Dynamic Control of Airport Departures: Algorithm Development and Field Evaluation

Ioannis Simaiakis, Melanie Sandberg, and Hamsa Balakrishnan

***Abstract*—This paper proposes dynamic programming based algorithms for controlling the departure process at congested airports. These algorithms, called *Pushback Rate Control* protocols, predict the departure throughput of the airport, and recommend a rate at which to release aircraft from their gates in order to control congestion. The paper describes the design and field-testing of a variant of Pushback Rate Control at Boston airport in 2011, and the development of a decision-support tool for its implementation. The analysis of data from the field trials shows that during 8 four-hour test periods, fuel use was reduced by an estimated 9 US tons (2,650 US gallons), and taxi-out times were reduced by an average of 5.3 min for the 144 flights that were held at the gate.**

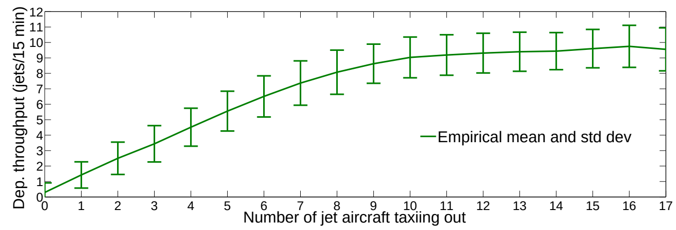


Fig. 1: Variation of departure throughput with the number of aircraft taxiing out, for the 22L, 27 | 22L, 22R configuration at BOS, under Visual Meteorological Conditions (VMC).

I. Introduction

A. Motivation and background

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]. Recent studies have also shown that low-thrust taxi emissions have significant impacts on the local air quality near major airports [2–4]. The objective of this paper is to develop, implement and evaluate a control policy that can reduce surface congestion.

B. 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. N-Control is an implementation of the virtual queue concept described in the Departure Planner [5], and variants of it have been extensively studied [6–9]. N-Control is based on the typical variation of departure throughput with the number of departures on the surface (denoted N): As more aircraft pushback from their gates onto the taxiways, the throughput of the departure runway initially increases. However, as the number of taxiing departures exceeds a threshold, denoted N^*, the departure runway capacity becomes the limiting factor, and there is no additional increase in throughput. Any additional aircraft that pushback simply incur taxi-out delays [10]. Figure 1 illustrates this behavior for the most frequently used runway configuration at Boston Logan International Airport (BOS) in 2011, using surface surveillance data from a system known as ASDE-X [11].

I. Simaiakis is with McKinsey & Company, Washington, D.C.
M. Sandberg is with MIT Lincoln Laboratory.
H. Balakrishnan is with the Massachusetts Institute of Technology. email: hamsa@mit.edu.
Manuscript received March 15, 2013.

The N-Control policy is effectively a threshold heuristic: If the total number of departing aircraft on the ground exceeds a certain threshold, N_{ctrl}, where $N_{\text{ctrl}} \geq N^*$, aircraft requesting pushback are held at their gates until the number of aircraft on the ground is less than N_{ctrl}. While the choice of N_{ctrl} must be large enough to maintain runway utilization, too large a value will be overly conservative and reduce the benefits of the control strategy. 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 [12]. The N-Control policy is also closely related to the *constant work-in-process* (CONWIP) policy used in manufacturing systems. The main benefits of CONWIP systems are their simplicity, implementability and controllability [13]. They present an efficient way to control congestion by accepting an adjustable risk of capacity loss.

Optimization-based policies have also been considered recently for surface traffic. Burgain et al. used advanced modeling tools for the characterization of optimal pushback policies. Their control protocol was a generalization of N-Control, where the state of the surface at any time was mapped to an *on-off* input signal. The solutions were full-state feedback policies, which presented implementation challenges [14].

There has been much prior research on the optimal control of a variety of queuing systems, considering different decision variables and control objectives [15–17]. However, several challenges remain when attempting to apply these results to the control of the airport 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 prefer a pushback rate that is valid for a predefined time period, after which it can be updated. Controllers prefer such pushback rate recommendations for workload and procedural reasons, and airlines prefer them because of their predictability, which helps with ground crew planning. Secondly, the control input is applied at the gates during pushback, whereas the main bottleneck is the runway. The control strategy has to accommodate stochastic travel times between the gate and the runway, due to factors such as the pushback process, flight checklists, communication delays and variable taxi speeds.

Several approaches to departure metering have been proposed, including the Ground Metering Program at New York's JFK airport [18], the field-tests of the Collaborative Departure Queue Management concept at Memphis (MEM) airport [19], the human-in-the-loop simulations of the Spot and Runway Departure Advisor (SARDA) concept at Dallas Fort Worth (DFW) airport [20], the trials of the Departure Manager (DMAN) concept [21] in Athens International airport (ATH) [22], and the field-tests of an N-control based heuristic, Pushback Rate Control (henceforth referred to as PRC_v1.0) [23]. However, none of these efforts have explicitly estimated the stochasticity of the underlying processes, or developed optimal control policies that account for the uncertainty.

Surface traffic control has also been formulated as an optimization problem minimizing a delay function [24–27]. These formulations are MILP problems that have been shown to be NP-hard [24, 28]. Practical implementations deterministically schedule a small number of operations, typically 20-30 flights at a time, using a rolling horizon. In other words, the solutions are open-loop policies subject to periodic reoptimization. They assume knowledge of the location and intent of all aircraft, no uncertainty, and the ability to instantaneously modify aircraft speeds; they are not applicable in the current operational environment. Receding horizon approaches have also been used for multi-airport capacity management [29].

In this paper, a queuing model is used for the prediction of the takeoff rate to derive two control algorithms, using dynamic programing and approximate dynamic programming (referred to as PRC_v2.0 and PRC_v2.1, respectively). The paper also presents the design of a decision support tool (DST) for air traffic controllers, and describes the field tests of PRC_v2.1 using the new DST at BOS in 2011. Finally, the proposed congestion control algorithm, the takeoff rate predictions, and the results of the field-tests are evaluated.

II. Control strategy design requirements

The objective of the control strategy is to minimize the number of aircraft taxiing out and thus taxi-out times, while still maintaining runway utilization. It needs to be compatible with currently available information, automation and operational procedures in the airport tower, and have a minimal impact on controller workload. It must also account for uncertainties in the taxi-out process.

As mentioned before, the preferred form of a congestion control strategy is one that recommends a pushback (release) rate to air traffic controllers [23] at the beginning of each time period and is periodically updated. In general, the length of the time period, Δ, should equal the lead time of the system, that is, the delay between the application of the control input (setting an arrival rate at 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 value of Δ that is approximately equal to the travel time from the gates to the runway, the flights released from the gate during a given time period are expected to reach the departure queue in the next time period.

Careful monitoring of off-nominal events and constraints is also necessary for implementation at a particular site. Particular concerns are gate conflicts (for example, an arriving aircraft is assigned the same gate as a departure that is being held) and the ability to meet controlled departure times (Expected Departure Clearance Times or EDCTs) and other traffic management constraints. In consultation with the BOS Tower, flights with EDCTs were handled as usual and released First-Come-First-Served. Pushbacks were expedited to accommodate arrivals if needed. Finally, since departures of propellor-driven aircraft (props) were known not to significantly affect jet departures [30], props were exempt from Pushback Rate Control.

III. Departure process model

A. State variables

At the beginning of each time-window (called an *epoch*), the state of the airport system was observed. The takeoff rate in this time-window and the state of the departure process at the beginning of the subsequent time-window were predicted and used to recommend a pushback rate. For the purposes of control, the state was derived from the following inputs:

1) Meteorological conditions and runway configuration, $(MC;RC)$.
2) Number of jet aircraft traveling from the gates to the departure runway, G.
3) Number of jet aircraft in the departure queue, D.
4) Expected number of arrivals in the next 15 min, A.
5) Number of props taxiing out, P.

