held at the gate, showing that such a congestion control strategy could yield significant benefits. A detailed analysis of the field trials can be found is another paper by the authors [20].

VI. CONCLUSIONS

This paper presented a pushback rate control strategy for the reduction of taxi-out times, by formulating surface congestion management as a dynamic control problem. The runway queuing system was modeled as a semi-Markov process, and optimal pushback rates were determined. The final control policies accommodate the practical challenges of time-delay and current operational procedures.

The proposed pushback rate control refinement (PRC_v2.0) was adapted to Boston airport and field tested. Over five periods with significant congestion, the strategy was demonstrated to be effective in limiting congestion while maintaining runway utilization. Data from the test periods showed that the algorithm was successful in maintaining a desired level of demand on the surface, and that the behavior of the system (state, as well as departure throughput) was as predicted. Over 750 min of taxi-out time savings, and a corresponding 9-11 metric tonnes of reduction in jet fuel burn, are estimated from the use of the proposed control strategy during these periods.

ACKNOWLEDGMENTS

The authors thank Harshad Khadilkar, Melanie Sandberg, John Hansman and Tom Reynolds at MIT, Vivek Panyam, the BOS ATCT (Brendan Reilly), air carriers at BOS, and the FAA (Steve Urlass) for their help with the field tests. They also thank Eric Feron (Georgia Tech) and Richard Jordan (MIT Lincoln Labs) for helpful conversations.

REFERENCES

[1] Federal Aviation Administration, "Aviation System Performance Metrics," accessed September 2011, `http://aspm.faa.gov`.
[2] E. R. Feron, R. J. Hansman, A. R. Odoni, R. B. Cots, B. Delcaire, W. D. Hall, H. R. Idris, A. Muharremoglu, and N. Pujet, "The Departure Planner: A conceptual discussion," Massachusetts Institute of Technology, Tech. Rep., 1997.
[3] N. Pujet, B. Delcaire, and E. Feron, "Input-output modeling and control of the departure process of congested airports," *AIAA Guidance, Navigation, and Control Conference and Exhibit, Portland, OR*, pp. 1835–1852, 1999.
[4] F. Carr, "Stochastic modeling and control of airport surface traffic," Master's thesis, Massachusetts Institute of Technology, 2001.
[5] P. Burgain, "On the control of airport departure processes," Ph.D. dissertation, Georgia Institute of Technology, 2010.
[6] R. Clewlow and D. Michalek, "Logan control tower: Controller positions, processes, and decision support systems," Massachusetts Institute of Technology, Tech. Rep., 2010.
[7] M. Spearman and M. Zazanis, "Push and pull production systems: issues and comparisons," *Operations research*, pp. 521–532, 1992.
[8] P. Burgain, O. Pinon, E. Feron, J. Clarke, and D. Mavris, "On the value of information within a collaborative decision making framework for airport departure operations," in *Digital Avionics Systems Conference*. IEEE, 2009.
[9] S. Gershwin, "Design and operation of manufacturing systems: the control-point policy," *IIE Transactions*, vol. 32, no. 10, pp. 891–906, 2000.
[10] T. Crabill, D. Gross, and M. Magazine, "A classified bibliography of research on optimal design and control of queues," *Operations Research*, vol. 25, no. 2, pp. 219–232, 1977.
[11] S. Stidham and R. Weber, "A survey of Markov decision models for control of networks of queues," *Queueing Systems*, vol. 13, no. 1, pp. 291–314, 1993.
[12] S. Stidham Jr, "Analysis, design, and control of queueing systems," *Operations Research*, vol. 50, no. 1, pp. 197–216, 2002.
[13] J. Le Ny and H. Balakrishnan, "Feedback Control of the National Airspace System to Mitigate Weather Disruptions," in *Proceedings of the 49th IEEE Conference on Decision and Control*, 2010.
[14] S. Roy, B. Sridhar, and G. Verghese, "An aggregate dynamic stochastic model for an air traffic system," in *Proceedings of the 5th USA/Europe ATM 2003 R&D Seminar*, 2003.
[15] I. Simaiakis, H. Balakrishnan, H. Khadilkar, T. Reynolds, R. Hansman, B. Reilly, and S. Urlass, "Demonstration of reduced airport congestion through pushback rate control," in *9th USA-Europe Air Traffic Management Research and Development Seminar*, 2011.
[16] D. Low, "Optimal dynamic pricing policies for an M/M/s queue," *Operations Research*, vol. 22, no. 3, pp. 545–561, 1974.
[17] S. Lippman, "Applying a new device in the optimization of exponential queuing systems," *Operations Research*, vol. 23, no. 4, pp. 687–710, 1975.
[18] I. Simaiakis and H. Balakrishnan, "Queuing Models of Airport Departure Processes for Emissions Reduction," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2009.
[19] H. Khadilkar, "Analysis and modeling of airport surface operations," Master's thesis, Massachusetts Institute of Technology, 2011.
[20] I. Simaiakis, M. Sandberg, H. Balakrishnan, and R. J. Hansman, "Design, testing and evaluation of a pushback rate control strategy," in *ICRAT*, 2012, to appear.
[21] I. Simaiakis and H. Balakrishnan, "Departure throughput study for Boston Logan International Airport," Massachusetts Institute of Technology, Tech. Rep., 2011, No. ICAT-2011-1.