A Decision Support Tool for
Pushback Rate Control of Airport Departures

Melanie Sandberg, Ioannis Simaiakis,
Hamsa Balakrishnan, Tom G. Reynolds and R. John Hansman

Abstract

Airport surface congestion control has the potential to mitigate the increase in taxi times and fuel burn at major airports. This paper describes the implementation of a congestion control strategy at Boston Logan International Airport (BOS). The approach predicts the departure throughput in the next 15 minute interval, and recommends a rate at which to release pushbacks from the gate in order to control congestion. During the first phase of field-testing in 2010, the recommended rates were communicated to the air traffic controllers using color-coded cards. In 2011, two Android tablet computers were used for the implementation of the strategy; one to input the data, and the other to display the recommended rate to the air traffic controllers. Two potential decision-support displays were tested: a rate control display that only presented a color-coded suggested pushback rate, and a volume control display that provided additional support to the controllers on the number of aircraft that had called-ready and had been released. A survey of controllers showed that they had found the decision-support tool easy to use, especially the additional functionality provided by the volume control display. The field tests were also found to yield significant operational benefits, showing that such a congestion control strategy could be effective in practice.

I. INTRODUCTION

Airport operations are known to have significant impacts on the local air quality near major airports [1–4]. Surface congestion is a major reason for aircraft emissions at airports, while also increasing fuel burn and taxi delays. As a consequence, there is considerable potential to reduce these impacts through the implementation of surface congestion management strategies. This paper describes an effort to develop and evaluate an air traffic controller decision support tool for airport surface congestion control.

M. Sandberg and T. G. Reynolds are with MIT Lincoln Laboratory. I. Simaiakis is with McKinsey & Company, Washington, D.C. H. Balakrishnan and R. J. Hansman are with the Massachusetts Institute of Technology.

Corresponding author’s email: hamsa@mit.edu.

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).

Manuscript received June 20, 2013.
The *N-Control* strategy is a simple airport congestion control policy aimed at reducing surface congestion. It relies on the virtual queue concept that was first proposed in the Departure Planner, and that has been extensively studied since [5–9]. N-Control is a threshold strategy based on the typical variation of departure throughput with the number of active 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, 11]. These departure throughput saturation curves can be determined for any airport/configuration/weather combination. Figure 1 illustrates this behavior for the most frequently used runway configuration at Boston Logan International Airport (BOS) in 2011, using data from a surface surveillance system known as ASDE-X [12].

![Empirical mean and std dev](image)

**Fig. 1**: Empirical variation of departure throughput with the number of aircraft taxiing out, for the 22L, 27 (arrival runways) | 22L, 22R (departure runways) configuration at BOS, under Visual Meteorological Conditions (VMC) [11].

In N-Control, if the total number of departing aircraft on the ground exceeds a certain threshold denoted $N_{\text{ctrl}}$, 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 $N_{\text{ctrl}}$. While the value of $N_{\text{ctrl}}$ must be large enough to maintain runway utilization, too large a value will reduce the benefits of the control strategy. This strategy is similar to the concept of an Acceptable Level of Traffic (ALOT), which is employed by Air Traffic Controllers at some airports in times of extreme congestion [13].

**II. CONTROL STRATEGY DESIGN REQUIREMENTS: THRESHOLD POLICY VS. RATE CONTROL**

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.

While a threshold strategy such as N-Control is simple, it presents implementation challenges. It became evident from conversations with air traffic controllers at Boston that a recommended pushback rate valid over some extended period of time was much preferred to an on-off threshold strategy, which would require constant intervention. The
recommended rate would then be updated periodically based on the state of the system. Such a strategy is referred to as Pushback Rate Control (PRC).

A good choice for the length of the time period over which the recommended pushback rate is valid is 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 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. For the case of BOS, a time period of 15 min was selected based on the above considerations. In other words, immediately prior to the start of each 15 min period, the recommended pushback rate for that time period was determined.

Careful monitoring of off-nominal events and constraints is also necessary for implementation at a particular site. Particular concerns were 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, props were exempt from Pushback Rate Control [14].