The above quantities are all known in the current tower environment: G is the number of jet flight strips in the ground controller's position, D is the number of jet flight strips in the local controller's position, P is known from the same positions, and A is given by the Traffic Situation Display (TSD).

Given the meteorological conditions and runway configuration $(MC;RC)$ at any time t, the state N_t of the departure process consists of the number of jet aircraft traveling from the gates to the departure queue (G_t) and the number of aircraft in the departure queue (D_t):

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

Total number of aircraft taxiing out, also known as the total work-in-process of the departure process, W_t, is given by

$$W_t = G_t + D_t. \tag{2}$$

B. Selection of time period

The value of Δ should equal the *lead time* of the system, that is, the delay between the application of the control input (setting an arrival rate at the runway by controlling the pushback rate) and the time at which the runway sees that arrival rate. By choosing a time horizon that is approximately equal to the expected travel time from the gate to the departure queue, flights pushing back during a given time period will reach the departure queue in the next time period.

C. Pushback process

At each epoch (the beginning of each time period), the decision maker chooses a pushback rate (arrival rate into the surface system), $\lambda \in \Lambda = \{0, 1, \cdots, \lambda_{max}\}$. λ is expressed as the number of pushbacks per Δ minutes. By setting a pushback rate at epoch τ, the air traffic controller 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.

D. Runway service process

Each departure runway can be modeled as a single server at which aircraft queue to await takeoff. For the case of BOS, the departure runways of each configuration are modeled as a single server with a single queue, as departure runways of each major configuration are not independent and are fed by the same queue [23]. This approach has been extended to the case of multiple departure runways as well [31]. The runway system has finite queuing space C, which depends on the airport layout and operational procedures. The runway service times are modeled as being Erlang distributed, with shape and rate $(k, k\mu)$ extracted from surveillance (ASDE-X) data [32]. The arrival times at the queuing system are assumed to be random and independent from each other. At each epoch, the total number or aircraft traveling from the gate to the departure queue is known (denoted R_τ); it is assumed that all of them will have reached the runway server in the next epoch. This assumption will be relaxed later in Section IV.

This departure runway resembles a $M(t)/E_k/1$ system with queuing space C, with the additional constraint that there are R_τ arrivals during the $(\tau, \tau+\Delta]$ time interval. Such a system is denoted as $(M(t)|R_\tau)/E_k/1$. Assuming that at epoch τ there are R_τ aircraft traveling to the departure runway, the probability density function g of the r^{th} arrival at the departure runway at time $t \in (\tau, \tau+\Delta]$ is:

$$g(r,t) = \frac{R_\tau - (r-1)}{(\tau+\Delta) - t},\ t \in (\tau, \tau+\Delta],\ r = 0, 1, \ldots R_\tau \qquad (3)$$

(3) is derived by considering $R_\tau - (r-1)$ uniformly distributed random variables in the interval $(t, \tau+\Delta]$. The probability that one of these lies in $(t, t+dt]$ is $\big(R_\tau - (r-1)\big)\mathrm{d}t/(\tau+\Delta-t)$.

The state of the queuing system at time t is denoted by

$$S_t = (R_t, Q_t) \qquad (4)$$

where R_t is the number of aircraft that were traveling to the departure runway 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 2.

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 $k\mu$. Each such stage is known as a *stage of work*. A state of the Markov chain (r, q) implies that there are r aircraft that have been traveling to the runway since the start of that epoch, and there are q *stages of work* to be completed at the departure runway server. In other words, there are $\min(1, q)$ aircraft in service and $\max(\lceil (q-k)/k \rceil, 0)$ aircraft in the departure queue.

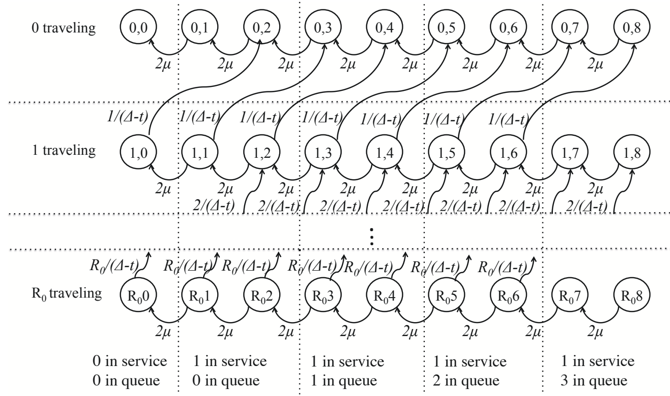


Fig. 2: State transition diagram for an $(M(t)|R_0)/E_2/1$ system with queuing space of 4 customers in the system.

At epoch 0, the Markov chain is at the bottom level of Figure 2 in state (R_0, Q_0), namely, R_0 aircraft traveling to the departure runway and Q_0 stages of work to be completed. By the end of epoch Δ, all R_0 aircraft will have reached the departure queue, and the Markov chain will be at the top level (0 aircraft traveling). 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 the state of the queuing system at the end of the time interval Δ, and can be determined by considering the possible transitions of the Markov chain (for example, in Figure 2). The resultant set of equations are known as the *Chapman-Kolmogorov equations* [33]. For $0 < t \leq \Delta$, and $1 \leq r < R_0$:

$$\frac{dP_{0,0}}{dt} = k\mu P_{0,1} \qquad (5)$$

$$\frac{dP_{0,q}}{dt} = k\mu P_{0,q+1} - k\mu P_{0,q}, \quad 1 \leq q < k \qquad (6)$$

$$\frac{dP_{0,q}}{dt} = k\mu P_{0,q+1} + \frac{1}{\Delta - t} P_{1,q-k} - k\mu P_{0,q},\ k \leq q < kC \qquad (7)$$

$$\frac{dP_{0,kC}}{dt} = \frac{1}{\Delta - t} P_{1,k(C-1)} - k\mu P_{0,kC} \qquad (8)$$

$$\frac{dP_{r,0}}{dt} = k\mu P_{r,1} - \frac{r}{\Delta - t} P_{r,0} \qquad (9)$$

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

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

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

$$\frac{dP_{r,kC}}{dt} = \frac{r+1}{\Delta - t} P_{r+1,k(C-1))} - k\mu P_{r,kC} \qquad (13)$$

$$\frac{dP_{R_0,0}}{dt} = k\mu P_{R_0,1} - \frac{R_0}{\Delta - t} P_{R_0,0} \qquad (14)$$

$$\frac{dP_{R_0,q}}{dt} = k\mu P_{R_0,q+1} - \left(\frac{R_0}{\Delta - t} - k\mu\right) P_{R_0,q}, \quad 1 \leq q \leq k(C-1) \qquad (15)$$

$$\begin{aligned}\frac{dP_{R_0,q}}{dt} &= k\mu P_{R_0,q+1} - k\mu P_{R_0,q},\, k(C-1) < q < kC && (16)\\ \frac{dP_{R_0,kC}}{dt} &= -k\mu P_{R_0,kC} && (17)\end{aligned}$$

