Design and Simulation of Airport Congestion Control Algorithms

Ioannis Simaiakis and Hamsa Balakrishnan
Massachusetts Institute of Technology, Cambridge, MA, USA
Email: {ioa_sim, hamsa}@mit.edu

***Abstract*—This paper proposes a new airport model and a dynamic programming algorithm for controlling the departure process at congested airports. Using a multi-variable state description that includes the capacity forecast, the runway system is modeled as a semi-Markov process. The paper then proposes a queuing model for modeling the controlled departure process that enables the efficient calculation of optimal pushback policies using decomposition techniques. The developed algorithm is simulated at Philadelphia International Airport, and compared to other potential control strategies including a threshold-policy. The algorithm is also shown to effectively adapt to changes in airport departure capacity, maintain runway utilization and efficiently manage congestion.**

I. Introduction

A. Motivation

Major airports worldwide frequently suffer from surface congestion and its undesirable impacts, such as excessive taxi-out times, fuel consumption and emissions. In the United States, Philadelphia (PHL) airport has been seen to exhibit frequent periods of excessive congestion, with significantly more active aircraft than needed to maintain throughput. During such congested periods, the average taxi-out time at PHL was 38 min even in good weather, far more than the unimpeded taxi-out time of 12 min [1].

Congestion at an airport such as PHL can be analyzed using data from the Federal Aviation Administration's (FAA) Aviation System Performance Metrics (ASPM) database [2]. Figure 1 (top) shows the counts of aircraft pushbacks and takeoffs during each 15-minute time window, averaged over all days in 2011 during which aircraft landed on Runway 27R and took off from Runway 27 L. The average departure capacity of this runway configuration at PHL, estimated to be 13 aircraft/15 min [3], is also shown. This plot illustrates that while the departure capacity is limited, the demand (pushback rate) can be much higher. The detrimental effect of this imbalance on taxi-out times is seen in Figure 1 (bottom).

B. Background and related work

There are several possible options in designing a congestion control strategy. The simplest approach is an open-loop control policy that would restrict the demand to a value approximately equal to the departure capacity. This form of demand management, known as slot control, is employed at most major European airports and at a handful of US ones [4].

A simple closed-loop control strategy would be a state-dependent pushback policy aimed at reducing surface congestion, such as *N-Control* [5], [6], [7], [8]. The N-Control policy is a threshold heuristic that stops pushbacks when the number of departing aircraft on the ground exceeds a certain value, and restarts them when the number of departures on the ground drops below it. This approach is similar to *constant work-in-process* (CONWIP) policies in manufacturing systems [9].

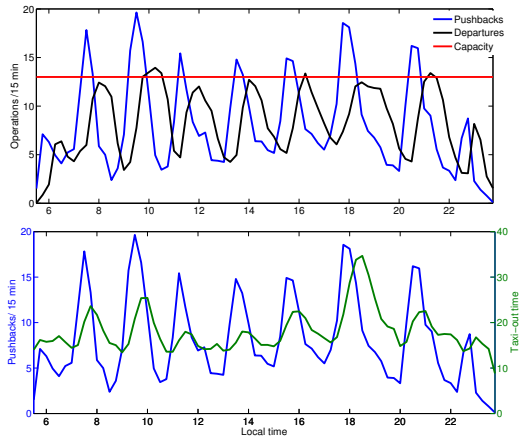

Fig. 1: (Top) Average number of pushbacks, average number of takeoffs and departure capacity per 15 minutes at PHL in 2011; (Bottom) Average number of pushbacks per 15 minutes and average taxi-out times.

In prior work, we showed that on-off or event-driven policies for controlling the pushback process are difficult to implement in practice [10]. Air traffic controllers prefer being given a dispatch rate that is valid for a predefined time period, after which it can be updated. We refer to this class of policies as *Pushback Rate Control (PRC)*. In addition, the actuation occurs at the gates during pushback, while the chief constraint is the runway. The control strategy therefore has to accommodate stochastic taxi-out times between the gate and the runway. In order to address these issues, we developed and tested two variants of PRC at Boston airport. The first, PRC_v1.0, was a rate-based approximation of the N-Control policy [11], while the second, PRC_v2.0, used dynamic programming to suggest a rate at which aircraft would push back from their gates, so as to keep the airport from becoming highly congested [10]. The suggested rate was was periodically updated depending on operating conditions (weather, configuration, fleet mix and arrival demand), the number of aircraft taxiing out, and the load of the departure queue.

