Integrated Control of Airport and Terminal Airspace Operations

Harshad Khadilkar
IBM Research

Hamsa Balakrishnan
Massachusetts Institute of Technology

***Abstract*—Airports are the most resource-constrained components of the air transportation system. This paper addresses the problems of increased flight delays and aircraft fuel consumption through the integrated control of airport arrival and departure operations. Departure operations are modeled using a network abstraction of the airport surface. Published arrival routes to airports are synthesized in order to form a realistic model of arrival airspace. The proposed control framework calculates the optimal times of departure of aircraft from the gates, as a function of the arrival and departure traffic as well as airport characteristics such as taxiway layout and gate capacity. The integrated control formulation is solved using dynamic programming, which allows calculation of policies for real-time implementation. The advantages of the proposed methodology are illustrated using simulations of Boston's Logan International Airport.**

I. Introduction

Airport surface delays are a major problem faced by air transportation systems worldwide. A steady growth in demand, coupled with the slow growth in air transportation infrastructure, has led to large increases in air traffic delays. The economic and environmental impacts of airspace and airport congestion are borne by the passengers, the airlines, and society as a whole. Government policies, public pressure and voluntary initiatives have motivated many airlines and regulatory authorities to strive for carbon-neutrality. Recent studies indicate that efficiency improvements in aviation could reduce fuel consumption by 9 million tons per year, and CO_2 emissions by 28 million tons per year [1]. Such operational improvements also have the potential to yield benefits in the near-term, compared to long-term strategies such as new aircraft designs.

Some flight delays are unavoidable, such as those due to severe weather or maintenance issues. However, approximately 17% of flight delays at the 35 major airports in the United States (US) in 2013 were attributed to high traffic volume [2]. The annual taxi-out delays (the difference between actual and unimpeded taxi-out times) at major airports in the United States are estimated to exceed 32 million minutes [3].

Air traffic control procedures in the US currently allow aircraft to depart from their gates as soon as they are ready, regardless of the congestion level in the active movement area of the airport [4]. This protocol is referred to as *push-back at pilot's discretion* in the rest of this paper. Combined with the scheduling of a larger number of departures than the capacity of the airport [5], this unrestricted policy results in a large queue of aircraft at the runway. A direct consequence of this phenomenon is an excessive amount of fuel burn, which is approximately proportional to taxi times [6]. The purpose of this work is to propose an approach that substantially reduces aircraft fuel burn on the surface, satisfies practical constraints such as arrival airspace capacity as well as airport gate capacity, and is compatible with near-term technologies and procedures.

The remainder of this paper is structured as follows. Section II describes related work in several areas relevant to this paper, including airport congestion control protocols, modeling of airport operations and network congestion control. Section III describes the system models for the airport surface as well as for the arrival airspace around an airport. Section IV formulates the control problem and describes the solution procedure for obtaining optimal control policies. Section V evaluates the proposed control strategy, and also compares its performance with current operations.

II. Related Work

Current air traffic control procedures at most airports in the United States allow aircraft to pushback from their gates as soon as they are ready for departure, and join the runway queue. This results in large taxi times during periods of peak demand, as aircraft spend a large amount of time waiting for their turn to take off [7], [8]. Several studies have shown that when an airport is experiencing congestion, holding aircraft at the gate can help reduce taxi times and fuel burn [9], [7], [10], [11]. Aircraft waiting at the gate have their engines turned off, in contrast to those in the runway departure queue. This leads to direct fuel savings. Indirect fuel savings are also realized by the gate-holding strategy since aircraft now encounter lower congestion on the surface, thus reducing their taxi times.

Better management of surface operations is one of the major objectives listed in the Federal Aviation Administration's plan for improvement of air traffic operations [12]. There have been several efforts to develop and implement surface congestion management strategies, including the field-testing of the Pushback Rate Control strategy at BOS [4], [13], the Tower Flight Data Manager (TFDM) demonstration at Dallas-Fort Worth (DFW) airport [14], the field evaluation of the Collaborative Departure Queue Management concept at Memphis (MEM) airport [15], and the Ground Metering Program at New York's JFK airport [16], [17].

While these strategies have been shown to be beneficial, several aspects of the airport congestion management problem are yet to be addressed. Firstly, the primary control objective

Harshad Khadilkar is a Research Scientist with IBM India Research Lab, Bangalore 560045, India harshad.khadilkar@in.ibm.com.

Hamsa Balakrishnan is an Associate Professor in the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA hamsa@mit.edu.

This work was supported in part by NSF through CPS:Large:ActionWebs (Award number 0931843) and FORCES (Award number 1239054).

of these prior studies was to stabilize the level of surface traffic, and to then evaluate the incidental benefits in terms of reduced taxi times and fuel burn [4]. By contrast, the problem formulation presented in this work explicitly targets these benefits. Secondly, the algorithms found in prior literature do not account for operational constraints such as the availability of gates at the airport while determining the optimal control policies. These factors are important in practice, especially when flights are being held at the gate. Arriving aircraft that are waiting for an occupied gate to be vacated can block active taxiways and/or alleyways, which is undesirable at space-constrained airports. Gate availability is included as an explicit constraint for the calculation of optimal strategies in this work.

The emphasis in prior literature on airborne control algorithms has been on computing optimal trajectories, with safety and efficiency as the objectives. Effective formulations that aim to maximize safety in free flight are rare, due to the prohibitive computational requirements associated with optimizing unconstrained aircraft trajectories. Even studies that focus on time-optimality typically solve small-scale versions of the problem, since optimization formulations for stochastic systems of realistic size quickly become computationally intractable [18], [19]. Therefore, while maintaining the current airspace structure imposes somewhat stricter constraints on airspace capacity, it is shown in [20], [21] that significant efficiency gains can still be obtained with this approach without paying the accompanying computational penalty. The arrival airspace control algorithm utilized in this paper is adapted from prior work [20], and combines distributed control in low-density airspace with centralized control in the high-density terminal areas.

The algorithms and policies proposed in this work leverage two new air traffic management technologies. Airport Surface Detection Equipment, Model-X (ASDE-X) is primarily a safety tool designed to mitigate the risk of runway collisions [22]. It uses real-time tracking of aircraft on the surface to detect potential conflicts and to monitor conformance. There is potential, however, to use the data generated by it for surface operations analysis [23], [24] and modeling of aircraft behavior. Reported parameters in ASDE-X include the position, velocity, altitude and heading of each aircraft. The update rate is 1 Hz for each individual flight track. ASDE-X data from Boston Logan International Airport is used for the illustration of the methodologies proposed in this paper. The second air traffic management technology leveraged in this paper is the Automatic Dependent Surveillance - Broadcast (ADS-B). This is a Next Generation Air Transportation System (NextGen) surveillance and communication technology, in which aircraft broadcast on-board flight information obtained using satellite navigation to ground stations or other similarly equipped aircraft via a datalink [25].