The state probabilities $P_{0,0}(\Delta), P_{0,1}(\Delta), ...P_{0,kC}(\Delta)$ are obtained by numerically solving (5)-(17) for $t = \Delta$, with initial value (R_0, Q_0). The probability of the queuing system state at time Δ being $i = Q_\Delta$ is given by

$$\begin{aligned} Q_\Delta &= f(R_0, Q_0)\\ \text{with } p_{q(i)}(R_0, Q_0) &= P_{0,i}(\Delta) \quad \text{for } 0 \le i \le kC\\ \Longrightarrow \quad \mathbf{p}_q(R_0, Q_0) &= \mathbf{P}_0(\Delta) \triangleq [P_{0,0}(\Delta), P_{0,1}(\Delta), ...P_{0,kC}(\Delta)]'.\end{aligned}$$

E. System dynamics

Suppose that at epoch τ, there are R_τ aircraft traveling to the departure runway, Q_τ stages of work left 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 traveling, and $Q_{\tau+\Delta} = f(R_\tau, Q_\tau)$ stages of work will remain in the queue. The queuing system therefore evolves according to the following equation:

$$(R_{\tau+\Delta}, Q_{\tau+\Delta}) = (\lambda_\tau, f(R_\tau, Q_\tau)) \quad (18)$$

Given that the chain is in state (r,q) at the epoch τ and the pushback rate λ is chosen, the probability that the chain is in state (i, j) at the next epoch $\tau+\Delta$ is:

$$\Pr_{(r,q)\to(i,j)}(\lambda) = \begin{cases} p_{q(j)}(r,q) & \text{if } i = \lambda\\ 0 & \text{otherwise}\end{cases} \quad (19)$$

The state S_t of the queuing system maps to the state N_t of the departure process as follows:

$$N_t = \begin{cases} \big(\lambda_{t-\Delta}, \max(\lceil (Q_t-k)/k\rceil, 0)\big), & t \in \{0, \Delta, \dots\}\\ \big(V_t + R_t, \max(\lceil (Q_t-k)/k\rceil, 0)\big), & \text{otherwise}\end{cases} \quad (20)$$

where V_t is the number of aircraft that pushed back between the start of the time period in which t lies, and time t. By sampling the system every Δ minutes, the departure process is decoupled into the pushback and the runway service processes, which are independent of each other within a time period.

F. Choice of cost function

The control strategy sets the arrival rate of aircraft to the queuing system, namely, the pushback rate, to balance two objectives: Minimize the expected departure queue length, and maximize the runway utilization. The cost function associated with a state (r,q) of the queuing system is denoted $c(q)$. This cost is a combination of the queuing cost and the cost of non-utilization of the runway, that is, $q = 0$. 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 which is usually assumed to be a monotonically nondecreasing function of q with increasing marginal costs [34, 35]. It is assumed to scale quadratically with the state of the queue, because the expected system delay is a quadratic function of queuing state ($[D\cdot(D+1)/2]/\mu$). A candidate cost function with these properties is:

$$c(q) = \begin{cases} H, & q = 0\\ (\lceil (q-k)/k\rceil)^2 & q = 1, \dots, kC\end{cases} \quad (21)$$

where H is the cost of a loss of runway utilization.

Equations (5)-(17) are solved numerically to calculate

$$\mathbf{p}_q(R_0, Q_0, t) = \Big[\sum_{r=0}^{R_0} P_{r,0}(t), \sum_{r=0}^{R_0} P_{r,1}(t), \dots, \sum_{r=0}^{R_0} P_{r,kC}(t)\Big]' \quad (22)$$

at time t. Numerical experiments show that sampling 10 times a minute is sufficient for accurately calculating the expected cost of each state, $\bar{c}$, over Δ min [32]:

$$\bar{c}(R_0, Q_0) = \sum_{i=0}^{10\Delta-1} \frac{1}{10}\mathbf{p}_q(R_0, Q_0, i/10)\cdot \mathbf{c}(Q_0) \quad (23)$$

IV. Dynamic programing formulation

The optimal costs, $J^*(r,q)$, at each state, (r,q), for the infinite horizon problem with discount factor α are given by the Bellman equation [36]:

$$\begin{aligned} J^*(r,q) &= \min_{\lambda\in\Lambda}\{\bar{c}(r,q) + \alpha\sum_{j=0}^{kC}\Pr_{(r,q)\to(\lambda,j)} J^*(\lambda, j)\}\\ &= \min_{\lambda\in\Lambda}\{\bar{c}(r,q) + \alpha \mathbf{p}_q(r,q)\cdot \mathbf{J}^*(\lambda)\end{aligned}$$

where $\mathbf{J}^*(\lambda) = [J^*(\lambda,0), J^*(\lambda,1), \dots, J^*(\lambda,kC)]'$, $r \in \{0, 1, \dots, \lambda_{max}\}$ and $q \in \{0, 1, \dots, kC\}$.

The earlier assumptions that the R_τ aircraft traveling at epoch τ will all reach the queue during the time interval $(\tau, \tau+\Delta]$ and that the pushback rate (λ_τ) set at epoch τ will arrive at the runway at $t > \tau+\Delta$ are now relaxed. For each value of λ_τ and R_τ, i out of the λ_τ aircraft are assumed to reach the runway during the time interval $(\tau, \tau+\Delta]$ with probability β_i. Similarly, i out of the R_τ aircraft are assumed to reach the runway at $t > \tau+\Delta$ with probability γ_i. Therefore, R_τ aircraft reach the runway during the time interval $(\tau, \tau+\Delta]$, and λ_τ aircraft at $t > \tau+\Delta$, with probability $1 - \sum\beta_i - \sum\gamma_i$.

Equation (18) therefore becomes:

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

The above equation is seen to maintain the Markov property. For these system dynamics, the Bellman equation for the infinite horizon problem with discount factor α is:

$$J^*(r,q) = \min_{\lambda\in\Lambda}\Big\{(1 - \textstyle\sum\beta_i - \sum\gamma_i)[\bar{c}(r,q) + \alpha\mathbf{p}_q(r,q)\cdot\mathbf{J}^*(\lambda)] + \sum\beta_i[\bar{c}(r+i,q) + \alpha\mathbf{p}_q(r+i,q)\cdot\mathbf{J}^*(\lambda-i)] + \sum\gamma_i[\bar{c}(r-i,q) + \alpha\mathbf{p}_q(r-i,q)\cdot\mathbf{J}^*(\lambda+i)]\Big\} \quad (24)$$

Equation (24) illustrates the tradeoffs involved with the choice of time period, Δ. A long time period requires less frequent updates of the optimal policy, making implementation easier. It is, however, necessary to predict runway performance and maintain runway utilization over a longer period of time. If Δ is significantly less than the lead time, a smaller inventory will be necessary at the departure queue to maintain runway utilization. However, γ_i will be large, and only a fraction of aircraft taxiing will arrive at the runway by the next epoch. As a result, the state at epoch τ, (R_τ, Q_τ), will not be closely related to the queuing system state at the next epoch, $Q_{\tau+\Delta}$.

More frequent updates of the optimal policy will also be necessary, increasing air traffic controller workload.

This problem can be shown to satisfy the property of weak accessibility. Suppose that at epoch 0, the embedded chain is at state (r_0, q_0). At 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 nonzero probability. Suppose that the following control law is applied: For all (r_0, q_0), $\lambda_0 = \lambda_{max}$, where $\lambda_{max} > \mu$. Then, the queuing system will reach the state (λ_{max}, kC) within a finite number of epochs with nonzero probability. Also, at the next epoch, the state will be in any of the states $(\lambda_{max}, 0), (\lambda_{max}, 1), \ldots (\lambda_{max}, kC))$ with nonzero probability. As before, from any of these states, the chain will reach the state (λ_{max}, kC) within a finite number of epochs with nonzero probability. The state (λ_{max}, kC) is therefore recurrent under this control law, and weak accessibility is satisfied.