In order to achieve widespread implementation, PRC protocols need to be adapted to different operational environments. PRC_v2.0 presents several challenges to such adaptation, including the dimension of the state-space, sensitivity of the solution to uncertain parameters, and the definition of

the objective function [3]. This paper presents an alternate approach (henceforth referred to as PRC_v3.0) that resolves these problems using a new runway service process model. Using simulations of operations at PHL, N-control and the PRC variants are compared, and the tradeoffs between different airport congestion control strategies are assessed.

II. Design requirements

The objective of the control strategy is to minimize the amount of taxiing-out traffic, and thus taxi-out times, while maintaining runway utilization. In addition, the desired form of a congestion control strategy is one that periodically recommends a pushback (release) rate to air traffic controllers [11]. The suggested pushback rate is updated at the beginning of each time-window, and is valid for the duration of it. If the length of the time-window, Δ, is approximately equal to the delay between actuation and control (that is, the expected travel time from the gates to the departure queue), the flights released from the gate during a given time period would be expected to reach the departure queue in the next time period.

III. Control strategy

A. System dynamics

The state N_t of the departure process at time t consists of the number of aircraft traveling from the gates to the departure queue (R_t), and the number of aircraft in the departure queue (Q_t), that is, $N_t = (R_t, Q_t)$. Both elements of the state, R_t and Q_t, can be observed using surface surveillance data.

The start of each time-window is called an *epoch*. Suppose the state at epoch τ is (R_τ, Q_τ), and the decision maker selects a pushback rate λ_τ for that time period. Let us assume that i out of the λ_τ aircraft reach the runway during the time interval $(\tau, \tau+\Delta]$ with probability β_i. Similarly, i out of 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 $1-\sum\beta_i-\sum\gamma_i$. In all cases, the queue at time $\tau+\Delta$ will be a function f of the aircraft that were in the queue at τ and the number of aircraft that reach the queue during the time interval Δ. The queuing system therefore evolves as follows:

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

B. Control algorithm

At the beginning of each time-window, the algorithm recommends a pushback rate $\lambda \in \Lambda = [0, \hat{\lambda}]$, expressed as the number of pushbacks per Δ minutes. 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 operational procedures and airport layout.

The control policy tries to balance two objectives, namely, to minimize the expected departure queue length and to maximize the runway utilization. These objectives are reflected in the cost function $\bar{c}(r,q)$ (Section V). The optimal average cost per stage, c^* is given by Bellman's equation:

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

IV. Parametric throughput forecasts

A wide range of factors such as fleet mix and the expected number of landings in the next time period can provide a conditional forecast for the runway service time distribution [12]. These parameters help explain some of the variance in the departure throughput, and provide a more accurate estimate of the expected departure capacity. We extend the state space to include the throughput forecast (F) as a state of the system:

$$N_t = (R_t, Q_t, F_t) \quad (2)$$

The optimal average cost per stage, c^* is then given by:

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

where p_f is the probability of each throughput forecast and $\mathbf{p}_q(r,q,\mathrm{f})$ is the probability vector of the state of the queue at the end of the time-window given that the state at the beginning of the time window is (r,q,f). The resulting policy is denoted PRC_v3.0.

V. Runway service process

In prior work, the runway service process was modeled as an Erlang distribution whose parameters were estimated from empirical data [10]. While the Erlang distribution is convenient for modeling the evolution of a runway queuing system, the numerical solution of the Chapman-Kolmogorov equations can be computationally expensive [13], especially in case of multiple possible throughput distributions with different shape parameters.

A. $(M(t)|R_0)/D_s/1$ model

We assume that during each time window the service rate is deterministic, but unknown. It is one of a finite set $\mu_1, \mu_2, \ldots, \mu_s$, with probabilities derived from the empirical distribution (for example, Figure 2). The set $\mu_1, \mu_2, \ldots, \mu_s$, of cardinality s is the support of the empirical distribution function.

