

Networked Control of Aircraft Operations at Airports and in Terminal Areas

by

Harshad Khadilkar

Bachelor of Technology, Indian Institute of Technology (Bombay)
Master of Science, Massachusetts Institute of Technology

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

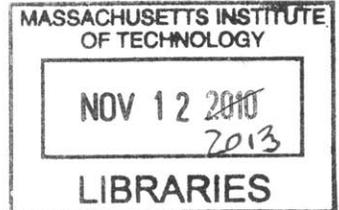
at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2013

© Massachusetts Institute of Technology 2013. All rights reserved.

ARCHIVES



Author
Department of Aeronautics and Astronautics
June 14, 2013

Certified by
Hamsa Balakrishnan
Associate Professor of Aeronautics and Astronautics
Thesis Supervisor

Certified by
Amedeo Odoni
Professor of Aeronautics and Astronautics
Doctoral Committee Member

Certified by
Emilio Frazzoli
Associate Professor of Aeronautics and Astronautics
Doctoral Committee Member

Certified and Accepted by
Eytan Modiano
Professor of Aeronautics and Astronautics
Doctoral Committee Member
Chair, Graduate Program Committee

Networked Control of Aircraft Operations at Airports and in Terminal Areas

by

Harshad Khadilkar

Submitted to the Department of Aeronautics and Astronautics
on June 14, 2013, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

The goal of this thesis is to develop a control strategy for airport operations that integrates the management of arrivals and departures. The strategy is based on four central ideas: (1) the objective of reducing aircraft flight times, taxi times and fuel burn, (2) the emphasis on developing models using data from actual aircraft operations, (3) the need to be compatible with current air traffic control procedures, and (4) the requirement to not adversely affect airport performance. The scope of this work covers the airport surface and arrival airspace, which are two of the most congested regions of the air transportation network.

A new approach is proposed for modeling airport surface operations. Drawing an analogy from the field of network congestion control, the airport surface is assumed to be a network consisting of major taxiways and their intersections. Posing the problem in this framework relaxes the requirement of precisely predicting the taxi time of each aircraft, instead emphasizing the accurate representation of the underlying stochastic processes. At the same time, it allows one to address the issues of network stability and performance through analytical approaches. Based on this model for surface operations, a control algorithm is developed for regulating the time of entry of aircraft into the network. Simulations show that this strategy can significantly reduce surface congestion and aircraft fuel burn without hampering airport performance.

The arrival airspace control algorithm presented in this thesis proposes a hybrid centralized / distributed algorithm for conflict detection and resolution. It combines distributed control in low-density airspace with centralized control in high-density terminal areas. This approach has the advantage of reducing ground infrastructure cost due to decentralization, while still operating at an efficiency level close to that of a fully centralized control strategy. The arrival and departure control algorithms are then combined to formulate an integrated strategy for managing airport operations, significantly improving the separate gains that can be obtained from each component.

Thesis Supervisor: Hamsa Balakrishnan

Title: Associate Professor of Aeronautics and Astronautics

Acknowledgments

Several people have been instrumental in supporting me and driving me to earn this degree. The most important has been Prof Hamsa Balakrishnan, who has guided me throughout my four years at MIT. I would like to thank her for being extraordinarily supportive of my work.

Just as importantly, I also want to thank my parents for inculcating in me the principles that have shaped all aspects of my life, and for their continued support through my journey.

I am grateful to my committee, Profs Odoni, Frazzoli and Modiano, for their help and guidance during the development of my thesis. I would also like to thank Tom Reynolds and Prof John Hansman for their valuable inputs. Thanks also to Ioannis, Varun, Lishuai, Alex, Hanbong, Lanie, Yi-Hsin and Yash for their suggestions and collaborative efforts during my time in ICAT. Similarly, I would also like to thank Pangun Park and Prof Claire Tomlin of UC Berkeley for hosting me during the summer of 2012. Their research on communications algorithms forms a complementary foil to the control algorithms in Chapter 4 of this thesis.

Last but certainly not the least, a huge thank you to my circle of friends and well-wishers: Amod, Abhishek, Ashish, Aditi, Shamel, Somani, Radhika, Sivaraman, Rutu, Himanshu, Atulya, Nihit, Vikram, Siddharth, and many, many more. You were all there for me through the good times and especially the bad ones, and I will always be grateful for your presence.

Contents

1	Introduction	17
1.1	Challenges to improving airport operations	19
1.2	Related literature	21
1.2.1	Modeling of airport operations	22
1.2.2	Airport congestion management techniques	23
1.2.3	Network congestion control	26
1.2.4	Emerging surveillance and navigation technologies	28
1.3	Thesis contributions	30
2	Network Model of Airport Surface Operations	33
2.1	Background	33
2.1.1	Data used for parameter estimation and model testing	34
2.1.2	Model structure	34
2.1.3	Analysis of empirical data	34
2.2	Model development	37
2.2.1	Random processes governing link travel times	37
2.2.2	Variation of parameters with surface traffic	38
2.2.3	Model for total taxi-out time from gate to runway	43
2.2.4	Variation of stopping probability on links with surface traffic level	43
2.3	Model verification	45
2.3.1	Consistency of estimated parameters	46
2.3.2	Comparison of variable parameters with empirical data	46
2.3.3	Comparison of taxi-out times conditioned on surface traffic levels	46

2.3.4	Comparison of full taxi-out time distributions on links	50
2.4	Additional insights	50
2.4.1	Parameter estimation for the taxi-in process	51
2.4.2	Parameter estimation for LaGuardia airport	52
3	Congestion Control of Departure Operations	55
3.1	The k_{ctrl} algorithm: maintaining constant traffic levels	56
3.1.1	Objective and implementation protocol	56
3.1.2	Definition of control strategy	57
3.1.3	Simplified model of departure process from the network	58
3.1.4	Optimal control strategy	59
3.1.5	Simulation of single link networks	60
3.1.6	Simulation of the full airport network	61
3.1.7	Markov chain model with embedded control values	62
3.1.8	Calculation of transition probabilities	65
3.1.9	Shortcomings of the k_{ctrl} strategy	66
3.2	Control based on dynamic programming	68
3.2.1	Markov process representation the taxi time model	68
3.2.2	Derivation of maximum link throughput	71
3.2.3	Derivation of maximum network throughput	73
3.2.4	Cost definition	76
3.2.5	Control strategy for single links	77
3.2.6	Control strategy for the full airport model	82
3.2.7	Comparison of k_{ctrl} and dynamic programming strategies	86
3.2.8	Adaptive algorithm for unmodeled parameter variation	86
4	Control of Arrival Operations	91
4.1	Characterization of arrival airspace	92
4.1.1	Automatic Dependent Surveillance - Broadcast	92
4.1.2	Proposed system architecture	93
4.2	Development of control strategy for arrivals	94