Using a discount factor as in (24) may not be appropriate, since the cost of an unutilized runway remains constant in time. An alternate formulation is to determine the policies that minimize the average optimal cost per stage, c^*:

$$c^* + h^*(r,q) = \min_{\lambda \in \Lambda} \left\{ \begin{aligned} &(1 - \sum \beta_i - \sum \gamma_i)[\bar{c}(r,q) + \mathbf{p}_q(r,q) \cdot \mathbf{h}^*(\lambda)] \\ &+ \sum \beta_i [\bar{c}(r+i,q) + \mathbf{p}_q(r+i,q) \cdot \mathbf{h}^*(\lambda - i)] \\ &+ \sum \gamma_i [\bar{c}(r-i,q) + \mathbf{p}_q(r-i,q) \cdot \mathbf{h}^*(\lambda + i)] \end{aligned} \right\} \tag{25}$$

V. A Pushback Rate Control policy for BOS

This section describes the application of PRC_v2.0, given by Equation (25), to the departure process at BOS. The focus here is on runway configuration 22L, 27 | 22L, 22R in VMC during the peak evening departure push, although other configurations were also studied. As explained in Section II, props at BOS were exempt from pushback control.

A. Selection of time period

The average unimpeded taxi-out time at BOS is 12.6 minutes under VMC [10]. There is an added delay due to taxiway congestion, which is 1-2 minutes under moderate traffic conditions. 15 minutes is therefore a suitable choice of time-window at BOS. Furthermore, due to a lack of accurate measurements, it is assumed that $\beta_i = \gamma_i = 0$ for all i. Equation (25) therefore becomes:

$$c^* + h^*(r,q) = \min_{\lambda \in \Lambda} \left\{ (\bar{c}(r,q) + \mathbf{p_q}(r,q) \cdot \mathbf{h}^*(\lambda) \right\} \tag{26}$$

B. Estimation of runway service process parameters

The parameters of the runway service process of BOS during peak evening times were extracted using ASDE-X data from Nov 2010-Jun 2011. An Erlang distribution was fitted using the approximate Method of Moments. The mean service time was 1.54 min, and the variance was 0.47 min^2 [32].

C. Maximum pushback rate and cost function

The set of permissible policies is defined as $\Lambda = \{0, 1, \ldots, \lambda_{max}\}$. At most airports, there is a natural threshold for the maximum admissible rate of arrivals into the departure process (pushbacks). λ_{max} is estimated to be 15 aircraft/15 min, and the queuing system capacity (C) is estimated to be 30 at BOS. The cost of underutilizing the runway, c(0), is chosen to be equal to the cost of a queue of 25 departures, reflecting the fact that at BOS, a very long queue can lead to surface gridlock and non-utilization of the runway [23].

D. Calculation 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, the optimal pushback policies can be obtained by solving (26). It can be solved efficiently using policy iteration with a suitable choice of initial policy. 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 quantity, N, through (20). For $0 \leq G \leq \lambda_{max}$, the optimal pushback rate is approximated by:

$$\bar{\lambda}(G,0) = \lfloor \frac{\sum_{j=0}^{k} \lambda^*(G,j)}{k+1} + 0.5 \rfloor \tag{27}$$

$$\bar{\lambda}(G,D) = \lfloor \frac{\sum_{j=Dk+1}^{(D+1)k} \lambda^*(G,j)}{k} + 0.5 \rfloor \quad \text{for } 1 \leq D < C \tag{28}$$

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 traveling to the runway (G). As expected, the optimal pushback rates decrease for increasing D and G. The optimal policies can also be characterized by the expected work-in-process at the next epoch, $\bar{W}_{\tau+\Delta}$, as a function of the current state using Equation (18), as shown in Figure 4. When $W_\tau \geq 23$, the optimal pushback rate is 0, but it is not sufficient to reduce $\bar{W}_{\tau+\Delta}$ to 13. By contrast, when $W_\tau \leq 13$, the optimal pushback policy increases $\bar{W}_{\tau+\Delta}$ to values higher than 13.

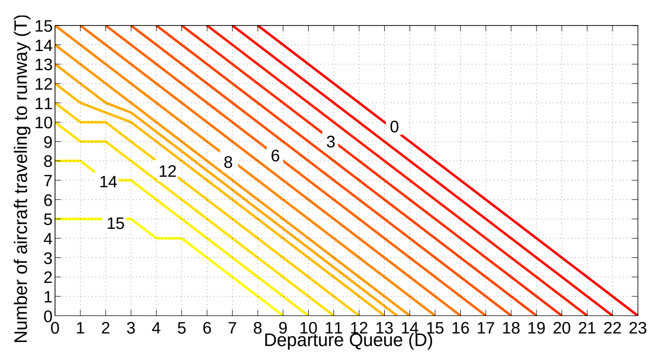


Fig. 3: Optimal pushback policy $\bar{\lambda}_\tau$ as a function of the number of aircraft in the departure queue (D_τ) and the number of aircraft traveling to the runway (G_τ).

Figure 4 suggests that the algorithm tries to target a desired value of W_τ. A comparison of Figures 3 and 4 shows that each $\bar{\lambda}_\tau$ is associated with one value of $\bar{W}_{\tau+\Delta}$ or $\bar{D}_{\tau+\Delta}$, suggesting

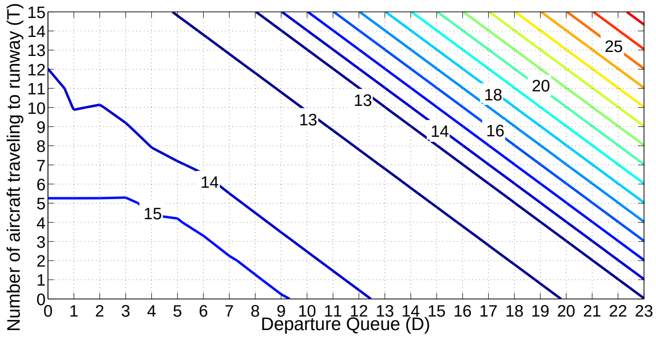


Fig. 4: Expected work-in-process as a function of the number of aircraft in the departure queue (D_τ) and the number of aircraft traveling to the runway (G_τ).

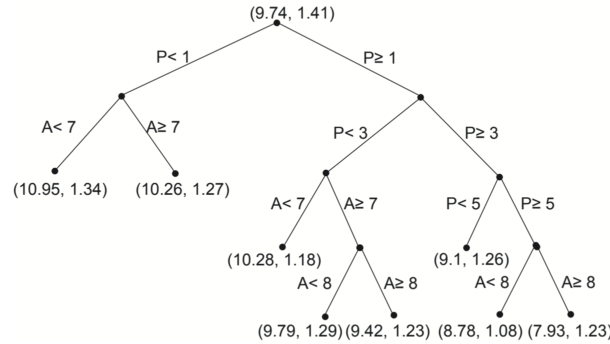


Fig. 6: Predicted jet departure throughput (mean, standard deviation) given the expected number of arrivals (A) in the next 15 minutes and number of props taxiing out (P).

that the optimal pushback policy at time τ is a function of the expected queue length at time $\tau+\Delta$.

Figure 5 shows the scatterplot of the optimal pushback rate $\bar{\lambda}_\tau(G_\tau, D_\tau)$ as a function of the expected $\bar{D}_{\tau+\Delta}(G_\tau, D_\tau)$ for all $0 \le G \le \lambda_{max}$ and $0 \le D \le C$, along with a fitted convex non-increasing function that minimizes absolute deviations from the calculated points. The equivalent PRC_v1.0 strategy which aims at keeping $W_{\tau+\Delta}$ at 13 aircraft is also shown [23]. 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, PRC_v2.0 increases $W_{\tau+\Delta}$ to 14 or 15 in order to better account for the risk of runway non-utilization.