For the example in Figure 2, the throughput process within each 15-minute window is assumed to be deterministic with rate 5, 6, ..., or 13. The probability of each rate equals the corresponding probability mass in the empirical throughput distribution. For example, the probability that the service rate is 10/15 min is 0.28. While this model reflects the empirical

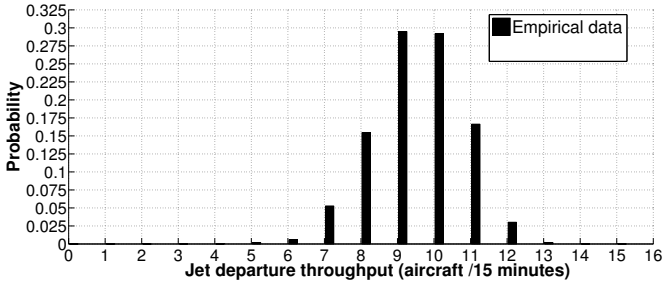

Fig. 2: Example of an empirical probability distribution of departure capacity.

probability distribution of the departure throughput, it does not reflect the fact that a service rate of 10 aircraft/15 min does not imply uniformly spaced service times of 1.5 min. We use the following notation:

- μ_i: Possible service rate (aircraft/15 min).
- Σ: Set, of cardinality s, of all possible service rates $\mu_1, \mu_2, \ldots, \mu_s$.
- $(R, Q; \mu_i)$: State of the queuing system, given the deterministic service rate μ_i.
- F: Set, of cardinality z, of all throughput forecasts $f_1, f_2, \ldots, f_z$.
- $w(i; f_j)$: Probability that the service rate equals μ_i, given throughput forecast f_j.

Consider the case of a single possible service rate, μ. In a given time-window, the system resembles a transient $M(t)/D/1$ queuing system, except that the number of arrivals to the queue during the time-window is known. We denote such a system as $(M(t)|R_0)/D/1$, and analyze it by extending the framework proposed by Koopman [14]. In this framework, the service epochs are a priori marked on the time axis. For example, when the service rate is 10/15, the (potential) service time epochs are marked at times $1.5, 3, \ldots, 15$ min from the beginning of the time window. If an aircraft arrives at an empty system in minute 1, it must wait until 1.5 min before its service starts. Delays at lower states may therefore be overestimated. This assumption however makes the analysis tractable: If at epoch 0, R_0 aircraft are taxiing out, the probability mass function $\hat{g}$ of k arrivals between the departure runway service times i and $i+1$ assuming that $j-k$ aircraft have already arrived, is:

$$\begin{aligned}\hat{g}(i,j,k) &= \Pr\{k \text{ arrivals in}(t_i, t_{i+1}] | (R_0-(j-k)) \text{arrivals in}(t_i, \Delta]\} \\ &= \begin{cases} \binom{R_0-(j-k)}{k} \left(\frac{\tau_{i+1}-\tau_i}{\Delta-\tau_i}\right)^k \left(\frac{\Delta-\tau_{i+1}}{\Delta-\tau_i}\right)^{(R_0-j)}, \\ \qquad\qquad \text{if } 0 \le k \le j, j \le R_0, \tau_{i+1} \le \Delta \\ 0, \quad \text{otherwise} \end{cases}\end{aligned} \tag{3}$$

In this case, the state of the runway system is denoted as $(R, Q; \mu)$, where R is the number of aircraft traveling to the runway, Q is the number of aircraft in the queuing system (in service or in queue) and μ is the deterministic service rate. Now, we observe that $\hat{g}(i,j,k)$ is the probability of transitioning from state $(R_0-(j-k), j-k+\mathbf{1}_{\{j-k\ge 1\}}; \mu)_{\tau_i}$ to state $(R_0-j, j; \mu)_{\tau_{i+1}}$. The condition $j-k \ge 1$ implies that there were one or more aircraft in the system before the arrival of the k aircraft between service times i and $i+1$, and one of them was served. At epoch 0 the system is in state (R_0, Q_0, μ). The state of the queuing system at time Δ, $\hat{Q}_\Delta(\mu)$, is a probabilistic function of the initial value (R_0, Q_0, μ), the functions $\hat{g}(i,j,k)$ describing the probability of each allowable transition, and the assumed service rate μ.