4.2.1	Geometry of engagement between a pair of aircraft	95
4.2.2	Optimal velocities for a pair of aircraft	98
4.2.3	Optimal velocities for multiple merging aircraft	100
4.2.4	Distributed control strategy	102
4.2.5	Centralized control strategy	103
4.2.6	Constraint on communication performance	104
4.2.7	Challenges to control implementation	107
4.3	Simulations of control strategy	108
5	Integrated Control of Arrivals and Departures	117
5.1	Combining the arrival and departure algorithms	117
5.2	Performance evaluation	120
5.2.1	Simulation procedure	120
5.2.2	Simulation results	121
5.2.3	Effect of control strategy on taxi-out times	122
5.2.4	Effect of control strategy on takeoff delay	126
5.3	From modeling to implementation: A summary	126
6	Conclusions	133
6.1	Summary	133
6.2	Future research opportunities	134
	Appendix A Parameter Values for Departures at BOS	137
	Appendix B Parameter Values for Arrivals at BOS	143

List of Figures

1-1	Departure queues at LaGuardia and Boston Logan airports.	18
1-2	Throughput saturation at Boston Logan for the 22L, 27 22R configuration.	20
2-1	Network model of Boston Logan International Airport.	35
2-2	Abstraction of the network model.	35
2-3	Flowchart for measuring empirical distribution of link travel times.	36
2-4	Distribution of taxi speed over the links 1→2 and 2→3.	36
2-5	Empirical distribution of number of stops and duration of stops, for four links.	39
2-6	Unimpeded taxi times on various links.	39
2-7	Number of stops on various links.	40
2-8	Stop durations on various links.	40
2-9	Information projection of the empirical distribution P on three candidate model families.	41
2-10	Variation of travel times over Links 1→2 and 4→5 with total traffic on the surface.	42
2-11	Variation of travel times over Links 1→2 and 2→3 with traffic on the same link.	44
2-12	Means and standard deviations of the parameters for several links at Boston Logan.	47
2-13	Variation of stopping probability on four links with the level of surface traffic.	48
2-14	Comparison of travel time distributions conditioned on number of stops.	49
2-15	Comparison of full travel time distributions over different links in the network.	50
2-16	Arrival stopping probabilities.	52
2-17	Stop time distribution for arrivals.	52
2-18	Unimpeded time distribution for arrivals.	53
2-19	Network model for LaGuardia Airport	54

2-20	Comparison of model for LaGuardia airport with independent test data.	54
3-1	Variation of surface traffic levels seen in simulations of control algorithm for single link, with $k_{ctrl} = 10$	60
3-2	Variation of average taxi-out times seen in simulations of control algorithm for single link, with $k_{ctrl} = 10$	61
3-3	Simulated traffic levels and throughput for the Boston Logan network, with departures from Runway 27.	63
3-4	Simulated gate delays and taxi times for the Boston Logan network, with departures from Runway 27.	64
3-5	Markov chain model for transitions under the proposed control strategy, if $k < k_{ctrl}$	65
3-6	Markov chain model for transitions under the proposed control strategy, if $k \geq k_{ctrl}$	65
3-7	Comparison of Markov-approximated steady state occupancy and average simulated occupancy, for $k_{ctrl} = 10$	67
3-8	Representation of the taxi-out model as a Markov process, for a single link.	69
3-9	Expected length of each segment in the single link model.	71
3-10	Comparison of theoretical transition probabilities and simulation-generated transition probabilities.	71
3-11	Representation of the taxi-out model as a Markov process, for a path with two links.	72
3-12	Network with two sources, three links.	74
3-13	Network with one source, multiple links.	75
3-14	Model for a single link with finite gate capacity.	80
3-15	Comparison of optimal policies for a single link, $\beta = 0.02$	80
3-16	Comparison of optimal policies for a single link, $\beta = 0.06$	81
3-17	Simulation of the control strategy with and without consideration for the available buffer capacity.	81
3-18	Network layout for departures from Runway 27.	83
3-19	Illustration of the departure control algorithm for calculating optimal gate delays.	83
3-20	Optimal pushback delays at Boston Logan, calculated for departures from source 1.	85
3-21	Comparison between the k_{ctrl} and dynamic programming based control strategies.	87
3-22	Simulation of the k_{ctrl} and dynamic programming based control strategies.	87

4-1	Overview of the model for arrival airspace.	94
4-2	Geometry for calculating the distance of closest approach (Plan view).	96
4-3	Possible geometries for multiple merging aircraft.	101
4-4	Information asymmetry in the distributed zone.	103
4-5	Setup for derivation of maximum allowable SUI.	105
4-6	Model of the airspace around Los Angeles International Airport, with procedures for arrivals on Runway 25L.	112
4-7	The onset of holding patterns with different radii of centralized control.	113
4-8	Number of holding patterns generated by the control strategy for different levels of incoming arrival demand.	113
4-9	Traffic levels in the simulated airspace for different radii of centralized control.	114
4-10	Average time required for aircraft on the periphery of the modeled airspace to land at the airport.	115
5-1	Model of the airspace around Boston Logan International Airport with procedures for arrivals on Runway 33L.	119
5-2	Arrival throughput characteristics at Boston Logan.	119
5-3	Schematic of the combined control algorithm and simulation procedure.	121
5-4	Operation counts, average taxi and pushback delay times, and surface counts in a full day's simulation.	123
5-5	Gate occupancy over the course of the simulated day, at each of the four sources (airport terminals).	124
5-6	Comparison of the distribution of taxi-out times for three different control strategies.	125
5-7	Comparison of the distribution of takeoff times relative to those attained using current procedures.	127
5-8	Schematic of proposed combined control strategy for arrivals and departures.	128