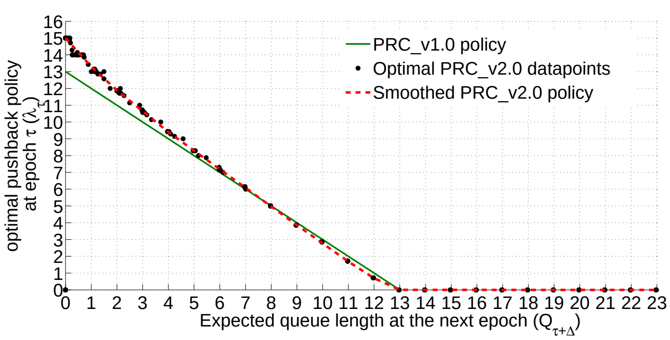


Fig. 5: Optimal pushback policy $\bar{\lambda}_\tau$ as a function of the expected *queue* $\bar{D}_{\tau+\Delta}$ at the next epoch $(\tau+\Delta)$.

E. Conditional throughput forecasts

Parameters such as the fleet mix and the expected number of landings can provide a conditional forecast for the runway service time distribution [30, 37]. These parameters explain some of the variance of the departure throughput, and provide a better estimate of the expected departure capacity. For example, the departure throughput in a given 15-min interval for runway configuration 22L, 27 | 22L, 22R in evening periods under visual meteorological conditions, given the arrival throughput and taxiing prop departures, can be estimated from the regression tree shown in Figure 6. This regression tree was validated using 10-fold cross validation.

These conditional forecasts are incorporated into the algorithm as follows:

- At epoch τ, the conditional throughput for the time window $(\tau, \tau+\Delta]$ is predicted from the regression tree, using the expected number of arrivals (A) and the number of props taxiing out (P).
- The expected takeoff rate in the time window $(\tau, \tau+\Delta]$ and queue length at $\tau+\Delta$ are calculated using a $(M(t)|R_\tau)/E_k/1$ queuing model with parameters fitted to the throughput forecast from the previous step.
- The PRC_v2.0 curve (Figure 5) is used to calculate the optimal pushback policy for this expected queue length.

The proposed approach, denoted PRC_v2.1, is a heuristic modification of PRC_v2.0 in the spirit of roll-out algorithms [38] to incorporate the conditional forecast. The intuition behind the derivation of PRC_v2.1 is that the conditional forecasts are used only to update the prediction of the queue length. While the reduction in the variance of the capacity distribution could yield a more aggressive control policy, this feature is not exploited. The optimal PRC_v2.1 policy at epoch τ is denoted $\check{\lambda}$, and is a function of the departure queue, the number of aircraft traveling to the runway and the number of props taxiing out at τ, as well as the expected number of landings in $(\tau, \tau+\Delta]$.

F. Rounding of optimal policies

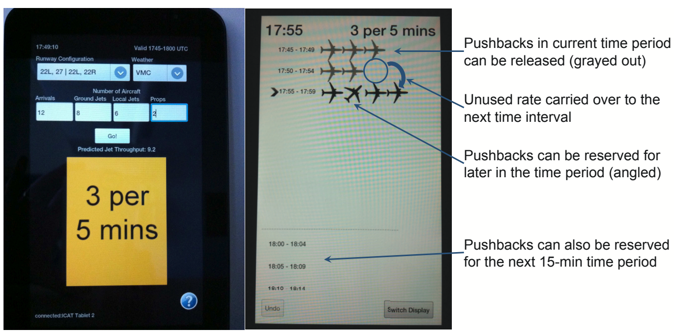
As explained in Section II, the optimal policy needs to be communicated to the controllers as a recommended rate. The optimal pushback rate for each 15-minute time-period is therefore rounded to one of the following: 0 aircraft/15 min (Stop), 1 aircraft/5 min, 1 aircraft/3 min, 2 aircraft/5 min, 1 aircraft/2 min, 3 aircraft/5 min, 2 aircraft/3 min, 4 aircraft/5 min or 1 aircraft/min [23].

VI. Design of a Decision Support Tool

A Decision Support Tool (DST) was designed in order to implement Pushback Rate Control algorithms, and PRC_v2.1 particular, in the airport tower environment. The device used was a 7" Samsung Galaxy Tab™ tablet computer with the Android™ operating system, which is convenient for application development, while being compact and portable. Two tablet computers were used in the implementation, namely, the rate control transmitter and the rate control receiver. Inputs were entered into the rate control transmitter, which then determined the optimal pushback rate and communicated it via a Bluetooth wireless link to the rate control receiver. The receiver displayed the recommended rate to the Boston Gate (BG) controller, who authorized aircraft to pushback.

A. Inputs

The inputs to the rate control transmitter were the runway configuration, meteorological conditions, expected number of arrivals in the next 15 minutes, numbers of jets under ground control and local control and number of props taxiing out. The input interface is shown in Figure 7b. The expected takeoff rate and the recommended pushback rate were then calculated using a look-up table for the PRC_v2.1 algorithm, and transmitted to rate control receiver.



(a) Input interface. (b) Volume control display mode.

Fig. 7: Design of the DST input and output interface.

B. Outputs

The receiver conveyed the suggested pushback rate to the BG controller through one of two display modes: the rate control and the volume control displays.

1) Rate control display: The output in this mode was a color-coded image of the suggested pushback rate. In this display mode, the BG controller kept track of the time intervals and the number of aircraft that have already pushed back. When the demand for pushbacks exceeded the recommended rate, aircraft were held at the gate until the next time interval. The BG controller kept track of aircraft holds, and released them at the appropriate time.

2) Volume control display: This display mode helped the BG controller keep track of the number of aircraft that had called and had been released. It was an alternative to the handwritten notes that controllers otherwise used to keep track of gate-holds. The volume control mode also provided visual cues of the timeline and upcoming actions.

On the volume control display, a 15-minute time period is broken down into smaller time intervals. For example, if the rate is 3 per 5 minutes, the display shows three rows of three aircraft icons, with each row corresponding to a 5-minute time interval (illustrated in Figure 7a). Aircraft can only be released during an ongoing time interval, indicated by a small black arrow to the left of it. Future positions can only be reserved. Any unused release spots roll over to the next time interval.

C. DST deployment

During the field trials at BOS in 2011, a member of the research team gathered and input data into the rate control transmitter. The rate control receiver was located next to the BG controller, who chose between the rate control and volume control displays. It is expected that in a long-term deployment, the traffic management coordinator (TMC) or the tower supervisor would input the data. For a part of the field-tests, the BG position was merged with another position, either clearance delivery or the TMC to investigate the potential implementation of PRC without requiring an additional controller at the BG position.

VII. Field-tests at BOS

PRC_v2.1 was field-tested at BOS during 19 evening (4PM-8PM) demo periods between July 18th and September 11th 2011. However, there was little congestion when the airport operated in its optimal configuration (4L, 4R | 4L, 4R, 9) or when the demand was low. There was enough congestion to warrant gate-holds in only eight of the demo periods. A total of 144 flights were held, with an average gate-hold of 5.3 min. During the most congested periods, up to 44% of flights experienced gate-holds.

This section presents an example describing one test period (July 21, 2011), the methodology used to calculate taxi-out time and fuel burn savings, and a comparison of the predicted and observed takeoff rates during the test periods.