III. DETERMINATION OF RECOMMENDED PUSHBACK RATE

There are two possible approaches to determining the recommended pushback rate. The first corresponds to an adaptation of the N-Control strategy (PRC_v1 [11]), while the second determines the optimal pushback rate using dynamic programming (PRC_v2 [15]).

A. PRC_v1

In this variant of Pushback Rate Control, the pushback rate is determined so as to keep the number of jets taxiing out near a suitable value ($N_{cut}$), where $N_{cut}$ is greater than $N^*$, in order to mitigate risks such as under-utilizing the runway [11]. Figure 2 shows a schematic of the decision process to determine the suggested pushback rate in PRC_v1.

Prior to the start of each 15 min period, the operating runway configuration, meteorological conditions and the predicted number of arrivals in the next 15 minutes are used as inputs in order to predict the expected number of jet takeoffs in the next 15 minutes. The number of departing jets currently active on the surface are then determined, either through visual observations or from surface surveillance. A departure is counted as active once the pushback tug is attached to the aircraft and it is in the process of pushing back. The difference between the current number of
active jet departures and the expected jet takeoffs determines the number of currently active jets that are expected to remain on the ground through the next 15 minutes; the difference between $N_{ctrl}$ and the above number, if positive, yields the additional number of pushbacks to recommend in the next 15 minutes.

**B. PRC_v2**

An alternative approach to PRC formulates an optimal control problem with the objectives of maintaining runway throughput and limiting the size of the departure runway queue [15]. Prior to the start of each 15 min period, the algorithm considers the runway configuration, meteorological conditions, the number of jet aircraft taxiing from the gates to the departure runway, the number of jet aircraft in the departure queue, the expected number of arrivals in the next 15 min and the number of props taxiing out. Optimal control policies can then be derived using dynamic programing. The optimal pushback rate (in aircraft per 15 min) is shown for the BOS runway configuration 22L, 22R during evening times and VMC in Figure 3.

**C. Communication of recommended pushback rates and gate-hold times**

The final rate recommended to the air traffic controllers is rounded to an equivalent rate over a smaller time period (for example, 5 pushbacks in 15 min was translated to 1 pushback per 3 min). The standard format of the gate-hold instruction communicated by the Boston Gate controller to the pilots would include both the current time, the length of the gate-hold, and the time at which the pilot could expect to be cleared. For example, the BOS Gate controller would state: “AAL123, please hold push for 3 min. Time is now 2332, expect clearance at 2335. Remain on my frequency, I will contact you.”

In this manner, pilots would be aware of the expected gate-holds, and could inform the controller of constraints such as gate conflicts due to incoming aircraft. In addition, ground crews could be informed of the expected gate-hold time, so that they could be ready when push clearance was given. The post-analysis of the tapes of controller-pilot
Fig. 3: Optimal pushback policy as a function of the number of jets taxiing to the departure queue and the length of the departure queue (BOS 22L, 27 | 22L, 22R configuration, VMC).

communications from the field-tests show that the controllers clear aircraft for push at the times they initially state (i.e., an aircraft told to expect to push at 2335 would be cleared at 2335), and that they also accurately implement the push rates.

IV. DESIGN OF DECISION SUPPORT TOOL (DST)

A. Color-coded cards

In order to minimize distractions, the research team was not allowed to talk to the Tower controllers during the test periods. In 2010, color-coded cards were used to communicate suggested pushback rates to the air traffic controllers, thereby eliminating the need for verbal communications. One of eight 5” × 7.5” laminated cards, with pushback rate suggestions that ranged from “1 per 3 min” to “1 per min”, in addition to “Stop” (zero rate) and “No restriction” cards was used, as shown in Figure 4 (left). The deployment required two researchers in the Tower: One to collect the necessary inputs for the algorithm, and one to process those values through the algorithm to come up with the appropriate pushback release rate. After the proper pushback rate had been calculated, one of the researchers would place the appropriate color-coded card on the display in front of the Boston Gate controller, for example 3 per 5 min, signifying that only 3 aircraft were recommended to be released every 5 minutes. Any aircraft beyond that would need to be held at the gate. The setup of the suggested rate card in the Boston Gate controllers position is shown in Figure 4 (right).