Nomenclature

a	Weight on pushback delay for k_{ctrl} algorithm
ADS-B	Automatic Dependent Surveillance - Broadcast
ASDE-X	Airport Surface Detection Equipment, Model-X
ASPM	Aviation System Performance Metrics
ATC	Air Traffic Control
α	Discount factor in dynamic programming formulation
BOS	Boston Logan International Airport
β	Arrival rate of aircraft to the airport
β'	Rate of incoming arrival demand at periphery of modeled airspace
β_i	Arrival rate of aircraft to buffer i
C	Cost function for dynamic programming formulation
\hat{C}	Online estimate of stage cost for rollout algorithm
C_k	Cost function for k_{ctrl} algorithm
c_1	Coefficient of expected taxi time in stage cost
c_2	Coefficient of pushback delay in stage cost
c_3	Coefficient of loss of throughput in stage cost
χ	Set of parameters
$\chi_{\mathbb{M}}^*$	Optimal parameters for model family \mathbb{M}
D	Destination node
D_{KL}	Kullback-Leibler divergence
$d_{i,0}$	Distance of aircraft i from merge point at time $t = 0$
$d_{i,t}$	Distance of aircraft i from merge point at time t
δ	Angle made by two merging arrival paths
δ_r	Angle between relative position and relative velocity vectors
E	Erlang family of probability distributions
η_l	Portion of expected taxi time on link l that is independent of surface traffic level k
η_{0l}	Nominal value of η_l
$\hat{\eta}_l$	Online estimate of η_l given taxi times of first r aircraft
\mathbb{E}	Expectation operator
FAA	Federal Aviation Administration
G	Gaussian family of probability distributions
Γ_i	Maximum allowable state update interval for aircraft i
γ	Buffer overflow tolerance
\hat{i}	Unit vector along x axis
$J(\theta)$	Optimal infinite horizon cost-to-go from state θ
\hat{j}	Unit vector along y axis
k	Surface departure traffic level
k_{ctrl}	Target departure surface traffic level
k_{max}	Maximum modeled traffic level
$k_p(u)$	Projected traffic level after time u

L	Log-normal family of probability distributions
\mathcal{L}	Lagrangian function
LAX	Los Angeles International Airport
l	Variable denoting a link in the airport surface network
λ_l	Rate of unimpeded Erlang on link l
\mathbb{M}	Variable denoting family of probability models
μ_l	Rate of exponential distribution for duration of stop on link l
NAS	National Airspace System
$N_{s,l}$	Random variable denoting number of stops on link l
\bar{N}_i	Number of empty gates in buffer i
N_i	Number of occupied gates in buffer i
$N_{i,\max}$	Gate capacity of buffer i
n_l	Order of unimpeded Erlang on link l
ν_s	Rate of entry of aircraft into the network from source s
P	Empirical distribution
\mathcal{P}	Taxi path
PDARS	Performance Data Analysis and Reporting System
p_{kl}	Geometric probability of stopping on link l when traffic level is k
$p_{\theta_1, \theta_2}(u)$	Probability of transition from state θ_1 to state θ_2 after time u
ϕ	Full state of the airport in the dynamic programming formulation
ψ_i	Lagrange multipliers
Q	Theoretical distribution
$q_{\mathbb{M}}^*$	Information projection of empirical distribution on model family \mathbb{M}
$R_{k,\mathbb{C}}$	Approximate aircraft departure rate from network in k_{ctrl} algorithm, for runway configuration \mathbb{C}
$R'_{k,\mathbb{C}}$	Approximate aircraft departure rate from network in dynamic programming algorithm, for runway configuration \mathbb{C}
\bar{r}	Relative displacement vector from one aircraft to another at time t
\bar{r}_0	Relative displacement vector from one aircraft to another at time $t = 0$
r_c	Distance of closest approach between two aircraft
S	Stopped state
SSR	Secondary Surveillance Radar
SUI	State Update Interval
s_{\min}	Minimum separation requirement between two aircraft in flight
s_{next}	Source supplying the next aircraft for pushback
σ_l	Maximum departure rate from link l
TRACON	Terminal Radar Approach and Control
t	Time
t_{av}	Expected taxi time left for an aircraft in the network, when observation is started at a random time
t_c	Time of closest approach between two aircraft
t_l	Random variable denoting total travel time on link l
$t_{r,l}$	Realized taxi time of r^{th} aircraft on link l
$t_{s,l,i}$	Random variable denoting duration of i^{th} stop on link l

$t_{u,l}$	Random variable denoting unimpeded travel time on link l
τ_i	Slack variables in the optimization of Lagrangian \mathcal{L}
θ	Aggregate state of the airport in the dynamic programming formulation
θ_s	State of a single link in the dynamic programming formulation
\mathbb{U}	Set of allowable controls
u	Assigned pushback delay
u^*	Optimal pushback delay for the k_{ctrl} algorithm
u_{max}	Maximum allowable pushback delay
\bar{v}_i	Velocity vector of aircraft i
v'_i	Off-nominal velocity of aircraft i
$v_{i,0}$	Initial velocity of aircraft i
$v_{i,f}$	Final velocity of aircraft i
$v_{i,\text{max}}$	Maximum velocity of aircraft i
$v_{i,\text{min}}$	Minimum velocity of aircraft i
\bar{v}_r	Relative velocity vector between two aircraft
W	Gain of adaptive algorithm for online estimation of $\hat{\eta}_l$
X_l	Rate at which expected taxi time on link l increases on addition of aircraft to the airport surface, as a fraction of the expected time of each stop on the link
x_0	Relative displacement between two aircraft along the x axis
y_0	Relative displacement between two aircraft along the y axis
$\mathbb{Z}_{\geq 0}$	Set of non-negative integers

Chapter 1

Introduction

“A commercial aircraft is a vehicle capable of supporting itself aerodynamically as well as economically”

- William Stout,
Designer of the Ford Trimotor

The field of air transportation has been subjected to intense scrutiny over the last three decades, due to a remarkable increase in demand. A steady growth in the volume of operations, coupled with the slow growth in air transportation infrastructure, has led to a large increase in flight delays. The economic cost of airspace and airport congestion is borne by the passengers as well as the airlines. High and fluctuating fuel prices are an especially difficult challenge to airline profitability. A new dimension to this challenge has been added by the emerging awareness about climate change and the environmental cost of aviation. Government policies driven by public pressure and voluntary initiatives have made many airlines and regulatory authorities declare their intention to move towards carbon-neutrality. An obvious step towards achieving this objective is a reduction in the amount of fuel consumed during aircraft operations. According to a commitment originally signed in 2008 and reaffirmed in 2012, airlines are aiming to reduce fuel consumption by 1.5% per year up to the year 2020 [1]. Current studies indicate that improving the efficiency of the air transportation system could reduce fuel consumption by 9 million tons per year, and CO₂ emissions by 28 million tons per year [2]. It should be noted that these savings can be targeted in the near future, as opposed to the longer term procedure of replacing old aircraft with new, more fuel efficient ones.