In the full system, for each throughput forecast f, we have a finite set $\mu_1, \mu_2, \ldots, \mu_s$ of possible service rates, each with probability $w(1; f), w(2; f), \ldots, w(s; f)$. The state of the queuing system at Δ, Q_Δ is therefore given by:

$$Q_\Delta(f) = \sum_{i=1}^{s} w(i; f) \cdot \hat{Q}_\Delta(\mu_i). \tag{4}$$

Equation (4) shows the benefit of this formulation: The probability vector of the state of the system $Q_\Delta(f)$ given a throughput forecast f is decomposed in a weighted sum of $\hat{Q}_\Delta(\mu_i)$'s, which are independent of the weights of the summation $w(1; f), w(2; f), \ldots, w(s; f)$. A different throughput forecast f can be modeled by merely changing the weights $w(i; f)$ in Equation (4).

We denote this queuing model of deterministic service times sampled from a finite set and a known number of arrivals at random times as $(M(t)|R_0)/D_s/1$. Moreover, Equation (4) offers the ability to track each individual arrival at the queue. Each possible transition is assigned a probability $(\hat{g}(i,j,k))$ and a cost. The cost has two components, that of queuing and non-utilization of the runway. For the queuing cost, the first of k arrivals between service times i and $i+1$ will encounter a system with $j-k$ aircraft, the second $j-k+1$ aircraft, etc. Each transition is penalized in terms of its expected queuing delay (in minutes). Similarly, each transition from an empty system ($j=k$), is penalized in terms of a loss of runway utilization. The runway non-utilization cost is the minutes of additional delay for later flights due to the capacity loss. The optimal pushback policies are obtained from the optimal cost-per-stage solutions for the $(M(t)|R_0)/D_s/1$ model.

B. Comparison of $(M(t)|R_0)/D_s/1$ and $(M(t)|R_0)/E(k)/1$ models

In Figure 3, the $(M(t)|R_0)/D_s/1$ model is compared to the $(M(t)|R_0)/E(k)/1$ model used by PRC_v2.0 [15] in terms of predicting the state of the queue after a 15-minute period given the range of possible initial conditions Q and R. In the $(M(t)|R_0)/E(k)/1$ model, the runway service time are modeled as Erlang-distributed, and the state probabilities are calculated by deriving the first-order differential equations (*Chapman-Kolmogorov equations*) [15]. We assume a single throughput forecast (Figure 2), and that parameters β_i and γ_i in Equation (1) equal zero.

The two models predict the same value of expected queue for most initial conditions, except for states in which the initial queue length is very low (0-3 aircraft). For example, given 0 aircraft queuing and 12 aircraft traveling to the runway, $(M(t)|R_0)/D_s/1$ predicts an expected queue of 4 after 15 min,

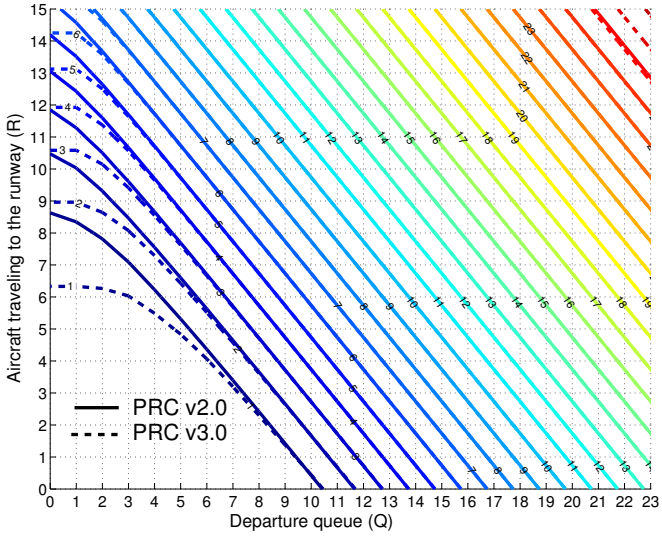

Fig. 3: Expected queue length after 15 min as a function of the number of aircraft in the departure queue (R) and the number of aircraft traveling to the runway (Q) for the $(M(t)|R_0)/E(k)/1$ model (solid line), and the $(M(t)|R_0)/D_s/1$ model (dashed line).