B. Tablet computers

In 2011, a more advanced DST was designed in order to implement Pushback Rate Control algorithms in the airport tower environment. A particular goal was to develop a process that the controllers would (in principle) be
able to implement without the help of any researchers. This meant creating a program which would generate the desired rate output given the necessary inputs. A mobile device was preferable because the flexibility in how, and by whom, the program was used.

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. The layout of the Boston Logan Airport Traffic Control Tower (ATCT) is shown in Figure 5.
1) 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 6. The expected takeoff rate and the recommended pushback rate were then automatically calculated by an application on the input device, and transmitted to rate control receiver.

Fig. 6: Rate control transmitter, showing the input interface.

The inputs only need to be updated every 15 minutes. They can all be determined by either looking at the flight data screen or by manually counting the number of flight strips on the Ground or Local Controller positions. Because these numbers are easy to determine, it is quite simple to update the inputs to the tablet.

2) Outputs: The receiver conveyed the suggested pushback rate to the Boston Gate controller through one of two display modes: the rate control and the volume control displays.

   a) Rate control display: The output in this mode was a color-coded image of the suggested pushback rate. In this display mode, the Boston Gate controller keeps track of the time intervals and the number of aircraft that have already pushed back. When the demand for pushbacks exceeds the recommended rate, aircraft are held at the gate until the next time interval. The Boston Gate controller also keeps track of aircraft holds, and releases them at the appropriate time.

   b) Volume control display: This display mode was developed to help the Boston Gate controller keep track of the number of aircraft that had called and had been released. It is an alternative to the handwritten notes that controllers otherwise use to keep track of gate-holds. The volume control mode also provides visual cues of the passage of time, and upcoming actions. The volume control display was expected to reduce controller workload, and to possibly help merge the Boston Gate controller position with another position.

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 each, with each row corresponding to a 5-minute time interval (illustrated in Figure 7). A time interval becomes active when the current time is within that time interval, and is indicated by a small black arrow to the left of the time interval. Aircraft can only be released during an active time interval, otherwise pushback positions can only be reserved. Any unused release slots in a given time interval roll over to the next time interval, up to a maximum of twice the rate. The following actions are available in the volume control display mode:

1) **Releasing an aircraft:** If a flight calls for pushback, one of the aircraft icons in the ongoing time interval is selected. The color of the icon changes from black to gray, indicating that it has been released.

2) **Reserving a pushback spot:** If a flight calls for pushback and there are no more positions available in the current time interval, the BG controller tells the aircraft to hold and reserves the next available spot. This is done by selecting the appropriate aircraft icon on the display, which then rotates by 45 degrees to indicate that it has been reserved. When that aircraft is eventually released, the controller selects the aircraft icon again; the icon then rotates back and turns gray.

3) **Reserving a position in a future time period:** A pushback position for a future 15-minute time period can be reserved by touching the blank space next to that time period. A rotated aircraft icon then appears in order to indicate a reservation. When the corresponding time period arrives, that aircraft icon will appear as already reserved on the display.

V. DST DEPLOYMENT AND TESTING

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 Boston Gate 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 Boston Gate position was merged with another position, either clearance delivery or the Traffic Management Coordinator to investigate the potential implementation of PRC without requiring an additional controller at the Boston Gate position.

The recommended rate is valid for the next 15 minutes, and then needs to be updated. A timer was included in the application to remind the controllers that it was time to update the inputs and rerun the control algorithm to receive a new rate. Similarly, the rate control receiver is notified by a popup window that the rate has been updated.

VI. 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. There were 21 respondents in all, 15 of whom filled the Boston Gate position in 2010 [11], 13 who did so in 2011, and 12 who did so in both years. The remaining respondents served in other positions in the BOS tower.