Flight delays in the National Airspace System (NAS) can be caused by several factors. Some

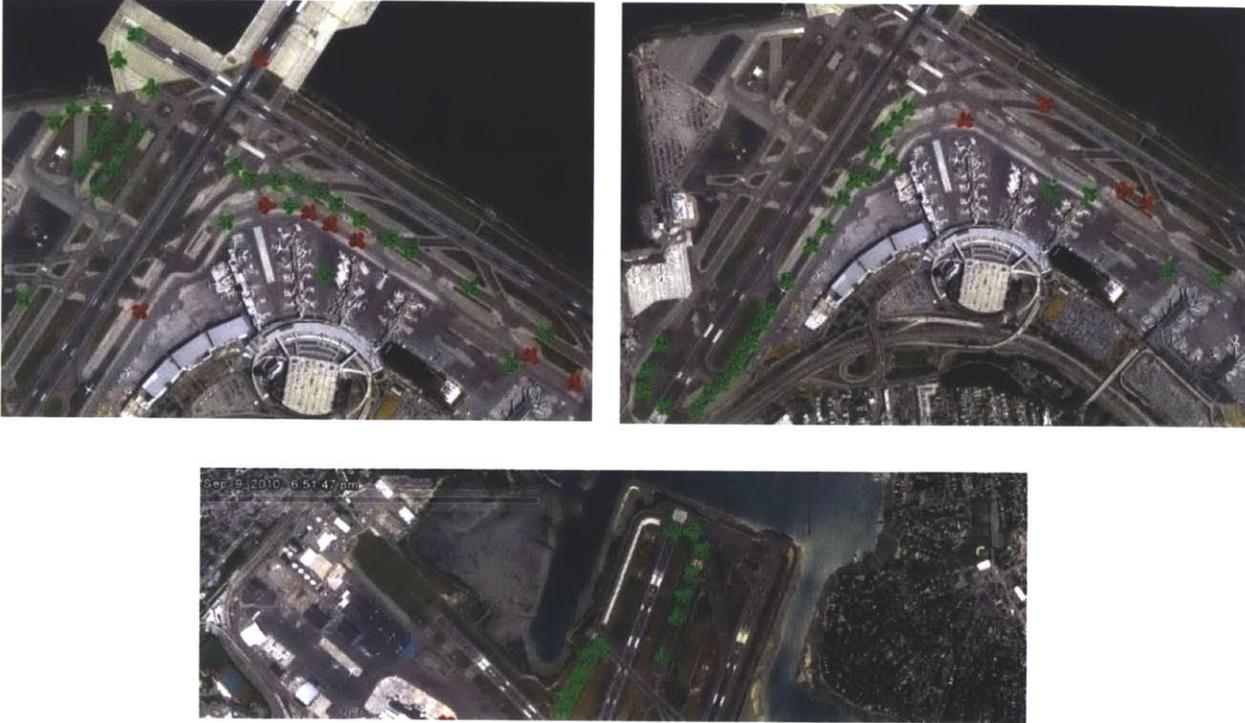


Figure 1-1: [Clockwise from top left] Departure queues at (1) Runway 13 at LaGuardia Airport on Aug 09, 2011, (2) Runway 04 at LaGuardia Airport on Aug 11, 2011, (3) Runway 22R at Boston Logan International Airport on September 09, 2010. Visualization of actual surface surveillance data using Google Earth¹.

kinds of delay are unavoidable, such as those due to severe weather conditions or maintenance issues. However, approximately 12.5% of flight delays in the United States in 2012 were caused by traffic volume in the vicinity of the airport [3]. The Federal Aviation Administration's Aviation System Performance Metrics (ASPM) database estimates that annually, taxi-out delays (the difference between actual and unimpeded taxi-out times) at major airports in the United States exceed 32 million minutes [4].

The delay problem is most acute in the region composed of the airport surface and the surrounding airspace, since it is the most constrained part of the air transportation network [5]. Inefficiencies in this area lead to excess fuel burn, which can be reduced in two ways. Firstly, the total amount of delay can be reduced by improving the efficiency of flight operations. Secondly, the distribution of delay can be shifted such that the phase of flight that absorbs the most delay is that which burns the least fuel. For example, delay absorbed on the ground requires less fuel than delay absorbed

¹ Visualization of LaGuardia data courtesy of Melanie Sandberg.

while airborne. Shifting the ground delay itself from active movement areas (where aircraft engines are on) to inactive holding areas (where aircraft engines are off) can further decrease the amount of fuel consumed.

Figure 1-1 shows three instances of long departure queues at LaGuardia Airport and Boston Logan International Airport (BOS). The images depict operational data obtained from a surface surveillance system, with green aircraft denoting departures and red aircraft denoting arrivals. Queue formation is commonly seen at busy airports, during periods of high demand. Each aircraft in the queue has its engines running and is consuming fuel at a significant rate. Even aircraft that employ single-engine taxi procedures typically start all engines before they are in the departure queue. The total fuel burn would be reduced if the surface traffic was spread out such that a reasonable number of aircraft were in queue, others were taxiing towards the runway, and the rest were waiting at their gates. The justification for this claim is based on studies of airport performance as a function of the surface traffic levels.