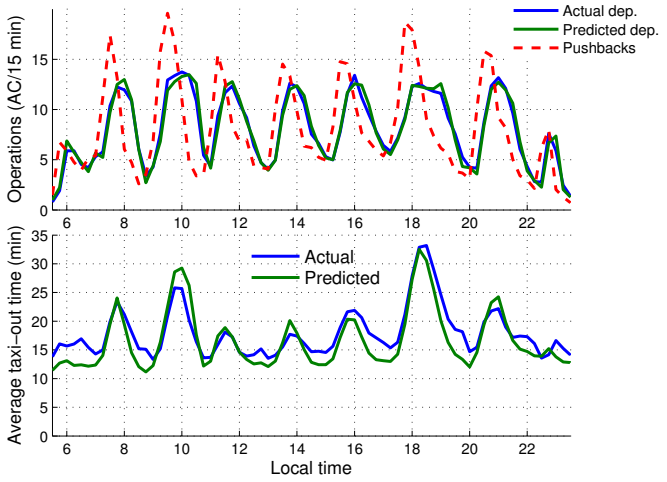

Fig. 4: (Top) Average number of pushbacks, and average numbers of actual and predicted takeoffs by time of day at PHL in 2011; (Bottom) Average actual and predicted taxi-out times by time of day.

whereas $(M(t)|R_0)/E(k)/1$ predicts an expected queue of 3. The $(M(t)|R_0)/D_s/1$ model is conjectured to underestimate the throughput (and overestimate the queue) in these cases, because of the assumption that service times are a priori equally-spaced in the time-window.

VI. Simulation of Congestion Control Strategies

A. Departure process model

We focus on the departure process for the runway configuration 26, 27R, 35 | 27L, 35, which was in use 74% of the time under VMC in 2011. In this runway configuration, the main departure runway is 27L, and the main arrival runway is 27R. ASDE-X analysis shows that there is one departure on runway 35 for every 11.5 departures on 27L. Additionally, the thresholds of the two runways are very close to each other, and the aircraft heading to both of them are part of the same flow [3].

The model can be used to predict departure throughput and taxi-out times through a day at PHL. Figure 4 (top) shows the average number of pushbacks and the average number of takeoffs (or departures) that were recorded during each 15-minute time window for all days in which this runway configuration was in use in 2011. It also shows the average number of departures predicted by the model. Figure 4 (bottom) shows the actual and predicted average taxi-out times for the flights that pushed back in each 15-minute time window. We observe that the model is representative of an average day at PHL.

B. Calibration of PRC_v3.0

The μ and w parameters of Equation (3) that are necessary for estimating the queuing transition probabilities are estimated from empirical throughput distributions. In particular, the model makes use of seven departure throughput distributions, with average throughput ranging between 10 and 14 aircraft/15 min depending on arrival demand, route blockage and fleet mix factors [3]. As explained in Section V, it is not necessary to recalculate the transition probabilities, nor the congestion costs. The maximum pushback rate is set to 24 aircraft/15 min, which is the maximum pushback rate observed at PHL after eliminating outliers. This rate is assumed to be the maximum admissible rate of arrivals into the departure process (pushbacks).

The next step involves the decision of the optimal time window, Δ, for this runway configuration. For this, we do a simple flow analysis. On average, in a controlled scenario, aircraft enter the system at the same rate as they exit, namely, the average departure throughput (13 aircraft/15 min). The corresponding average travel time of each aircraft is 14.6 min [3]. Δ is therefore chosen to be 15 min.

The last step involves calculating the probabilities $\beta_i(R,\lambda)$ and $\gamma_i(R,\lambda)$, which are necessary for deriving the system dynamics (Equation (1)). We use Monte Carlo simulations to estimate the empirical distribution of the number of aircraft traveling to the runway at the next epoch, $R_{\tau+\Delta}$ given the current number of aircraft traveling, R_τ, and the current pushback rate, λ_τ. The system evolves from a randomized initial condition, and we select random pushback rates every 15 minutes. These rates are allocated to airlines according to their relative presence at the airport. Every 15 minutes, we record the transition $R_{\tau+\Delta}$, given the current R_τ and pushback rate λ_τ. Finally, we derive β_i and γ_i from the simulated empirical distributions $R_{\tau+\Delta} = g(R_\tau, \lambda_\tau)$.

C. Simulation setup

The simulations are used to evaluate the models used in PRC_v3.0, and also to compare it to other congestion control mechanisms such as *N-Control* and *Slot-Control*.

The airport is seen to saturate when 20 aircraft are taxiing-out, that is, $N^* = 20$ [3]. A value of $N_{ctrl} = N^* = 20$, is used for the N-Control policy. Simulations show that for $N_{ctrl} = 20$, aircraft do not incur additional delay resulting from gate-holding. Given that $N_{ctrl} \geq N^*$, the resulting taxi-out time reduction is the highest that can be achieved with N-Control.