A. An illustrative example from the field-tests

This section illustrates the typical outcomes of the PRC_v2.1 field-tests using a case day with significant gate-holds (July 21, 2011). Figure 8 depicts the events of the demo period, divided into 15-minute windows. The top plot shows the demand for pushbacks (that is, the number of aircraft that called for pushback), the number of pushbacks that were cleared, and the resulting number of jet aircraft actively taxiing out. The middle plot shows the predicted and measured throughput. Finally, the bottom plot shows the average taxi-out times and gate-holding times for aircraft that pushed back in each time interval.

The top plot in Figure 8 shows that as the number of jet aircraft taxiing-out exceeds 14, gate-holds are initiated in order to regulate the traffic to the desired state. For this configuration, the desired state is 13-14 aircraft on the surface. The airport stays in the desired state despite the high variance of the departure throughput (middle plot of Figure 8) and the rounding-off of the recommended pushback rates.

A key objective of the field-test was to maintain pressure on the departure runways, while limiting surface congestion. By maintaining runway utilization, it is reasonable to expect that gate-hold times translate to taxi-out time reduction. Runway utilization was shown to not be adversely impacted by the control strategy by checking that there was always at least one aircraft in the runway queue during the demo periods. This validation was performed both through visual observations and through the analysis of ASDE-X data [39].

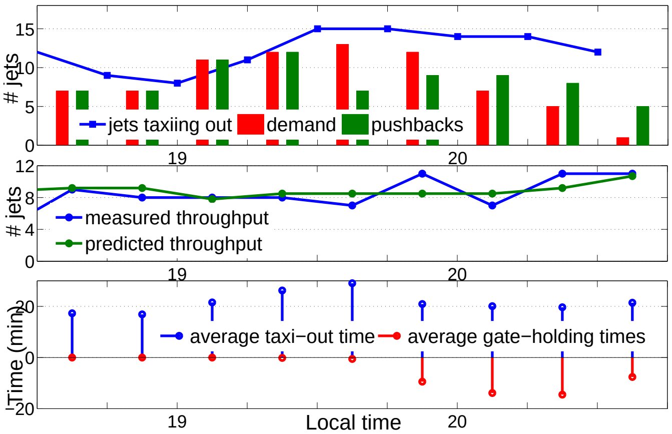


Fig. 8: (Top) Surface congestion, demand and pushbacks; (Middle) Departure throughput measurements and predictions; and (Bottom) Average taxi-out times and gate holds, during each 15-min interval on July 21, 2011.

B. *Translating gate-hold times to taxi-out time reductions*

The next step is to quantify the benefits achieved during the field-tests of PRC_v2.1. Taxi-out time and fuel burn reductions are the main benefits addressed in this paper. As long as runway throughput is maintained, it is reasonable to use the gate-hold times as surrogates for taxi-out time savings. Simulations of departure operations are used to confirm the above hypothesis, estimate the taxi-time savings and investigate the fairness of the control strategy.

The outcomes of three scenarios are compared in this analysis:

1) **Data from actual operations:** Taxi-out times and queuing times during demo periods are measured using ASDE-X data. They describe the actual system behavior during the field-tests.
2) **Simulations of actual operations:** These simulations are compared to the observed data in order to validate the simulation model. Pushback clearance times and gate-hold times for the simulation are obtained from observations during the demo periods, and correspond to the actual values.
3) **Simulations of hypothetical scenarios with no pushback rate control:** Finally, the model is used to simulate what would have happened if pushback rate control was not in effect, that is, if flights had been cleared for pushback as soon as they called ready to push. The pushback clearance times in the simulations are set to be equal to the call-ready times, that is, all gate-hold times are set to zero.

1) Simulation set-up: The following assumptions are made in all simulations: (1) The runway takeoff slots are determined by actual operations, to reflect differences in runway performance due to factors not related to pushback rate control. Fixed departure slots are a reasonable assumption as long as there is a nonzero departure runway queue. (2) Fights with traffic management constraints are assumed to have the same takeoff times as those observed in the data.

The unimpeded taxi-out times of flights are determined using ASDE-X data [32]. Given the pushback clearance time, the sum of the unimpeded taxi-out time and a taxi-out delay due to congestion [32], each flight is propagated to the runway, where it is assigned to the next available takeoff slot in that time period. The difference between this wheels-off time and the pushback clearance time is the expected taxi-out time. The comparison of the actual and predicted runway queuing times helps validate the simulations and assess the impact of the control strategy.

Table I presents detailed simulation results for two days with significant gateholds, Jul 21, 2011 and Jul 22, 2011. The results correspond to flights that were cleared for pushback between 1645 and 2045 hours. The table shows that the model predictions of the mean taxi-out and queuing times match the observed data from the actual operations.

TABLE I: Gate-hold times, mean taxi-times and queuing times from actual data, simulations of actual operations, and simulations of hypothetical operations with no gate-holds.

Date	# Flts	Gatehold time (min)	Scenario	Avg. taxi-out time (min)	Avg. queue time (min)
7/21	121	368	Act. data	16.5	5.7
			Act. sim.	16.5	5.8
			Hyp. sim.	19.5	7.9
7/22	121	279	Act. data	17.9	7.2
			Act. sim.	17.9	7.4
			Hyp. sim.	20.2	9.2

Figure 9 (top) shows the instantaneous actual and simulated queue on July 21, 2011. The actual queue is seen to be accurately predicted by the simulations. The bottom plot on the same figure shows the actual queue on July 21 with PRC, and the simulated prediction without PRC. The difference between the two queue lengths illustrates the benefit of PRC_v2.1.

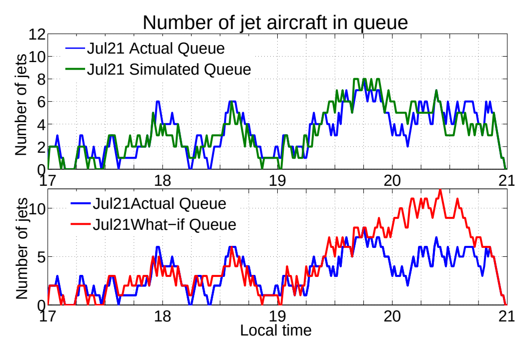


Fig. 9: Observed and predicted queue lengths on Jul 21, 2011.

The overall estimation of fuel burn savings is conducted by using the simulated taxi-out time savings for all test periods, and models of taxi fuel burn [23, 39]. The total fuel burn reduction estimated to be 2,650 US gallons, which translates to an average savings of about 57 kg per gate-held flight. Table II summarizes the results of the eight demo periods with significant gate-holds.

2) Distribution of benefits: Equity is an important criterion in evaluating potential congestion management strategies. The PRC approach, as implemented here, invokes a First-Come-First-Serve (FCFS) policy in clearing flights for pushback,

TABLE II: Summary of the eight demo periods with significant gate-holds during the PRC_v2.1 field-tests in 2011.