Consider, for example, the relationship between the number of takeoffs per 15 min and the number of active departing aircraft as shown in Figure 1-2. The x axis denotes the number of active aircraft, i.e., those that have left their gates but have not yet taken off. It can be seen from the figure that as the number of active aircraft increases, the takeoff rate increases. However, the marginal increase in takeoff rate exhibits saturation. Limiting the number of active aircraft to a reasonable number will thus reduce fuel consumption without adversely affecting airport performance. While this idea is fairly obvious in theory, there are several subtleties that can jeopardize their implementation in practical situations. The notion of a ‘reasonable number’ is somewhat vague, and varies from airport to airport. In addition, it is necessary to operate within established procedures and to understand the constraints imposed by existing infrastructure at airports. It is also desirable to characterize the relationship between airborne terminal-area procedures and aircraft ground movement, in order to maximize the potential reduction in fuel burn and emissions.

1.1 Challenges to improving airport operations

It has been estimated that 25% of all aviation emissions are produced in the Landing and Take-Off (LTO) cycle of a flight, with taxi operations being the largest contributor in this phase [6]. Fuel for taxi operations costs approximately 7 billion USD per year, and results in 18 million tons of

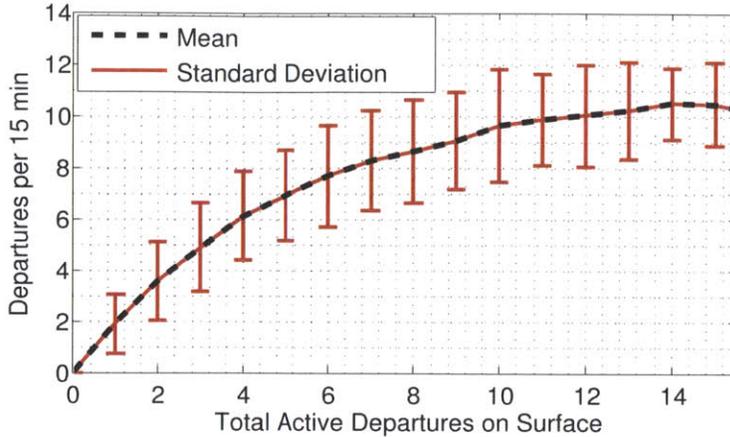


Figure 1-2: Relationship between the average number of takeoffs in 15 min and the surface traffic level at Boston Logan International Airport, when departures are taking place from Runway 22R and arrivals are taking place on Runway 27 and Runway 22L.

CO₂ per year [6]. Analysis of operations in Europe has found that aircraft spend 10-30% of their flight time taxiing, and that a short/medium range A320 expends as much as 5-10% of its fuel on the ground [7]. Additionally, it has also been shown that significant surface congestion at some US airports leads to taxi times as high as three times those when there is no congestion [8]. There are two major reasons for the delays experienced at airports and in terminal airspace. The first cause is the large number of aircraft occupying a limited region of space, both in the air and on the ground. This problem is magnified by the existence of a second limitation: the largely legacy infrastructure used for air traffic control. The basics of air traffic control have remained roughly the same since its advent in the 1930s. The entire system is based on the use of radars for surveillance, paper flight strips for keeping track of aircraft, and radios for communication. The aircraft being managed by this system carry 21st century technology, but are still directed using standards defined in the 1960s. When new technologies such as those described in Section 1.2.4 are introduced, algorithms that take full advantage of their potential need to be developed. At the same time, such algorithms should also be compatible with legacy systems during the deployment period.

Any effort to reduce fuel burn and emissions in the vicinity of airports needs to overcome a number of challenges. Safety is the overriding concern in air traffic management. Operational procedures are in place for each phase of a flight. These rules are sometimes deliberately conservative,

in order to account for human error, surveillance uncertainties, communication lags as well as rare events. It is not practical to propose radical changes to procedures and technology in order to improve operational efficiency. While implementing unrestricted flight paths, or free flight, may be desirable from the point of view of efficiency [9], its practical realization is still several years away. In the meanwhile, it is important to introduce operational measures that improve efficiency by merging new technologies with the existing airspace structure. Similar improvements can be obtained on the airport surface by using new surveillance technology to reduce fuel burn, while remaining within the current procedural framework. Strategies that aim to achieve synergy between new technology and existing procedures are the central focus of this thesis.

1.2 Related literature

Improving the efficiency of the air traffic management system has been the subject of research for a considerable length of time. Traditional approaches to this problem focus on traffic flow management (TFM) [10], airport slot controls [11, 12], airport/sector capacity estimation and prediction [13, 14, 15, 16], delay estimation and prediction [17], and mitigation of weather impacts [18]. These studies are strategic in nature, and are most relevant to the high-level design of the air transportation system. For an impact on near-term efficiency, tactical modifications to operating procedures are needed. The development of such methods is based on two important components: the modeling exercise and the proposed implementation protocol.

There have been several studies in the past few years that have focused on airport operational efficiency. Some propose efficiency improvements in the airspace surrounding an airport [19, 20, 21], while others are specific to surface operations [22, 23, 24, 25]. The mathematical techniques used to solve the formulations also vary, and draw from the fields of queuing theory, mixed-integer linear programming and regression analysis. However, there does not appear to be any prior study that provides an integrated approach to air traffic management near airports. The aim of this thesis is to present such an integrated control strategy, one that has been developed from the bottom up through the process of data analysis, operations modeling, mathematical formulation, simulation and a proposal for practical implementation.

1.2.1 Modeling of airport operations

Models of airport operations typically focus on one of two aspects: the airborne phase, or airport surface movement. The most unrestricted form of airborne air traffic control is the free flight concept, first formalized in 1995 [26]. This concept allows each aircraft to calculate its own optimal trajectory based on information about surrounding traffic conditions. Several definitions of optimality have been proposed in literature, including maximizing safety [27] and minimizing flight times [19, 28, 29, 30]. However, the airspace and traffic models differ from one study to another, and are rarely implementable in a practical scenario with current communication, computation and operational constraints.