The capacity envelope of PHL is used for simulating Slot-Control. The average departure capacity of this runway configuration at PHL is 13 aircraft/15 min, and is not found to change significantly with arrival throughput [3]. Slot-Control can therefore be simulated by limiting pushbacks to the average departure capacity. This policy is open-loop and easy to simulate.

For all control policies, we impose the additional constraint that the pushback rate does not exceed 4 aircraft/min, which was the maximum number of pushbacks/minute achieved at PHL in 2011. This additional constraint reflects the fact that pushback coordination and communication require a certain minimum time.

Finally, the earliest possible pushback time of each flight is its recorded actual pushback time. This means that in all scenarios, pushbacks can only be delayed, and not advanced. In addition, for the cases of PRC and Slot-Control, if the pushback requests in a 15-minute time-window are fewer than the optimal pushback rate and the pushback cap respectively, the remaining slots are unutilized. This loss of runway utilization is because of a lack of sufficient demand at this time period, and not because of the control scheme. In the case of N-Control, there are similar instances without sufficient demand to bring the number of aircraft on the surface to the N_{ctrl} value.

D. Simulation results

For simulating the three strategies, we run 100 Monte Carlo simulations sampling the unimpeded taxi-out time of each flight from the corresponding distribution, and sampling the runway service time. We also simulate a *do-nothing* scenario as a baseline. The results are summarized in Table I, for the 136,430 flights that pushed back and departed in this configuration at PHL in 2011. The column "mean delay" lists the additional takeoff delay that flights incur as a result of the control scheme, and is the difference between the take-off time in the baseline scenario and the takeoff time in the controlled scenario.

Table I shows that PRC_v3.0 reduces average taxi-out times by 1.66 min, while increasing the average delay by only 0.03 minutes, compared to the do-nothing (baseline) case. It also reduces the variability of taxi-out times. The N-Control strategy yields slightly smaller taxi-out time savings, but with no added delays. It is worth noting that the PRC_v3.0 strategy achieves similar results to the N-Control strategy despite only being applied every 15 minutes. We conjecture that this is because of the predictive nature of PRC. Instead of aiming to keep the taxiing-out traffic below 21 aircraft, it uses information on the current state of the airport to predict the departure capacity and the queue in the next 15 minutes. The pushback rate is then set so as to optimize the load of the queue.

Table I also suggests that the taxi-out time savings from the Slot-Control policy are less than those of either N-Control or PRC_v3.0, and are achieved with an increased average delay of 0.27 min or a total added delay of 614 hours per year. This weaker performance is due to the level of stochasticity in the PHL departure process, which makes it unsuitable for an open-loop policy. Despite the fact that the number of pushbacks is capped at the average departure capacity of the system, loss of runway utilization occurs often enough that this capacity loss propagates to delay later aircraft. We also remark that Slot-Control would result in more variable taxi-out times than N-Control and PRC_v3.0. Taxi-out times grow much higher under Slot-Control because of the absence of a feedback mechanism at times of significant congestion.

Figure 5 shows the average traffic by time of day resulting from a single run of the N-Control simulation. Similarly, Figure 6 shows the average traffic by time of day resulting from a single run of PRC simulation. From the lower plots of the figures, we notice that both strategies are very effective in reducing long taxi-out times, and in particular removing the taxi-out time peaks at 1000 and 1900 hours. The upper plots show that the controlled pushback rates exhibit a similar trend under both strategies. It is high in the beginning of each departure push, but it is subsequently rapidly reduced. Both strategies initially try to load the runway queue, and subsequently regulate the flow of aircraft on the surface.

Pushback rates at the beginning of each departure push are slightly higher under the PRC_v3.0 strategy (for example at 1745 hours). At low traffic levels, the optimal pushback rate under PRC_v3.0 is high, since it aims to build up the queue at the runway. Subsequently, given the current state of the surface, (R,Q), and the predicted capacity, the pushback rate is regulated so as to maintain a desired inventory of aircraft at the queue, in this case 5-6 aircraft. In steady state, there will be 5-6 aircraft in queue and 13 aircraft taxiing to the runway. Although the initial level of traffic is higher for PRC_v3.0 than for N-Control, it subsequently stabilizes at lower values (18-19 aircraft) on average.