III. System Models

A. Network model of airport surface

Fig. 1 shows the runways and taxiways at BOS that are represented in the network model. The taxiways form the links of the network, and their major intersections are marked as the nodes. The taxi-out phase for an aircraft is defined to begin when an aircraft leaves the gate, and to end when it starts its takeoff roll from the runway threshold. Therefore, the origin/source nodes in the network are the ones adjoining the gates, and the destination/sink nodes are the runway thresholds. While the figure shows the union of the networks for all possible airport configurations (allocation of runways to landings and takeoffs), only one configuration is active at a time, and each aircraft has a unique source node and sink node. For example, the highlighted part of the network in Fig. 1 is active when departures are taking place from Runway 27.

An abstraction of the model for this runway configuration is illustrated in Fig. 2. Aircraft enter the network through the source nodes (sources) 1, 2, 3 and 8. Each of these sources is associated with a *buffer*, which corresponds to its gate capacity. Note that Fig. 2 does not show the actual number of gates at Boston Logan, and is only illustrative. Each gate in a buffer can be in one of three states: available (empty circle), occupied-inactive (filled square) and occupied-active (filled circle). Gates that are empty and can be occupied by arriving aircraft are said to be *available*. When an aircraft arrives at a gate, it becomes *occupied-inactive* for a period of time, while the aircraft is being serviced. Once ready to leave, the pilot calls the air traffic controller for permission to push back. The gate is then tagged as being *occupied-active*.

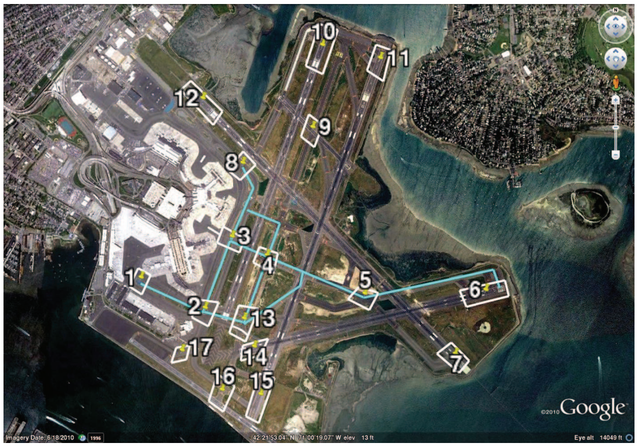

Fig. 1. Layout of the surface at BOS. Nodes in the network model are marked with white boxes. The configuration-specific network for departures from Runway 27 has been highlighted in blue.

B. Model of taxi-out time

Aircraft movement through the network described above is based on a taxi time model developed in prior work [26]. Depending on the buffer occupied while at the gate, each aircraft begins taxiing from a specific source node. It is assigned a set of connecting links in the network as its *taxi path* to the runway. The travel time t_l over each link l in the network is modeled as,

$$t_l = t_{u,l} + \sum_{i=1}^{N_{s,l}} t_{s,l,i} \tag{1}$$

This model indicates that the travel time over each link is the sum of two types of variables: (i) $t_{u,l}$, the time taken to traverse

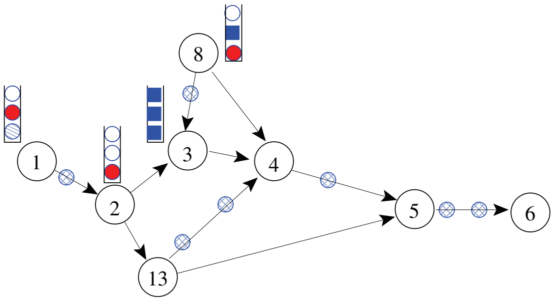

Fig. 2. Network layout for departures from Runway 27. Note that there are few alternate taxi paths from each source. Empty circles are empty gates, filled squares are occupied-inactive and filled circles are occupied-active. The circle with a one-sided hash pattern marks the aircraft that is next in line for push-back, while the double-hashed circles are aircraft that are already inside the network and are taxiing towards the runway at node 6.

the link without encountering any obstacles, and (ii) $t_{s,l,i}$, the time for which the aircraft is stationary on the link. Intuitively, the former variable is the result of the physical dynamics of the aircraft, while the latter is the result of conflicts with other aircraft on the surface. The unimpeded travel time over the link l, denoted $t_{u,l} > 0$, is modeled as an Erlang random variable with order n_l and rate λ_l. The number of stops on link l, denoted $N_{s,l} \in \{0, 1, 2, ...\}$, is modeled as a geometric random variable with parameter $p_{kl} \in [0, 1]$, where k is the number of departing aircraft already taxiing on the surface. Finally, $t_{s,l,i} > 0$ is the stationary time corresponding to the i^{th} stop on link l, modeled as an exponential random variable with rate $\mu_l > 0$. Each instance of $t_{s,l,i}$ is assumed to be independent and identically distributed.

If the number of stops is $N_{s,l} = 0$, then $t_l = t_{u,l}$. Furthermore, each instance of travel time on a link is assumed to be independent of all other instances, whether on the same link or on other links, when conditioned on the level of surface traffic. Taxi-out times at airports increase with increased surface congestion, and this needs to be accounted for by the model. The *surface traffic level*, k, is defined as the total number of departing aircraft that have pushed back from their gates but have not taken off yet. The additional taxi-out time due to congestion can be accounted for by an increase in the stopping probability on each link, p_{kl} [26]. In addition, the average taxi time on each link increases linearly with the surface traffic level [26]. Therefore, $\mathbb{E}[t_l] = a_{k,l} + b_{k,l}\, k$, where $a_{k,l}$ and $b_{k,l}$ are constants for each link l.

The taxi-out time model for a path consisting of two links is depicted as a Markov process in Fig. 3. The departure rate out of each state in Link 1 is λ_1, the corresponding Erlang rate for unimpeded travel time. The time required for n such hops is therefore equal to the unimpeded travel time on Link 1. The n^{th} transition will be to the stop state S_1 with probability p_{k1}, or to the first state of Link 2 with probability $(1 - p_{k1})$. State S_1 undergoes a self-transition with probability p_{k1} and rate μ_1, thus generating a geometric number of stops, each with an exponential distribution and rate μ_1. The same stochastic process repeats for Link 2, finally achieving the departure state D. The total taxi time for the aircraft is the time taken to move from the first state of Link 1 to state D.