Prior literature on surface operations shows that taxi speeds, taxi routes and pilot behavior are highly variable in this context. The total number of aircraft on the airport surface at any one time is also relatively small. Due to these reasons, Eulerian (fluid) models which may be used for airspace modeling [31, 32, 33] are not suitable for representing aircraft ground movement. Most prior surface modeling approaches focus on *microscopic models* [34, 35, 36] or *macroscopic models* [22, 23, 24, 25]. Microscopic approaches model the precise location and velocity of each aircraft at all times, and control is implemented by specifying a time-based trajectory for each flight. The primary drawback of such methods is that precise conformance to specified trajectories is not possible given the current state of technology at airports. Macroscopic approaches to the problem model aircraft taxi-out times as the outputs of an aggregate service process, frequently using queuing theory. These approaches to the surface traffic management problem either make simplifying assumptions to obtain analytical results, or else require numerical solutions. In addition, a service process needs to be defined at every queuing node; it is typically assumed that queuing occurs only at the runway and not on the taxiways [22].

Chapter 2 proposes a new modeling approach which falls broadly in the category of *mesoscopic models* [37, 38]. The level of fidelity is high enough to highlight an airport's operational characteristics [39, 40], but there is also a recognition of the stochastic nature of the processes involved. The airport surface is assumed to be a network consisting of major taxiways (edges or links) and their intersections (nodes). The problem is then analogous to the network congestion control problem encountered in several other fields of research [41].

1.2.2 Airport congestion management techniques

A review of proposed congestion mitigation methods for airports is presented in this section, categorized by the specifics of the problem addressed. Many of these methods utilize models of airport operations similar to those described in Section 1.2.1.

Arrival airspace congestion

The emphasis in prior literature on airborne control algorithms has been on computing optimal trajectories for free flight, 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 [19, 20]. Therefore, while maintaining the current airspace structure imposes somewhat stricter constraints on airspace capacity, it is shown in [42, 43] that significant efficiency gains can still be obtained with this approach without paying the accompanying computational penalty.

The development of the free flight concept has also encouraged a great deal of research in conflict detection and resolution algorithms. These algorithms can be broadly classified into two types, namely, centralized and distributed approaches. Distributed algorithms [29, 44, 45, 46] are typically based on negotiations between aircraft to find optimal conflict resolution maneuvers, or can be developed based on non-cooperative game theory. Free-flight approaches typically fall into this category of algorithms. By contrast, centralized algorithms [19, 21, 28, 30] assume that information is consolidated at a single central facility, which then solves for the optimal trajectories for all aircraft. Information gathering is usually via ground radar systems, although recent papers have considered the availability of ADS-B. An extensive overview of strategies for conflict resolution can be found in [47].

At very short time-scales, conflict resolution is currently performed by on-board collision avoidance systems such as the Traffic-alert and Collision Avoidance System (TCAS) [48, 49], or the proposed Airborne Collision Avoidance System [50, 51]. These systems are tailored for short-range collision avoidance, and not for long-range efficiency. While conflict resolution over a longer time-scale is known to yield more efficient trajectories [52], current collision avoidance systems cannot

easily be extended due to their limited range and poor performance in high-density airspaces [49].

An extensive body of literature is devoted to the scheduling of arrival operations. Examples of this avenue of research include extensions to the FAA's Traffic Management Advisor [53], combined scheduling of arrivals and departures [54], and the Center-TRACON Automation System (CTAS) [55, 56] and its extensions [57]. These studies typically focus on optimal and conflict-free scheduling of aircraft operations, but do not specifically keep track of real-time collision risk [53, 54, 57]. However, they do note that small changes in speed are a desirable method of implementing conflict resolution maneuvers in high-density operations, as opposed to heading or altitude changes [57].

The arrival airspace control algorithm presented in Chapter 4 proposes a hybrid centralized / distributed algorithm for conflict detection and resolution. The proposed method for implementing conflict resolution maneuvers is a change in flight speed, with holding patterns being commanded when this is not possible. The algorithm combines distributed control in low-density airspace with centralized control in the high-density terminal areas. This hybrid approach offers the dual advantage of reduced ground infrastructure cost due to decentralization and the efficiency of centralization.

Surface congestion

Current air traffic control (ATC) policy at most airports in the United States is to 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 [58, 59]. Better management of surface operations not only offers a way of reducing this impact, but is integral to the future evolution of ATC architecture. As such, it is one of the major objectives listed in the Federal Aviation Administration's plan for improvement of operations in the NAS [60].

There have been several efforts in the United States and Europe to develop and implement surface congestion management strategies, especially in the context of Airport Collaborative Decision Making (A-CDM). Examples include the field-testing of the Pushback Rate Control strategy at BOS [61, 62], the Tower Flight Data Manager (TFDM) demonstration at Dallas Fort Worth (DFW) airport [63], the field evaluation of the Collaborative Departure Queue Management concept at Memphis (MEM) airport [64], the Ground Metering Program at New York JFK airport [65, 66],

the human-in-the-loop simulations of the Spot and Runway Departure Advisor (SARDA) concept at DFW [67], the trials of the Departure Manager (DMAN) concept [68] in Athens International airport (ATH) [69], and the Airport Collaborative Decision Making (A-CDM) implementations at London Heathrow (LHR), Frankfurt (FRA), Amsterdam (AMS), Helsinki (HEL) and Paris Charles De Gaulle (CDG) Airports [70]. There has also been increased interest from major airports in Asia (Bengaluru International Airport (BLR) in India, as well as China and Singapore [71]) on A-CDM. The above surface management approaches can be broadly classified as aggregate approaches that are implemented by the airport tower [61, 62, 63], airline-specific allocation approaches [64], and aircraft-specific metering approaches [65, 66, 67, 68, 69, 70].

Largely driven by the availability of data, field demonstrations of airport congestion control algorithms have traditionally focused on airline operations. Estimates of airport capacity are formed using a combination of the Aviation System Performance Metrics (ASPM) [4] and the Airline Service Quality Performance (ASQP) databases. These databases provide the times at which flights pushback from their gates, their takeoff and landing times, and the gate-in times, as reported by the airlines. ASPM also provides airport-level aggregate data, including records of runway configuration use and counts of the total number of arrivals and departures in 15 min increments. However, the level of detail in these approaches is insufficient to investigate other factors that affect surface operations, such as interactions between taxiing aircraft, runway occupancy times and departure queue characteristics.