Figure 7 shows the average traffic by time of day resulting from a single run of the Slot-Control simulation. The trend of pushbacks under Slot-Control is very different from that under the two other strategies. The pushback rate is always capped at 13 aircraft/15 min. For example, the evening departure push is evenly distributed in the 1-hour time window 1745 - 1845 hours. The lower plot of Figure 7 shows that the smoothing of the pushbacks results in significant taxi-out time reduction. Aircraft pushback at the rate that they takeoff, and delays build up very slowly.

Figure 8 compares the simulated taxi-out times from the three control strategies to the do-nothing approach, during the evening times. In the primary evening departure push between 1730 and 2000 hours, all control strategies achieve significant

TABLE I: Taxi-out time predictions from simulating different control strategies.

Control algorithm	Taxi-out time		Mean delay (min)	Mean gate-hold time (min)	Number of flights held
	Mean (min)	St. dev. (min)			
Baseline	18.46	8.53	0.00	0.00	0
N-Control	16.85	5.82	0.00	1.61	31,325
PRC_v3.0	16.83	5.86	0.03	1.66	28,594
Slot-Control	17.03	6.60	0.27	1.70	52,042

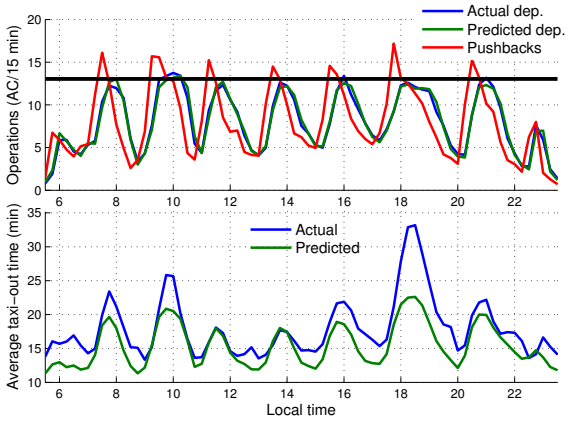

Fig. 5: N-Control simulation: Average departure capacity (in black), average number of pushbacks, average number of actual and simulated takeoffs at PHL in 2011 (top); Average actual and simulated taxi-out times (bottom).

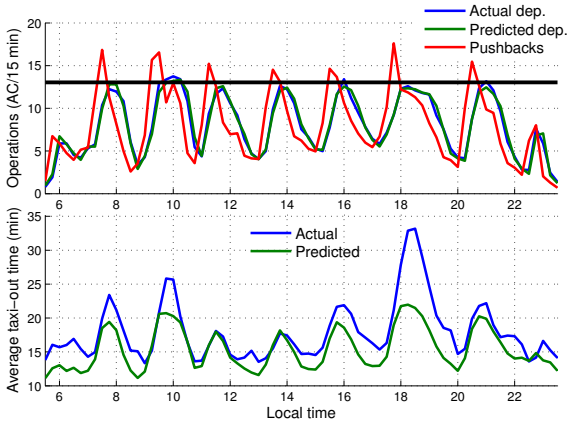

Fig. 6: PRC_v3.0 simulation: Average departure capacity (in black), average number of pushbacks, average number of actual and simulated takeoffs at PHL in 2011 (top); Average actual and simulated taxi-out times (bottom).

taxi-out time reductions. Under Slot-Control, taxi-out times are low at the beginning of the departure push, because aircraft push back at the same rate as the service rate. However, between 1800 and 1830 hours, a significant number of Heavy aircraft push back, and the departure capacity is reduced. As time progresses, aircraft arrive at the queue at rate greater than the service rate, and queuing delays increase. By contrast, both PRC_v3.0 and N-Control have delays higher than Slot-Control before 1815 hours, but they subsequently become significantly