Quantitative ratings were solicited on five topics: Whether they thought fuel burn decreased, whether surface traffic flows improved, whether throughput was adversely impacted, whether the volume control display was easier to use that the rate control display (referred to as the “card display” since color-coded cards were used in 2010 [11]), and whether they found the volume control mode easy to use. The survey results, shown in Figure 8, demonstrate that the responses were generally positive and that the controllers liked the DST. Several of the controllers who agreed that the throughput was adversely impacted also agreed that the surface traffic flow improved, suggesting that there may have been some confusion on the phrasing of this question.

![Fig. 8: Histogram of responses from air traffic controller survey regarding the PRC_v2 DST at BOS.](image)

Thirteen responses were also positive about combining Boston Gate and another position, removing the need
for a dedicated controller during gate-holds. Ten of these responses suggested BOS Gate should be combined with Clearance Delivery, three indicated it should be combined with the Traffic Management Coordinator, and one person each voted for Ground Control and Flight Data (it was possible to select more than one possible position for the combination).

The write-in questions also revealed that the controllers liked the volume control display format. Comments on the best features of the DST included “the ability to touch planes”, “reserve spots”, “[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 include increasing the icon sizes and maintaining more pressure on the runway. In general, the controllers were happy with the modifications made between 2010 and 2011, with one of them remarking, “Liked the improvement in just one year”.

A. 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. During highly convective weather, the abundance of these programs leads to many target departure times, schedule disruptions and flight cancellations. As a result, surface congestion often does not build up, and there is no need for gate-holds. However, there are exceptions to this general behavior, including two days during the PRC_v2 field tests (Jul 18 and Jul 21, 2011). During these days, 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 strategy. The integration of the MINIT restrictions with metering was simple: The total number of flights released per time window was set by the metering program, and the mix by the MINIT program. For example, if the recommended pushback rate were 3/5 min while westbound flights had 5 MINIT, the controller would release two flights with no MINIT restrictions along with a westbound departure, every 5 min. The field-tests also showed that if known in advance, delays due to controlled departure times could be efficiently absorbed as gate-holds.

2) Increased predictability: An additional benefit of the approach was the ability to communicate expected pushback times to pilots in advance. 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. These flights received their release times several minutes in advance, which can be useful in planning ground resources.

3) Natural metering effect: The suggested pushback rate in very low congestion time-periods is 1 per min. However, when the Boston Gate position was merged with another position, it resulted in a natural rate of 1/min without explicit gateholds. For example, when the Boston Gate position was merged with the Traffic Management Coordinator, after the controller cleared an aircraft that called for push, he/she would have to spend the rest of the
minute for a traffic management task (such as, calling the center to obtain an Expected Departure Clearance Time). As a result, the next aircraft would only be released after a minute, resulting in a natural metering of 1 per min unless a lower rate was recommended.

VII. Conclusions

The Pushback Rate Control approach has demonstrated how a simple airport congestion control strategy can be implemented in practice with minimal changes to current procedures and controller workload. The demo at BOS comprised of 15 four-hour periods of metering: Eight periods with the color-coded cards and PRC_v1 in 2010, and seven periods with the tablet computer DST and PRC_v2 in 2011. The benefits of pushback metering were significant, with over 23-25 US tons of fuel burn reduction [11, 15].

A key contribution of this work was the development of a rate control strategy, which was more amenable to implementation by air traffic controllers in the current operating environment. Two modes of decision support were investigated: The first was based on color-coded cards that removed the need for verbal communications with the controller on duty; the second was a more advanced decision support tool built on tablet computers. An application for the tablet computer was also developed to automate the task of determining the recommended pushback rate. A survey of the air traffic controllers showed that decision support tool was well-received. While the researchers or controllers had to manually input certain pieces of data in this prototype, the inputs could easily be obtained from a live surface surveillance data feed in the future, eliminating the need for a manual update every 15 minutes.

Acknowledgments

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 the Air Transport Association. Vivek Panyam (University of Pennsylvania) volunteered his effort in the development of the Android tablet application. Harshad Khadikar (MIT) participated in tower observations before and during the trials, and also analyzed the ASDE-X data. We thank Steve Urllass 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