A much-studied avenue of research in aggregate approaches to congestion control is the N-Control algorithm and its variants. In its most basic form, it is an event-driven control strategy that aims to maintain a constant level of traffic on the airport surface [37, 72]. The ideal implementation procedure is to clear an aircraft for pushback at the instant of every takeoff. In this way, the number of active aircraft remains steady. However, this method has two potential drawbacks. Firstly, it greatly increases air traffic controller workload. They have to track both takeoff times and pushback calls, tasks that are typically performed by separate controllers. Secondly, it is not possible for ground crews at airports to be available for pushing an airplane without any notice. A more practical version of the N-Control algorithm is Pushback Rate Control [61, 73]. This strategy sets an upper limit on the number of pushbacks within a given time window, usually 15 minutes long. While this strategy has been demonstrated to result in fuel savings, it is difficult to estimate

its efficiency in this regard. A modified version of the N-Control algorithm is studied in Section 3.1, where its advantages and limitations are discussed in detail.

A much finer level of control can be achieved by algorithms based on microscopic models. In these studies, optimal surface trajectories are usually calculated using integer or mixed-integer programming. It has been shown that the potential taxi time and fuel savings are much greater than those possible using aggregate approaches [36]. However, these methods are very difficult to implement in practice. Ground taxi is controlled manually by the pilots, and furthermore, the voice communication channel between pilots and air traffic control is highly congested, making it difficult to add the command and confirmation messages required for detailed control. Data communication channels are still used only for basic functions, and voice confirmation is considered mandatory for safety reasons.

The procedure presented in this thesis aims to allow for the feasibility of implementation, while still achieving a significant fraction of the potential benefits indicated by microscopic models. Posing the problem in the framework described in Chapter 3 relaxes the requirement of precisely predicting the taxi time of each aircraft [74], instead emphasizing the accurate representation of the underlying stochastic processes. At the same time, it allows one to address the issues of network stability and performance through analytical approaches, and does not require the specification of an explicit service process. Prior literature in related fields that employ similar techniques is described in Section 1.2.3.

1.2.3 Network congestion control

The properties of networks have been studied in great detail, with some of the most important basic results appearing in the 1950s [75, 76]. These studies addressed the problem of maximum throughput in general network topologies, and are particularly useful for the purposes of Chapter 3. While solving for the maximum throughput in generic graphs is difficult, the airport surface network has properties that reduce this complexity considerably. Detailed analysis pertaining to maximum throughput and network stability is available in literature [77, 78].

Urban Transportation Networks

Research into the properties and control of urban networks also goes as far back as 1956 [79], with discussions about their capacity and demand management using tolls. A review of recent research in the urban transportation network framework can be found in [80]. The focus in this field is mainly on Geographic Information Systems (GIS) such as in-car satellite navigation. The objective of these studies is to characterize the behavior of individual agents (vehicles) when they have different levels of knowledge about the state of the network at any given time. There are a large number of entry and exit points in a network consisting of roads. Therefore, these networks involve a very large number of sources and sinks, and are generally solved using heuristics [81, 82] such as approximate shortest path algorithms [83]. These algorithms are a useful reference for the problem described in this thesis, but are not directly applicable. The focus of the airport congestion control problem is on management of a handful of sources and sinks, with much more centralized information. The control policies also need to account for the performance of the entire system, as opposed to only individual users. The modeling of stochastic link travel times in an urban transportation context has been studied earlier [84, 85]. Analytical results are obtained in these studies for certain classes of travel time distributions.

Communication Networks

Communication networks also exhibit characteristics that are relevant to the problem at hand. Specifically, the objective in communications is to transmit packets of information from source to destination while avoiding the onset of congestion. This objective is compatible with the formulation of the airport congestion control problem. The relevant properties of communication networks and corresponding control policies are described in [86, 87, 88, 89]. In these studies, stable control laws for the Transmission Control Protocol (TCP) are derived. The focus is on utilizing network capacity while maintaining stability. The available information is assumed to be purely local in nature, which is not a restriction that applies to the airport problem. An algorithm that uses more detailed feedback is described in [90]. It also aims to maximize network utilization while minimizing persistent queues.

Back-pressure algorithms first proposed in [91] and further developed in [92, 93] are a good starting point when considering control algorithms for the airport network. These algorithms rely

on the difference between queue sizes at successive nodes, and allow entry into the network on this basis. In the present context, this ‘back-pressure’ is analogous to the number of active aircraft on the airport surface. A detailed discussion of internet network control algorithms and stability is available in [94, 95], and analysis formulations for wireless systems are developed in [96, 97, 98]. These studies also explore the fairness characteristics of different algorithms, which is of relevance to the airport congestion control problem with its multitude of users and stakeholders.

Control of Manufacturing Processes

The scheduling and control of manufacturing processes deals with the production of a set of parts, and the order in which they are processed by different machines. The real-time scheduling of manufacturing processes is considered in [99, 100]. These studies use a decentralized approach to control, while ensuring that scheduling decisions keep the system stable. It is shown that *clear-a-fraction* policies, or policies where parts with the largest queues are processed first, stabilize all acyclic systems. System performance measures that are relevant to this problem are considered in [101]. A version of the control-point policy, which is very similar to the N-Control algorithm and its variations [37, 61, 72], can be found in [102]. In addition to controlling entry into the network, this work describes an algorithm that can also meter part production rates at intermediate points. This is intuitively similar to the work of a ground controller at airports. The control-point policy described in this work does not allow parts to pass beyond a control point if the excess of production at that point over demand is too great. Since the only feasible control points at airports are the gates, delaying aircraft pushbacks is analogous to metering part production. Finally, a detailed survey of queuing approaches to manufacturing systems and the accompanying assumptions is available in [103].

In summary, each of these separate fields provides useful insights into the airport congestion management problem, as well as starting points for the development of a control strategy.

1.2.4 Emerging surveillance and navigation technologies

The algorithms and policies proposed in this thesis leverage upon two new technologies with air traffic management applications. Airport Surface Detection Equipment, Model - X (ASDE-X) is a surveillance system for the airport surface. Data from this system is used to build the aircraft taxi

models presented in Chapter 2, for parameter estimation as well as for verification. The control algorithm developed in Chapter 3 requires the number of active aircraft on the surface as an input. This number can be calculated from ASDE-X data as an alternative to physical counting. The second relevant technology referred to extensively is Automatic Dependent Surveillance - Broadcast (ADS-B), which is a combination of satellite-based navigation and wireless communication. All aircraft in controlled airspace in the United States will have to be equipped with at least a basic version of this system by 2020, according to an FAA mandate [60]. The broadcast functionality included in this system can be used to perform conflict detection and resolution in congested airspace, as described in Chapter 4.