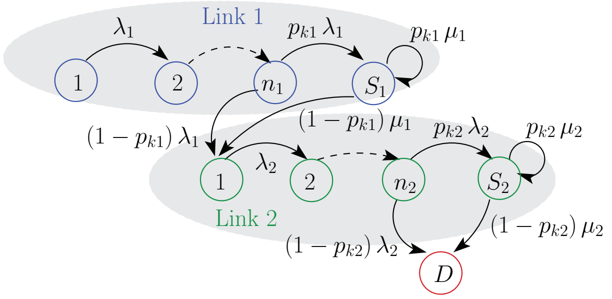

Fig. 3. Representation of the taxi-out model as a Markov process, for a path with two links. The transition time for each state is exponential with rate λ_i or μ_i. State D is the final state, when the aircraft departs from the airport.

C. Model of arrival airspace

Fig. 4 shows the layout of the different approach paths to Boston Logan International Airport. The outer boundary corresponds to a circle with a radius of 500 nm, centered at the airport. Arriving aircraft are depicted by solid circles, and traverse along the solid black lines corresponding to standard arrival procedures for Runway 33L. These aircraft are assumed to follow the communication and control architecture developed and described in detail in [20]. ADS-B messages are generated and broadcast by each aircraft after fixed intervals of time (typically every 0.5 sec). These messages are received by surrounding aircraft and by ground stations, subject to an electromagnetic propagation and loss model.

A minimum separation requirement between each pair of aircraft is imposed for safety. Assuming that the arrival procedure that each aircraft is following is known, the received ADS-B messages are used to detect potential conflicts several minutes in advance. Changes in aircraft velocity (computed either on board or by a ground facility, depending on aircraft location) are used to resolve these conflicts. Trajectory modifications (holding patterns) are avoided to the extent possible in order to maximize safety [27]. An aircraft is sent to a holding pattern (an elliptical trajectory designed to introduce separation between aircraft) only if no feasible velocity is found to resolve a projected conflict. The control algorithm can either be automatically implemented on board the aircraft involved in a potential conflict, or it can provide conflict resolution advisories to the pilot and the controllers. The net effect of the control strategy is meant to be (i) to minimise the airborne flight time of arrivals, (ii) to smooth the delivery of arriving aircraft to the airport, and (iii) to enable accurate estimates of aircraft landing times.

The specific instantiation of the control strategy in [20] divides the airspace around the airport into two parts: a small region of centralized control immediately surrounding the airport, and an outer, larger region with decentralized control. In the centralized region, surveillance is conducted by ADS-B ground stations and radar systems. Ground stations receive state information about aircraft that are within range, through their ADS-B transmissions. The radars scan through 360° of azimuth and send information to central facilities. Each radar interrogates aircraft transponders within its range, which respond using a directional antenna. The central facility

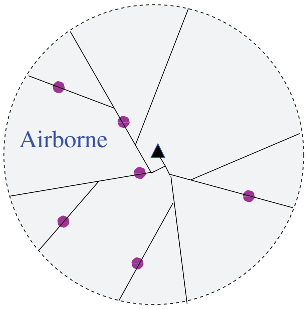

Fig. 4. Model of the arrival airspace around BOS (depicted by a triangle). The solid circles are aircraft on approach to the airport.

calculates velocities for all aircraft in the centralized zone by estimating the current state of the airspace, based on the last known location and velocity of each aircraft. Expected landing times are calculated for each aircraft, thus generating a priority order for the centralized zone. Conflict detection is carried in a pairwise fashion for each pair of aircraft, starting with the aircraft that has the highest priority.

Outside the centralized zone, the control algorithm uses local information received from ADS-B transmissions. Each ADS-B message is assumed to include a time stamp, and the maximum and minimum achievable velocities of the aircraft. When two aircraft in this region receive broadcasts from each other for the first time, they calculate a pairwise order based on their projected arrival times at the eventual merge point. If aircraft A projects itself as arriving before B at the merge point, it only notes the presence of B but does not adjust its velocity. If it projects that aircraft B will arrive at the merge point first, it computes a new velocity for itself in order to not conflict with B, while still flying as fast as possible. If B is also outside the centralized zone, it carries out a complementary set of calculations on detecting A for the first time. Since each pair of aircraft decides on a mutual order at the merge point, a unique ordering of all aircraft heading to a given merge point in the airspace is developed.

By collating the expected times of arrival of each aircraft at the merge points, an estimate of the number of landings at the airport over a given period of time can be formed. Airborne aircraft are assumed to have priority over aircraft on the ground, and therefore, the landing rate is assumed to be unaffected by surface traffic. This estimated airport landing rate as an input for determining an optimal control policy for aircraft on the surface.

IV. Integrated Control Algorithm

A. State definition

Fig. 5 shows a functional representation of the network layout depicted in Fig. 2. The purpose of the control formulation is to calculate optimal delays for aircraft that are active (ready for push back). Aircraft are allowed to push back on a first-come-first-served (FCFS) basis. That is, aircraft push backs are approved in the same order in which gates change from occupied-inactive to occupied-active. Only the first aircraft in the FCFS order is assigned a precise delay; the next aircraft is assigned a delay when the first aircraft leaves its gate.

The optimal control policy (for push back delay u) is a function of the state of the airport at the instant of calculation. This state is defined by three quantities: the level of surface traffic k, the set of available buffer capacities $\bar{N}_i$, and the source s_{next} for which the calculation is being carried out. The expected taxi time for a given aircraft depends on its taxi route, and hence on the source s_{next}. The full state representation is $\phi = (k, \bar{N}_1, \dots, \bar{N}_{n_s}, s_{\text{next}})$. The surface traffic level k is assumed to satisfy $k \in \{0, 1, 2, \dots, k_{\max}\}$, where $k_{\max}$ is a large but finite traffic level which is never exceeded in operation. The available buffer capacities $\bar{N}_i$ are equal to the number of empty gates at each source and are finite, non-negative integers. If $N_{i,\max}$ is the maximum gate capacity of source i, then $0 \leq \bar{N}_i \leq N_{i,\max}$. In addition, if N_i is the number of occupied gates (active and inactive), $N_i + \bar{N}_i = N_{i.\max}$ at all times. The source s_{next} for which the calculation is being carried out can take values from the set $\mathbb{S}$, which contains a list of the source nodes in the network.

The available buffer capacities $\bar{N}_i$ represent an implicit constraint on the delay assigned to the current aircraft, by determining the probability of buffer overflow. A buffer overflow occurs if an arriving aircraft finds no available empty gates, that is, $\bar{N}_i = 0$ for the relevant source. An estimate of the arrival rate β_i to each buffer can be used to determine the probability of this event as a function of the proposed delay u. The rate of aircraft arrivals to the airport is estimated from the landing time predictions, as explained in Section III-C. The departure control algorithm assumes that the airport arrival process is Poisson with a rate β proportional to the number of expected arrivals over a fixed time horizon. This aggregate arrival process is then split into sub-processes of rate β_i for each buffer, assuming that the rates are proportional to the gate capacities $N_{i,\max}$.