Date	Period	Configuration	No. of gate-holds	Total gate-hold time (min)
7/18	4.45-8PM	22L, 27 | 22L, 22R	14	28
7/21	5.15-9PM	22L, 27 | 22L, 22R	42	384
7/22	5.15-8.30PM	22L, 27 | 22L, 22R	50	290
7/24	5.15-8PM	4L ,4R | 4L, 4R, 9	12	13
7/28	5.30-8PM	4L ,4R | 4L, 4R, 9	7	13
8/11	5.30-8.15PM	22L, 27 | 22L, 22R	6	9
8/14	5.00-6.30PM	22L, 27 | 22L, 22R	1	1
	6.30-7.30PM	4L ,4R | 4L, 4R, 9	0	0
9/11	5.30-6.30PM	4L ,4R | 4L, 4R, 9	0	0
	6.30-8.15PM	22L, 27 | 22L, 22R	12	23
Total			144	761

since this is the widely accepted measure of fairness in air traffic control [40]. One would therefore expect no bias toward any airline with regard to gate-holds incurred, and that the number of gate-holds for an airline would be commensurate with its contribution to departure traffic during congested periods. However, in practice, inadvertent resequencing due to pushback procedures, taxi speeds or sudden changes in demand can potentially lead to differences between the gate-hold time and taxi-out savings of individual flights. Similarly, the benefit of a gate-hold can also extend to other flights [10]. Figure 10 shows that during the PRC_v2.1 field-tests, the FCFS sequence was mostly maintained, and that the gate-hold times were approximately equal to the taxi-time reduction experienced by each airline. It must be noted that the fuel burn

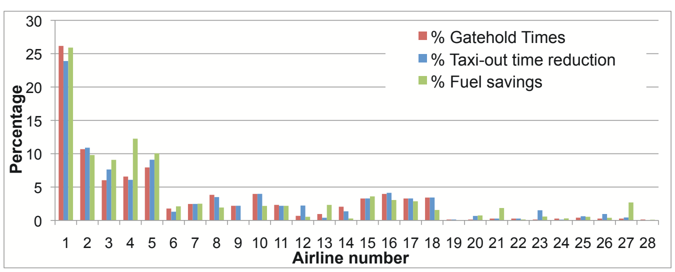


Fig. 10: Percentage of gate-hold times, taxi-out time reduction and fuel burn savings corresponding to each airline.

benefit to an airline depends on its fleet mix. Figure 10 shows that while the taxi-out time reductions were similar to the gate-hold times, some airlines (for example, Airlines 4, 13, 21 and 27) enjoyed a greater proportion of fuel savings. These airlines were typically those which operated several Heavy aircraft during the evening periods.

C. Takeoff rate prediction

As explained in Section V, the PRC v_2.1 algorithm was used to predict the jet takeoff rate. The predictions were validated during shadow testing by means of visual observations, and were subsequently used during the 19 days of the trials. Table III reports the mean error (ME), mean absolute error (MAE) and root mean square error (RMSE) of the predicted jet takeoff rate (relative to the observed value) over the 182 15-min periods of field-tests. The errors for the 93 periods with at least 10 taxiing jet aircraft are also shown, because gate-holds were most likely in these times.

TABLE III: Error statistics for PRC v_2.1 predictions of the jet aircraft takeoff rate.

	All traffic conditions			$\geq$ 10 jets taxiing-out		
	ME	MAE	RMSE	ME	MAE	RMSE
PRC v_2.1	-0.20	1.25	1.64	-0.03	1.14	1.58

Table III shows that the regression tree-based prediction algorithm used in PRC_v2.1 predicts the takeoff-rate quite well. The mean absolute error is only 1.14 during moderate and high traffic conditions (10 or more jets taxiing-out). Most importantly, the prediction errors are within the level of uncertainty considered in the design of the PRC_v2.1 strategy. For the 22L, 27 | 22L, 22R configuration with at least 10 jet departures taxiing, the takeoff rate was underestimated by at most 2.7, while the PRC_v2.1 algorithm tries to maintain a queue of at least 4 aircraft for these conditions. Similarly, for the 4L, 4R | 4L, 4R, 9 configuration with at least 10 jet departures taxiing, the takeoff rate was underestimated by at most 3.7, while the PRC_v2.1 algorithm tries to maintain a queue of at least 5 aircraft. These observations suggest that the inventory targeted by the algorithm at the queue was sufficient to avoid runway underutilization; a more aggressive congestion control policy may have resulted in an empty runway queue. The importance of maintaining a sufficiently large runway queue has been recognized by other researchers as well [22].

D. Qualitative observations

1) Compatibility with traffic flow management initiatives: An important goal of this effort was to investigate the compatibility of Pushback Rate Control with other traffic flow management initiatives. On two field-test periods, controllers demonstrated that they could handle airspace restrictions such as Minutes-In-Trail (MINIT) programs and target departure times (e.g., EDCTs) while executing the PRC_v2.1 strategy. It was shown that if known in advance, delays due to controlled departure times could be efficiently absorbed as gate-holds.

2) Increased predictability: During the field-tests, once the suggested pushback rate was given to the controller at the start of each time period, the controller communicated the expected release times to all aircraft on hold. The advance notice of expected pushback clearance times improves predictability, and can be useful in planning ground resources.

VIII. Evaluation of the DST

After the field-tests at BOS had been completed, air traffic controllers in the ATCT were surveyed regarding their opinions on the study as a whole, and specifically on the implementation and use of the DST. The survey responses were positive and the controllers liked the DST [41]. Several responses also supported combining BG and another position, removing the need for a dedicated controller during gate-holds. Comments on the best features of the DST included “the ability to touch planes”, “reserve spots”, “[the ability to] count the planes and account for aircraft with long delays”, "allows me to push and tells me to hold", and "easy to use and understand". Suggestions for improvement included increasing the icon sizes and maintaining more pressure on the runway.

IX. Conclusions

This paper presented the design and field-testing of a Pushback Rate Control strategy at Boston Logan International Airport (BOS). The proposed approach used historical data to predict the performance of the airport under a set of operating conditions, and used dynamic programming to balance the objectives of maintaining runway utilization and limiting surface congestion. The optimal policy was a recommended rate at which aircraft were cleared for pushback. A decision support interface was designed to display the suggested pushback rate, and to help air traffic controllers keep track of requests for pushback, gate-holds and other metering constraints.

During 8 four-hour tests conducted at BOS during the summer of 2011, fuel use was reduced by an estimated 9 US tons (2,650 US gallons), while carbon dioxide emissions were reduced by an estimated 29 US tons. Aircraft gate pushback times increased by an average of 5.3 minutes for the 144 flights that were held at the gate, but with a corresponding decrease in taxi-out times. Finally, a survey of the air traffic controllers involved in the 2011 demo indicated strong support for the Pushback Rate Control approach, the manner of implementation, and the decision support tools developed for the deployment of such strategies.

Acknowledgments

This work was supported by the Federal Aviation Administration's Office of Environment and Energy through MIT Lincoln Laboratory and the Partnership for AiR Transportation Noise and Emissions Reduction (PARTNER). The demo was helped make possible by Brendan Reilly, Deborah James, Pat Hennessy, John Ingaharro, John Melecio, Michael Nelson and Chris Quigley at the BOS Facility; Vincent Cardillo, Flavio Leo and Robert Lynch at Massport; and George Ingram and other airline representatives at the ATA. Vivek Panyam (University of Pennsylvania) volunteered his effort in the development of the Android tablet application. John Hansman, Harshad Khadikar and Tom Reynolds participated in tower observations before and during the trials; Harshad Khadilkar also analyzed the ASDE-X data. Alex Nakahara helped compute the preliminary fuel burn savings from the gate-hold data. We thank Steve Urlass and Lourdes Maurice at the FAA for their support, and James Kuchar and Jim Eggert of MIT Lincoln Laboratory for their support and help with the ASDE-X data.

References

[1] Federal Aviation Administration, "Aviation System Performance Metrics database," `http://aspm.faa.gov/aspm/ASPMframe.asp`, accessed February 2012.

[2] K. N. Yu, Y. P. Cheung, T. Cheung, and R. C. Henry, "Identifying the impact of large urban airports on local air quality by nonparametric regression," *Atmospheric Environment*, vol. 38, no. 27, pp. 4501–4507, 2004.

[3] D. C. Carslaw, S. D. Beevers, K. Ropkins, and M. C. Bell, "Detecting and quantifying aircraft and other on-airport contributions to ambient nitrogen oxides in the vicinity of a large international airport," *Atmospheric Environment*, vol. 40, no. 28, pp. 5424–5434, 2006.