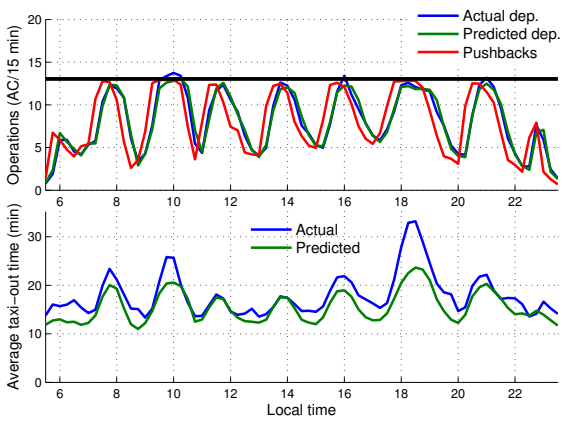

Fig. 7: Slot-Control simulation: Average departure capacity (in black), average number of pushbacks, average number of actual and simulated takeoffs at PHL in 2011 (top); Average actual and simulated taxi-out times (bottom).

lower. Between 1830 and 1930 hours, PRC_v3.0 outperforms N-Control, due to its ability to adapt to the impact of Heavy aircraft on departure throughput.

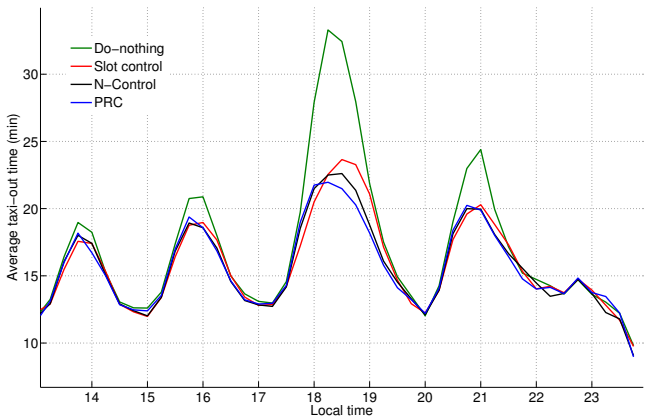

Fig. 8: Comparison of the performance of the control strategies in the evening times

VII. Conclusions

This paper proposed an airport model and a dynamic programming based approach to airport congestion control that can handle the multiple uncertainty factors that are present in airport operations. The proposed Pushback Rate Control approach, denoted PRC_v3.0, overcomes the challenges faced

by previous solutions to the problem. PRC_v3.0 used probabilistic forecasts of the departure throughput distribution and the observation of the current system state to determine the optimal rate at which aircraft pushback from their gates. Using simulations of operations at Philadelphia International airport, the performance of PRC_v3.0 is compared to a static Slot-Control strategy as well as a threshold policy known as N-Control.

The results demonstrate that PRC_v3.0 offers an effective compromise between state-dependent control and static congestion control. Congestion is efficiently managed, high runway utilization is achieved, even when pushback rates are allocated every 15 minutes. The algorithm is also shown to effectively adapt to changes in airport departure capacity.

REFERENCES

[1] I. Simaiakis and H. Balakrishnan, "Analysis and Control of Airport Departure Processes to Mitigate Congestion Impacts," *Transportation Research Record: Journal of the Transportation Research Board*, pp. 22–30, 2010.

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

[3] I. Simaiakis, "Analysis, Modeling and Control of the Airport Departure Process," Ph.D. dissertation, Massachusetts Institute of Technology, 2012.

[4] R. de Neufville and A. Odoni, *Airport Systems: Planning, Design and Management*. McGraw-Hill, 2003.

[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] 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.

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

[8] P. Burgain, "On the control of airport departure processes," Ph.D. dissertation, Georgia Institute of Technology, 2010.

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

[10] I. Simaiakis, M. Sandberg, and H. Balakrishnan, "Dynamic Control of Airport Departures: Algorithm Development and Field Evaluation," in *IEEE Transactions on Intelligent Transportation Systems*, 2013, forthcoming.

[11] 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.

[12] I. Simaiakis and H. Balakrishnan, "Departure throughput study for Boston Logan International Airport," Massachusetts Institute of Technology, Tech. Rep., 2011, No. ICAT-2011-1.

[13] N. Pyrgiotis, K. Malone, and A. Odoni, "Modelling delay propagation within an airport network," *Transportation Research Part C: Emerging Technologies*, 2011.

[14] B. Koopman, "Air-terminal queues under time-dependent conditions," *Operations Research*, vol. 20, no. 6, pp. 1089–1114, 1972.

[15] I. Simaiakis and H. Balakrishnan, "Dynamic control of airport departures: Algorithm development and field evaluation," in *American Control Conference*, 2012.