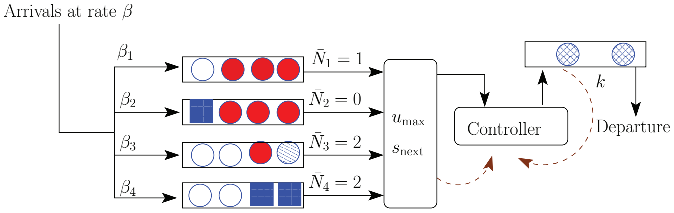

Fig. 5. Illustration of the departure control algorithm for calculating optimal gate delays.

B. Calculation of state transition probabilities

Since the arrival and departure processes at the airport are both stochastic, it is necessary to calculate the transition probabilities from one state to another, given the assigned push back delay. The evolution of s_{next} is driven by the push back order, which is decided on a first-come-first-served basis. The transitions between the buffer states $\bar{N}_i$ are governed by Poisson processes of rate β_i. Calculation of the transition probabilities between surface traffic levels is more complex due to the aggregate nature of these state variables. Aircraft that are already taxiing-out may be distributed randomly on the airport surface at any time, and it is difficult to develop exact analytical expressions for the remaining time to departure for each aircraft. Although Monte Carlo simulations with randomized push back policies can yield empirical estimates, they are highly time-intensive.

Approximate estimates of the transition probabilities for surface traffic levels are obtained using the model depicted in Fig. 3. The expected time to transition to state D is calculated by assuming that an observer appears at a random time [28]. The departure of each aircraft from the airport is then approximated by an exponential process with mean equal to the expected time required to reach state D. Since there are k aircraft on the surface, the combined departure process from the airport is Poisson with the same rate, and with order k. Since each component of the full state $\phi = (k, \bar{N}_1, \ldots, \bar{N}_{n_s}, s_{\text{next}})$ is independent of the others, the full transition probability is equal to the product of the transition probabilities for each component.

C. State aggregation

The full state representation ϕ defined in Section IV-A results in a very large number of states for a realistic airport. There are approximately 90 gates at BOS divided among four terminals, two of which have a capacity of 25 gates each, and two with a capacity of 20 gates each. A reasonable surface traffic model would need to account for a maximum surface traffic level of at least 25 aircraft. Combined with four possible sources for each aircraft, the total number of states exceeds 31×10^6. In order to make the calculation of optimal control policies tractable, the size of the problem is reduced by using state aggregation.

The set of buffer states, $\bar{N}_i$, encapsulates the risk of buffer overflow as a function of the assigned push back delay. An alternative parameterization of this risk is the maximum acceptable push back delay assigned for a given instance of $\{\bar{N}_1, \bar{N}_2, ...\}$. The state can then be redefined to be $\theta = (k, u_{\max}, s_{\text{next}})$, where $u_{\max}$ is the maximum push back delay. Suppose γ is the *buffer overflow tolerance*, or the maximum probability with which the capacity of at least one buffer is exceeded. The value of $u_{\max}$ equals the value of delay at which the probability of there being more than $\bar{N}_i$ arrivals in at least one of the Poisson processes with rate β_i exceeds γ. For ease of policy calculation, the resulting value of $u_{\max}$ is rounded off and mapped to a discrete finite set $\mathbb{U}$. There is a unique mapping from the available buffer capacities $\bar{N}_i$ to $\mathbb{U}$, as a function of γ and β_i. Consequently, there is also a unique mapping from states ϕ to states θ. Transition probabilities $p_{\theta_1\theta_2}(u)$ between the aggregate states θ are calculated by summing over all the transitions $\phi_1 \to \phi_2$ that correspond to transitions $\theta_1 \to \theta_2$. Combined with the stage cost definition developed in Section IV-D, a dynamic programming problem is formulated to calculate the optimal push back policy $u(\theta)$.

D. Cost definition

The calculation of the optimal push back policy for each state θ requires the knowledge of state transition probabilities and a definition of stage costs. The expected stage cost $\mathcal{C}$ for the aircraft/airport system is illustrated in Eq. (2). It consists of three terms: the fuel burn for each aircraft, the fuel burn for auxiliary power while aircraft are at the gate, and the airport throughput loss due to low surface traffic. The auxiliary power units are used to drive onboard aircraft systems such as radios and air conditioning when engine power is unavailable. The first and third terms in the cost function depend on the projected traffic level k_p at the assigned time of push back, which is in turn a function of the current state θ_1 and the push back delay u. The value of k_p can be calculated using the method described in Sections IV-B and IV-C. As explained in Section III-B, the expected taxi time increases linearly with the surface traffic level k_p, and is given by

$$\mathbb{E}\left[\sum_{l\in\mathcal{P}} t_l\right] = \left(\sum_{l\in\mathcal{P}} a_{k,l}\right) + \left(\sum_{l\in\mathcal{P}} b_{k,l}\right) k_p,$$

where $\mathcal{P}$ is the taxi path. The average separation between successive departures at this traffic level is

$$\frac{\mathbb{E}\left[\sum_{l\in\mathcal{P}} t_l\right]}{k_p} = \left(\sum_{l\in\mathcal{P}} a_{k,l}\right)\frac{1}{k_p} + \left(\sum_{l\in\mathcal{P}} b_{k,l}\right).$$

The minimum average inter-departure separation is achieved as $k_p \to \infty$, and the penalty for maintaining a finite traffic level is proportional to $\frac{1}{k_p}$.

The resultant cost function is therefore given by:

$$\mathcal{C}(\theta_1, u) = \sum_{\theta_2} p_{\theta_1\theta_2}(u)\left[c_1\, k_p(\theta_2) + c_2\, u + \frac{c_3}{k_p(\theta_2)}\right]. \quad (2)$$

The first term in Eq. (2) is a measure of the aircraft taxi time, and is therefore also a measure of the fuel burn during taxi out [6]. The second term captures the fuel cost of using auxiliary power while at the gate, and is proportional to the push back delay u. The third term is proportional to the difference between the ideal runway performance and the actual expected performance, as defined by the expected time between successive departures.