[4] M. A. Miracolo, C. J. Hennigan, M. Ranjan, N. T. Nguyen, T. D. Gordon, E. M. Lipsky, A. A. Presto, N. M. Donahue, and A. L. Robinson, "Secondary aerosol formation from photochemical aging of aircraft exhaust in a smog chamber," *Atmospheric Chemistry and Physics*, vol. 11, no. 9, pp. 4025–4610, 2011.

[5] 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.

[6] P. Burgain, E. Feron, J. Clarke, and A. Darrasse, "Collaborative Virtual Queue: Fair Management of Congested Departure Operations and Benefit Analysis," *Arxiv preprint arXiv:0807.0661*, 2008.

[7] F. Carr, A. Evans, E. Feron, and J. Clarke, "Software tools to support research on airport departure planning," in *Digital Avionics Systems Conference*. Irvine CA: IEEE, 2002.

[8] F. Carr, "Stochastic modeling and control of airport surface traffic," Master's thesis, Massachusetts Institute of Technology, 2001.

[9] 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*, pp. 1835–1852, 1999.

[10] I. Simaiakis and H. Balakrishnan, "Queuing Models of Airport Departure Processes for Emissions Reduction," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2009.

[11] Federal Aviation Administration, "Fact Sheet of Airport Surface Detection Equipment, Model X (ASDE-X)," October 2010.

[12] R. Clewlow and D. Michalek, "Logan Control Tower: Controller Positions, Processes, and Decision Support Systems," Massachusetts Institute of Technology, Tech. Rep., 2010.

[13] M. Spearman and M. Zazanis, "Push and pull production systems: Issues and comparisons," *Operations research*, pp. 521–532, 1992.

[14] 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.

[15] 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.

[16] 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.

[17] S. Stidham Jr, "Analysis, design, and control of queueing systems," *Operations Research*, vol. 50, no. 1, pp. 197–216, 2002.

[18] A. Nakahara, T. Reynolds, T. White, and R. Dunsky, "Analysis of a Surface Congestion Management Technique at New York JFK Airport," in *AIAA Aviation Technology, Integration and Operations Conference*, September 2011.

[19] C. Brinton, C. Provan, S. Lent, T. Prevost, and S. Passmore, "Collaborative Departure Queue Management: An Example of Collaborative Decision Making in the United States," in *9th USA/Europe Air Traffic Management Research and Development Seminar (ATM2011)*, Berlin, Germany, June 2011.

[20] Y. Jung, T. Hoang, J. Montoya, G. Gupta, W. Malik, and L. Tobias, "Performance Evaluation of a Surface Traffic Management Tool for Dallas/Fort Worth International Airport," in *9th USA/Europe Air Traffic Management Research and Development Seminar (ATM2011)*, Berlin, Germany, June 2011.

[21] D. Böhme, "Tactical departure management with the Eurocontrol/DLR DMAN," in *6th USA/Europe Air Traffic Management Research and Development Seminar, Baltimore, MD*, 2005.

[22] M. Schaper, G. Tsoukala, R. Stavrati, and N. Papadopoulos, "Departure flow control through takeoff sequence optimisation: Setup and results of trials at Athens airport," in *2011 IEEE/AIAA 30th Digital Avionics Systems Conference (DASC)*. IEEE, 2011, pp. 2B2–1.

[23] 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 Eurocontrol/FAA ATM R&D Seminar*, 2011.

[24] J. W. Smeltink, M. J. Soomer, P. R. de Waal, and R. D. van der Mei, "An Optimisation Model for Airport Taxi Scheduling," in *Thirtieth Conference on the Mathematics of Operations Research*, Lunteren, The Netherlands, January 2005.

[25] S. Rathinam, J. Montoya, and Y. Jung, "An Optimization Model for Reducing Aircraft Taxi Times at the Dallas Fort Worth International Airport," in *26th International Congress of the Aeronautical Sciences*, Anchorage, AK, September 16 2008.

[26] H. Balakrishnan and Y. Jung, "A Framework for Coordinated Surface Operations Planning at Dallas Fort Worth International Airport," in *AIAA Guidance, Navigation, and Control Conference*, Hilton Head, SC, August 20-23 2007.

[27] H. Lee and H. Balakrishnan, "Optimization of Airport Taxiway Operations at Detroit Metropolitan Airport (DTW)," in *AIAA Aviation Technology, Integration and Operations Conference*, September 2010.

[28] D. Bertsimas and S. S. Patterson, "The air traffic flow management problem with enroute capacities," *Operations Research*, vol. 46, no. 3, pp. 406–422, 1998.

[29] X. Hu, W. Chen, and E. D. Paolo, "Multi-airport capacity management:

Genetic algorithm with receding horizon," *IEEE Transaction on Intelligent Transportation Systems*, vol. 8, No. 2, pp. 254–263, 2007.
[30] I. Simaiakis and H. Balakrishnan, "Departure throughput study for Boston Logan International Airport," Massachusetts Institute of Technology, Tech. Rep., 2011, No. ICAT-2011-1.
[31] H. Lee, I. Simaiakis, and H. Balakrishnan, "A Comparison of Aircraft Trajectory-Based and Aggregate Queue-Based Control of Airport Taxi Processes," in *Digital Avionics Systems Conference*. IEEE, 2010.
[32] I. Simaiakis, "Analysis, Modeling and Control of the Airport Departure Process," Ph.D. dissertation, Massachusetts Institute of Technology, 2013.
[33] S. M. Ross, *Stochastic Processes*. Wiley Series in Probability and Statistics, 1995.
[34] S. Lippman, "Applying a new device in the optimization of exponential queuing systems," *Operations Research*, vol. 23, no. 4, pp. 687–710, 1975.
[35] D. Low, "Optimal dynamic pricing policies for an M/M/s queue," *Operations Research*, vol. 22, no. 3, pp. 545–561, 1974.
[36] R. Bellman, *Dynamic Programming*. Princeton University Press, 1957.
[37] E. Gilbo, "Airport capacity: Representation, estimation, optimization," *IEEE Transactions on Control Systems Technology*, vol. 1, no. 3, pp. 144–154, 1993.
[38] D. Bertsekas, *Dynamic programming and optimal control*. Athena Scientific, 2005.
[39] H. Khadilkar and H. Balakrishnan, "Metrics to evaluate airport operational performance using surface surveillance data," *Air Traffic Control Quarterly*, 2013, to appear.
[40] R. G. Dear, "The dynamic scheduling of aircraft in the near terminal area," MIT Flight Transportation Laboratory, Tech. Rep., 1976.
[41] I. Simaiakis, M. Sandberg, H. Balakrishnan, and R. Hansman, "Design, testing and evaluation of a pushback rate control strategy," in *International Conference on Research in Air Transportation*, 2012.

Ioannis Simaiakis is a Management Consultant with McKinseys Operations Practice in Washington, DC. He received his Ph.D. in Aeronautics and Astronautics from the Massachusetts Institute of Technology. His research focuses on modeling and predicting taxi-out times and airport operations planning under uncertainty.

Melanie Sandberg is associate staff in the Air Traffic Control Division of MIT Lincoln Laboratory. She obtained her Master of Science in Transportation from the Massachusetts Institute of Technology. Her research interests include operational strategies for improvements in air traffic management.

Hamsa Balakrishnan is an Associate Professor of Aeronautics and Astronautics at the Massachusetts Institute of Technology. She received her PhD in Aeronautics and Astronautics from Stanford University. Her research interests include optimization and control algorithms for air transportation systems.