E. Calculation of optimal policies

Following the state and cost definition and the aggregation procedure described in Section IV-C, the optimal push back delays can be calculated using the aggregate states θ. The transition probabilities $p_{\theta_1\theta_2}(u)$ in Eq. (3) are calculated using the methods outlined in Sections IV-B and IV-C. $\mathbb{U}(\theta_1)$ is the set of available push back delays for state θ_1. $J(\theta)$ is the optimal cost-to-go from state θ. Finally, α is the discount factor, which defines the weight on future costs with respect to the expected cost for the current transition. The Bellman equation for the infinite-horizon discounted cost problem is given by

$$J(\theta_1) = \min_{u\in\mathbb{U}(\theta_1)}\left(\mathcal{C}(\theta_1, u) + \alpha\sum_{\theta_2} p_{\theta_1\theta_2}(u)\, J(\theta_2)\right). \quad (3)$$

The infinite-horizon formulation is chosen as an approximation to the relative time scales of assigned push back delays and airport demand variations. While push back delays are of the order of a few minutes, demand variations at busy airports occur over the period of a few hours. The discounted-cost structure of the formulation provides numerical stability in the calculation of optimal policies [29]. In order to place sufficient emphasis on future costs, the value of α is set close to 1. This paper uses $\alpha = 0.99$, unless otherwise stated. Eq. (3) is solved directly using matrix inversion.

The resultant optimal policies for aircraft leaving from source 1 are shown in Fig. 6. Each curve corresponds to a different value of maximum push back delay, $u_{\max}$. As the level of surface traffic increases, the assigned push back delay increases, up to the maximum delay allowed. Similar policies can be calculated for the three remaining source nodes. Note that the number of states mapping into each of these aggregate curves changes, depending on the values assigned to c_1, c_2, c_3, α and γ. In an airport implementation, only the current surface traffic level and the gate occupancy are needed to calculate the optimal push back delay.

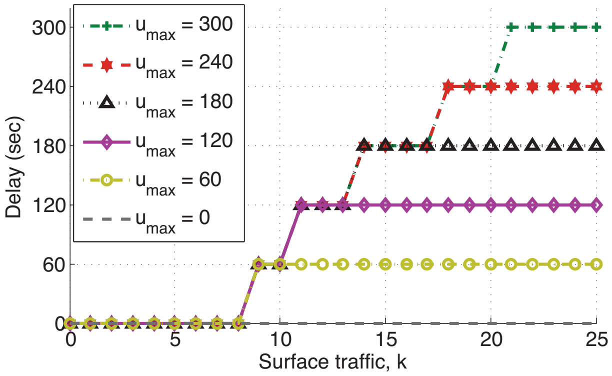

Fig. 6. Optimal push back delays at Boston Logan, calculated for departures from source 1. The total airport landing rate β is assumed to be 0.011, or 1 aircraft every 90 seconds.

F. Combined control of departures and arrivals

The airspace capacity can by estimated through simulations of the model shown in Fig. 4, by varying the traffic generation rates β', and observing the resultant landing rate, β. If the incoming traffic rate results in a stable airspace network, steady state is achieved with $\beta = \beta'$. The throughput of the airspace model is characterized by plotting the number of arrivals at the airport as a function of the number of airborne aircraft within the modeled airspace (Fig. 7). This metric is more useful than a plot of β as a function of β', since it is easier to count the number of aircraft in a given region of airspace than it is to count their rate of arrival at the periphery.

The estimated steady state arrival throughput in Fig. 7 is approximately 10 aircraft every 15 min or 90 sec between landings, and is equal to the traffic generation rate. In other words, the rate $\frac{1}{\beta'} = 90$ sec is feasible for the current airspace layout. In the case of the BOS model with arrivals on Runway 33L, the maximum capacity is found to be 1 aircraft every 70 sec. This methodology allows us to gauge the maximum rate at which the airborne portion of the model is likely to deliver aircraft to the airport surface. This estimate determines the arrival rates β, and hence the set of policies that are calculated for the departure control algorithm as described in Section IV-E. The optimal policy used in real-time is selected by considering the number of expected arrivals over a fixed time horizon, given by the current arrival rate β. The resultant control policy is expected to be more aggressive (that is, assign larger gate delays) when the expected β is low, and vice versa.

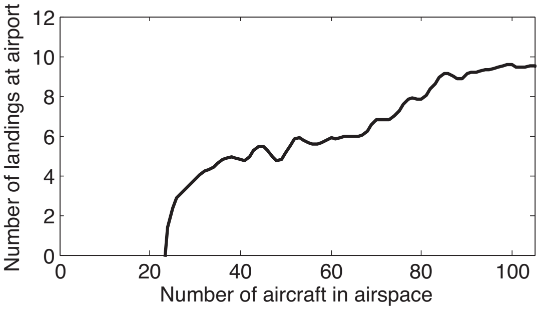

Fig. 7. Arrival throughput characteristics at Boston Logan. The y axis shows the average number of landings in a 15 min interval, when the corresponding number of active aircraft is on the x axis.

V. Performance Evaluation

A. Simulation procedure

The arrival airspace model described in Section III-C and the surface taxi model described in Section III-B are combined in the integrated simulations, as shown in Fig. 8. Incoming airborne traffic is generated at the periphery of the airspace and propagated to the airport. The airport predicts landing times for all aircraft that are within ADS-B broadcast reception range. These landing time predictions are fed to the departure control algorithm, which also predicts the taxi-in times and forms an estimate of the arrival rate to each buffer, β_i. These arrival rates, in combination with the departure surface traffic level k, drive the push back control policy. The surface traffic simulation generates stochastic taxi-in and taxi-out times for arrivals and departures, respectively.

Aircraft in the arrival airspace are assumed to use approach paths for landing on Runway 33L at the airport, as shown in Fig. 4. The surface movement simulation includes aircraft taxiing in to their gates after landing, gate occupancy at the airport, as well as aircraft taxiing out to Runway 27 (node 6 in Fig. 1) for takeoff. This runway is used for departures at Boston when aircraft land on Runway 33L (node 7 in Fig. 1). The four source nodes at the airport (nodes 1, 2, 3 and 8 in Fig. 1) have capacities of 25, 20, 25 and 20 gates respectively. An arrival is assumed to be able to park at any gate in its assigned terminal. While this assumption is convenient for demonstrating the proposed algorithm, in reality, each airline has access to only a specific subset of gates at a terminal. The formulation can be extended to accommodate this constraint by defining a node for each set of gates used by an airline.

B. Simulation results

A simulation of one day's operations at BOS is shown in Fig. 9 and Fig. 10. Aircraft appear at the periphery of the

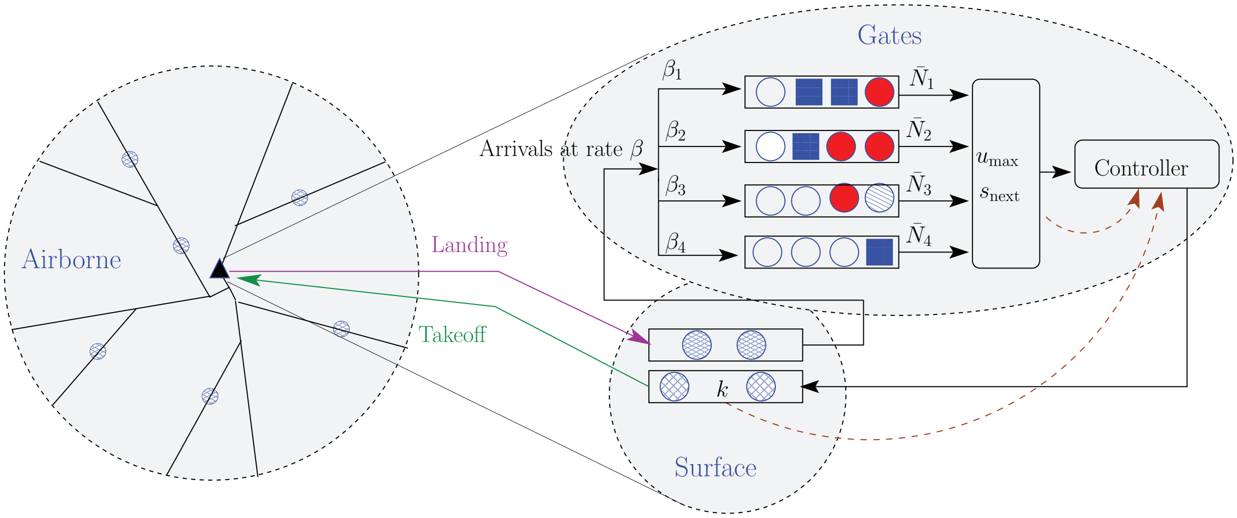

Fig. 8. Schematic of the combined control algorithm and simulation procedure.

airspace as a variable rate Poisson process. The governing rate is piecewise linear, and is based on the historical average seen for various times of the day. Fig. 9 (top) shows that the peak morning demand appears at the periphery of the airspace at approximately 03:00 local time. These aircraft reach the airport just before 06:00, when the number of landings increases to about 10 aircraft every 15 minutes, or an average separation of 90 seconds between successive landings. The surface simulator then propagates these arrivals through the airport network to their gates, based on the taxi time model described in Section III-B. A gate service time of between 30 and 45 minutes is generated for each arrival from a uniform distribution. During this time, the gate is marked occupied-inactive. Once servicing is completed, the aircraft is ready for push back and the gate is marked occupied-active. The occupied-active gates are assigned push back delays on a first-come-first-served basis.

After push back, the simulation propagates each aircraft to Runway 27 for departure. As seen in Fig. 9, the morning departure demand trails the arriving one by approximately 45 minutes. The departure rate at the airport is seen to stabilize at 10 aircraft every 15 minutes. Fig. 9 (middle) shows the average simulated taxi times for arrivals and departures. Since arrivals are not subject to the higher traffic levels seen by departures (Fig. 9, bottom), average arrival taxi times are significantly lower than departure taxi times. This observation is supported by empirical data. Fig. 9 (bottom) also shows a spike in the departure traffic level at 20:00, which can be correlated with the gate occupancy plotted in Fig. 10. A disproportionately large number of aircraft arrive at sources 2 and 3 just before 20:00, pushing the gate occupancy close to the maximum limit. The control algorithm responds by releasing aircraft with very small gate delays, thereby temporarily increasing the surface traffic level. In this manner, surface congestion is balanced against the risk of buffer overflow.

C. Effect of control strategy on taxi times and gate delays

Current airport procedures allow pilots to push back at their discretion, which means that each aircraft can leave the gate as soon as its servicing is completed. Therefore, the effect of the proposed control strategy can be evaluated by carrying out different simulations on the same 'push back ready' schedule. The first simulation assigns zero push back delay to all aircraft. The second simulation assigns push back delays to each aircraft, as calculated by the departure control algorithm. Finally, the third simulation incorporates information about expected arrival times of aircraft to each buffer in assigning pushback delays.

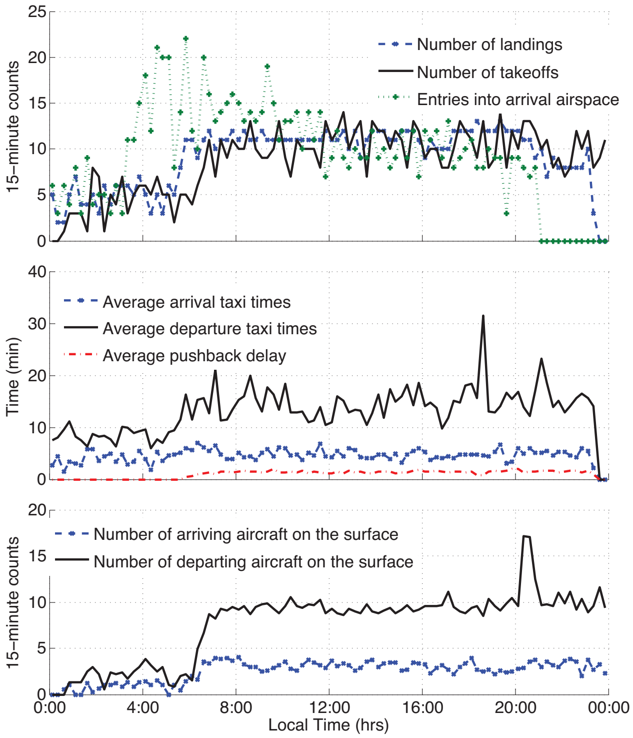

Fig. 9. Operation counts, average taxi and push-back delay times, and surface counts in a full day's simulation.

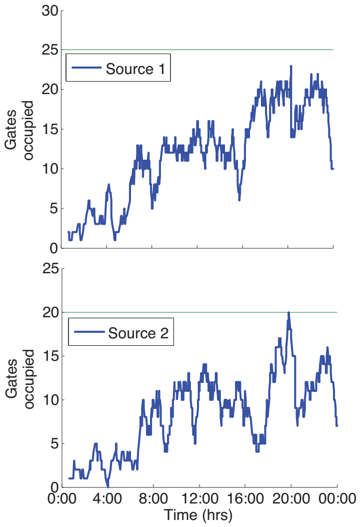

Fig. 10. Gate occupancy over the course of the simulated day, at each of the four sources (airport terminals).

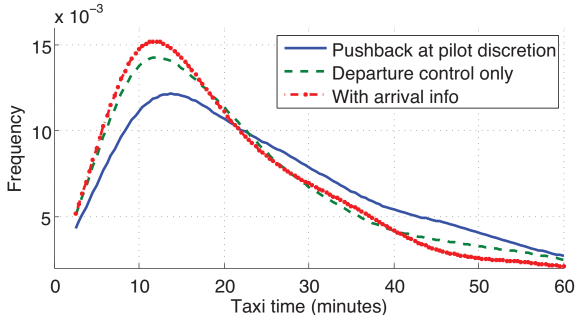

Fig. 11. Comparison of the distribution of taxi-out times for three different control strategies.

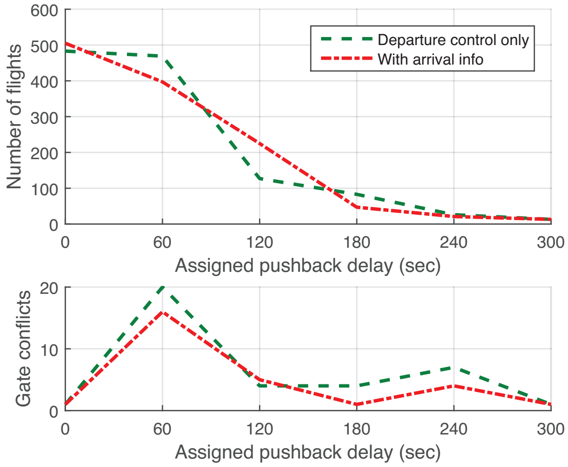

Fig. 12. Comparison of the distribution of assigned gate delays, with and without arrival information.

The distributions of aircraft taxi-out times from the three simulations are compared in Fig. 11. The base schedule used for this study is the same as that generated in Section V-B. The frequency of large taxi-out times is greatly reduced by the control strategy, which results in substantially lower taxi-out times. Incorporating predictions of arrival times further lowers the taxi-out times, by allowing the control algorithm to be more aggressive when the arrival rates are low. The mean simulated taxi-out time reduction from the departure control algorithm is 1.7 minutes per aircraft. The use of arrival information further reduces the mean taxi-out time by 1.5 minutes per aircraft. The estimated fuel burn reduction per medium-sized aircraft is 25 kg (9 gallons) for the departure control algorithm, and a total of 45 kg (16 gallons) per aircraft after including arrival information [6].

The corresponding gate delay assignments in the simulation are depicted in Fig. 12. This figure highlights two major features of the integrated control strategy. The first feature is that the trend in gate delays is smoother when arrival information is included. This indicates that there is a reduction in the number of unexpected events (such as gate conflicts), and therefore, better management of upcoming periods of high demand. The second feature is the reduction in number of gate conflicts, as depicted in the lower half of the figure. The largest improvement is delivered for a gate hold of 60 sec, which is assigned when the airport is entering high congestion states (and is therefore at high risk of gate conflicts).

A comparison of the average taxi-out times and frequency of gate conflicts seen for the three strategies is given in Table I. These averages are calculated over 10 simulation runs, with all three control strategies using the same push back schedule for a given simulation run. The buffer overflow tolerance γ is set to 5%. It can be seen that the departure control strategy reduces mean taxi-out times per aircraft at BOS by 5.2%. The departure control strategy with arrival information reduces taxi times by 10.2% compared to current procedures. Since holding aircraft at the gate increases the buffer occupancy, there is an increase in the number of gate conflicts (that is, instances of buffer overflow). However, the fraction of aircraft with gate conflicts is considerably less the tolerance level γ, possibly because of the simplifying assumptions made while calculating state transition probabilities in Section IV-B. It should also be noted that including arrival information reduces the number of gate conflicts by 30% in addition to reducing the mean taxi-out times. Depending on user preference, the expected number of gate conflicts can be traded off with taxi-out time reductions, using the tolerance level γ.

TABLE I
COMPARISON OF TAXI TIMES AND FREQUENCY OF GATE CONFLICTS. THE AVERAGES ARE CALCULATED OVER 10 SIMULATION RUNS. THE BUFFER OVERFLOW TOLERANCE IS $\gamma = 5\%$.

Control strategy	Avg. taxi-out time (min)	Avg. # gate conflicts/day	% flights with gate conflicts
Pilot's discretion	34.3	14	0.5%
Departure control only	32.5	33	2.7%
With arrival info	30.8	24	1.9%

D. *Effect of control strategy on takeoff delay*

The control strategy aims to not increase aircraft takeoff times beyond those attained using current procedures, thereby ensuring that the reduced fuel burn is not at the cost of airport performance. Fig. 13 shows the distribution of simulated takeoff delay (the difference between the takeoff times under a push back control strategy and under current procedures) for the two control strategies. The relative takeoff times are seen to be distributed equally on both sides of zero takeoff delay. The differences are therefore likely to be due to random error only. Over the course of one day's simulation, the mean takeoff delay is in fact negative, meaning that on average, aircraft will take off earlier than they do under current procedures. The average decrease in takeoff times is 0.75 minutes and 1.25 minutes under departure control without, and with arrival information, respectively.

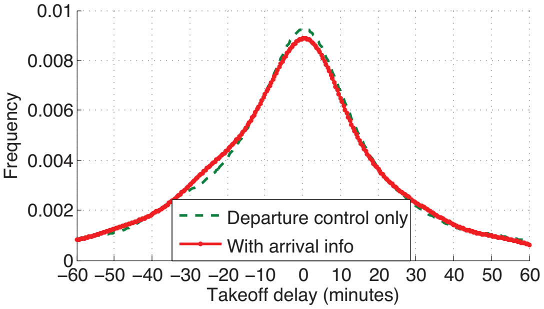

Fig. 13. Comparison of the distributions of takeoff times relative to those attained using current procedures.

VI. Conclusion

The paper proposed a paradigm for the management of aircraft operations in and around airports, with the focus on practical implementation in the future. The objective of the control strategy was to reduce aircraft fuel consumption, as well as congestion on the airport surface and in arrival airspace. At the same time, it was required to satisfy constraints on system performance and safety. The proposed methodology used several novel ideas in the realm of surface congestion control, including modeling the airport surface network using surface surveillance data. The departure control algorithm included relevant constraints such as limited gate capacity at the airport. Dynamic programming, with a state aggregation procedure, was used to determine the optimal control policy.

The arrival control algorithm assumed that the primary communications system would be Airborne Dependent Surveillance Broadcast (ADS-B). The strategy could then be implemented by transmitting commands over a data-link and displaying them as advisories to pilots. An integrated control strategy was then developed by combining arrival airspace and surface operations. This strategy considered various aspects of airport operations, and departures were released from their gates depending on surface traffic levels as well as expected aircraft arrival rates. The resultant policy balanced fuel burn reduction and the constraints imposed by airport gate capacity.

Simulations showed that the control strategy could substantially reduce aircraft taxi times and fuel burn. An average of 10% reduction in taxi times as compared to current procedures was noted. This corresponds to a 3.5 min reduction in taxi-out time per aircraft, equivalent to 10 gallons of aviation fuel. The implementation requires only the knowledge of the surface traffic level and the gate occupancy at each terminal, both of which are available in real-time. Although the simulations and results shown in this paper corresponded to one configuration at Boston Logan International Airport, they can be easily extended to other runway configurations, airports, and operating conditions. Airspace constraints imposed by operations at nearby airports can also be accounted for, by imposing suitable constraints on available approach paths. It is believed that significant airport efficiency improvements will still be achieved in a wide range of scenarios.

References

[1] "Revolutionising air traffic management," tech. rep., Air Transport Action Group, Geneva, Switzerland, November 2012.

[2] Federal Aviation Administration, "The operations network." http://aspm.faa.gov/OPSNET. Accessed February 2013.

[3] Federal Aviation Organization, "Aviation System Performance Metrics database." http://aspm.faa.gov/main/aspm.asp. Retrieved December 2010.

[4] I. Simaiakis, H. Khadilkar, H. Balakrishnan, T. G. Reynolds, R. J. Hansman, B. Reilly, and S. Urlass, "Demonstration of reduced airport congestion through pushback rate control," in *USA/Europe Air Traffic Management Research and Development Seminar*, (Berlin, Germany), June 2011.

[5] C. Barnhart, D. Fearing, A. Odoni, and V. Vaze, "Demand and capacity management in air transportation," *European Journal of Transportation and Logistics*, vol. 1, pp. 135–155, 2012.

[6] H. Khadilkar and H. Balakrishnan, "Estimation of aircraft taxi fuel consumption using flight data recorder archives," *Transportation Research Part D*, vol. 17, pp. 532–537, October 2012.

[7] H. Idris, B. Delcaire, I. Anagnostakis, W. Hall, N. Pujet, E. Feron, R. J. Hansman, J.-P. Clarke, and A. Odoni, "Identification of flow constraint and control points in departure operations at airport systems," in *AIAA Guidance, Navigation and Control Conference*, (Boston, MA), August 1998.

[8] M. Sandberg, H. Khadilkar, T. Reynolds, and H. Balakrishnan, "Airport characterization for the adaptation of surface congestion management approaches," in *USA/Europe Air Traffic Management Research and Development Seminar*, (Chicago, IL), June 2013.

[9] E. Feron, R. J. Hansman, A. R. Odoni, R. Cots, B. Delcaire, W. Hall, H. Idris, A. Muharremoglu, and N. Pujet, "The departure planner: A conceptual discussion." White paper, MIT, International Center for Air Transportation, December 1997.

[10] N. Pujet, B. Delcaire, and E. Feron, "Input-output modeling and control of the departure process of congested airports," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, (Portland, OR), August 1999.

[11] P. Burgain, *On the control of aircraft departure operations*. PhD thesis, Georgia Institute of Technology, November 2010.

[12] Federal Aviation Administration, *NextGen Implementation Plan*, March 2012.

[13] I. Simaiakis and H. Balakrishnan, "Dynamic control of airport departures: algorithm development and field evaluation," in *American Control Conference*, (Montréal, Canada), June 2012.

[14] H. Reynolds, M. Ishutkina, D. Johnson, R. Jordan, M. Kuffner, K. Lokhande, T. Reynolds, and S. Yenson, "Dallas/Fort Worth field demonstration 2 final report for Tower Flight Data Manager (TFDM)," tech. rep., Project Report ATC-393, MIT Lincoln Laboratory, Lexington, MA, 2012.

[15] 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 *USA/Europe Air Traffic Management Research and Development Seminar*, (Berlin, Germany), June 2011.

[16] A. Nakahara, T. Reynolds, T. White, and R. Dunsky, "Analysis of a surface congestion management technique at New York JFK airport," in *AIAA Conference on Aviation Technology, Integration and Operations*, (Virginia Beach, VA), September 2011.

[17] S. Stroiney, H. Khadilkar, B. Levy, and H. Balakrishnan, "Assessing the impacts of the JFK ground management program," in *Digital Avionics Systems Conference*, (Syracuse, NY), October 2013.

[18] L. Pallottino, E. M. Feron, and A. Bicchi, "Conflict resolution problems for air traffic management systems solved with mixed integer programming," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 1, pp. 3–11, 2002.

[19] C. E. van Daalen and T. Jones, "Fast conflict detection using probability flow," *Automatica*, vol. 45, no. 8, pp. 1903–1909, 2009.

[20] P. Park, H. Khadilkar, H. Balakrishnan, and C. Tomlin, "Hybrid communication protocols and control algorithms for NextGen aircraft arrivals," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, pp. 615–626, April 2014.

[21] P. Park, H. Khadilkar, H. Balakrishnan, and C. Tomlin, "High confidence networked control for next generation air transportation systems," *to appear in IEEE Transactions on Automatic Control*, 2014.

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

[23] H. Khadilkar and H. Balakrishnan, "Metrics to evaluate airport operational performance using surface surveillance data," *Air Traffic Control Quarterly*, vol. 21, August 2013.

[24] M. Sandberg, "Applications of ASDE-X data to the analysis of airport surface operations," Master's thesis, Massachusetts Institute of Technology, 2012.

[25] RTCA, Inc., *Minimum Aviation System Performance Standard for Automatic Dependent Surveillance Broadcast (ADS-B)*, 2002.

[26] H. Khadilkar and H. Balakrishnan, "Network congestion control of airport surface operations," *Journal of Guidance, Control and Dynamics*, vol. 37, pp. 933–940, May-June 2014.

[27] M. Lupu, E. Feron, and Z.-H. Mao, "Traffic complexity of intersecting flows of aircraft under variations of pilot preferences in maneuver choice," in *Conference on Decision and Control*, (Atlanta, GA), December 2010.

[28] R. Gallager, *Discrete Stochastic Processes*. Springer, 1996.

[29] D. Bertsekas, *Dynamic Programming and Optimal Control, Volume I*. Athena Scientific, 3 ed., 2005.

Harshad Khadilkar is a Research Scientist with IBM India Research Lab, Bangalore, India. He received his PhD in Aeronautics and Astronautics from the Massachusetts Institute of Technology in 2013. He received his Bachelors degree in Aerospace Engineering from the Indian Institute of Technology, Bombay. His research interests include algorithms for optimizing air traffic operations, and stochastic estimation and control.

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 is in the design, analysis, and implementation of control and optimization algorithms for large-scale cyber-physical infrastructures, with an emphasis on air transportation systems. She received the US National Science Foundation's CAREER Award in 2008, the Kevin Corker Award for Best Paper of the USA/Europe Air Traffic Management Seminar in 2011, the American Institute of Aeronautics and Astronautics' Lawrence Sperry Award in 2012, and the American Automatic Control Council's Donald P. Eckman Award in 2014.