Airport Surface Detection Equipment, Model-X

Airport Surface Detection Equipment, Model-X (ASDE-X) is primarily a safety tool designed to mitigate the risk of runway collisions [104]. It incorporates real-time tracking of aircraft on the surface to detect potential conflicts and monitor conformance. There is potential, however, to use the data generated by it for surface operations analysis [39, 40, 105] and modeling of aircraft behavior. Reported parameters in ASDE-X include each aircraft's position, velocity, altitude and heading. The update rate is 1 Hz for each individual flight track. The raw surface tracks are processed for the purposes of this work using a multimodal unscented Kalman filter developed in prior work [106]. ASDE-X data from Boston Logan International Airport, spanning a period of three years from 2010 to 2012 is used in this thesis.

Automatic Dependent Surveillance - Broadcast

Automatic Dependent Surveillance - Broadcast (ADS-B) is a Next Generation Air Transportation System (NextGen) surveillance and communication technology, in which aircraft broadcast on-board flight information via a datalink to ground stations or other similarly equipped aircraft [107]. The position and velocity data is obtained using satellite navigation. An important consideration for the deployment of ADS-B is its interaction with legacy infrastructure. Since it uses the same bandwidth as the replies to Secondary Surveillance Radars (SSRs) [60], high aircraft and SSR density near airports could result in interference and degraded surveillance performance. Therefore, the control algorithm presented in this thesis is mated to a communication algorithm that tunes

the transmission power of ADS-B for minimizing interference with radars [42].

1.3 Thesis contributions

The goal of this thesis is to develop a control strategy for airport operations that integrates the management of arrivals and departures. The strategy is based on four central ideas: (1) The objective of reducing aircraft flight times, taxi times and fuel burn; (2) The emphasis on developing models using data from actual aircraft operations; (3) The need to be compatible with current air traffic control procedures; and (4) The requirement to not adversely affect airport performance. The major contributions of this research effort are:

Detailed analysis of airport operations

The algorithms presented in this thesis are based on analysis of operational data, including surface surveillance (ASDE-X) and flight data recorders (FDR). The dynamics of congestion on the airport surface are modeled and the parameters are tuned using empirical ASDE-X data. The FDR data is used to estimate the fuel burnt by aircraft on the ground [108]. This estimate is an essential component of the optimization framework described in Chapter 3.

Network model of airport surface operations

Analysis of empirical data reveals that aircraft movement characteristics vary greatly, depending on their location on the airport surface. For example, aircraft near the terminals at the airport taxi slower and are more likely to stop because of conflicts with other taxiing aircraft. On the other hand, aircraft near the runway taxi at faster speeds when congestion levels are low, but stop more frequently in the departure queue when congestion levels are high. In order to maintain these distinctions in aircraft behavior, it is important to characterize aircraft movement based on the taxiways they are on, and the amount of surface traffic at the airport. A network model of the airport surface is a natural choice with these requirements. Major taxiways form the links in this network, while their intersections form the nodes. Aircraft movement characteristics are captured by the travel time distributions on each link [109]. Formulating the problem in a network congestion control framework makes it possible to draw analogies with several related problems in literature, such as the design of wireless networks, manufacturing processes and urban transportation systems.

Additionally, this formulation is also independent of specific airports. Once the nodes and links are selected and the relevant surface surveillance data is supplied to the algorithm, it automatically computes the model parameters. This capability has been simulated and tested for airports other than Boston Logan, which is the primary airport considered in this thesis.

Control formulation for managing departure operations

The airport congestion management algorithms published in prior literature are typically based on operational targets such as maintaining a constant traffic level on the surface. The disadvantage of such methods is that the primary objective of taxi time and fuel burn reduction is obtained as an incidental benefit. There is no guarantee that the strategy being employed is near-optimal in terms of fuel consumption. Additionally, the procedures for handling operational constraints such as gate capacity are typically *ad hoc*. Unlike the ‘black-box’ functionality provided by these algorithms, the control strategy proposed in this thesis directly targets the primary objectives. It also allows the end-users (air traffic controllers) to tune the strategy depending on their preferences, with respect to the relative importance given to fuel burn, airport performance and gate delay. The network formulation is naturally amenable to a control strategy developed using dynamic programming techniques [110]. Realistic constraints such as airport gate capacity and arrival/departure demand can be integrated into the solution procedure [111], and the inputs required by the system are currently available in the air traffic control tower.

Control algorithms for arrival airspace

Safety is the primary emphasis of control algorithms for airborne operations, and is harder to ensure for the in-flight phase than for surface operations. Current air traffic control procedures for airport arrivals are designed for radar surveillance with human supervision. However, the introduction of satellite-based surveillance systems such as ADS-B in aircraft has made it possible to increase airspace capacity without compromising on safety. Chapter 4 presents a control strategy for arriving aircraft that utilizes on-board ADS-B technology and reduces congestion in the crowded airspace around an airport [42]. It is proposed that aircraft can adjust their own velocity in order to deconflict with other aircraft in the region. Information about the surrounding airspace is obtained using ADS-B, and is based on realistic models of communication channel capacity and reliability.

In addition to reducing the congestion in crowded airspace, the proposed method can also reduce the dependence on expensive ground-based radars for surveillance. The maneuvers recommended by the control algorithm can be implemented automatically, or can be provided in the form of advisories to pilots.

Integrated control of arrivals and departures

Chapter 5 combines the control strategies for departure control and arrival control into a single integrated framework. It describes how predictions of aircraft landing times can be used for real-time tuning of the departure control strategy. Simulation of the integrated control strategy shows that it increases the fuel burn savings in times of low arrival demand, while decreasing the number of constraint violations when the arrival demand is high. Chapter 5 also describes the implementation procedure for the ideas presented in thesis at a generic airport, starting from the modeling exercise and ending with a proposal for practical demonstration.

Chapter 2

Network Model of Airport Surface Operations

As explained in Chapter 1, current models for airport operations fall into one of two categories. They either form aggregate estimates of airport throughput, or require exact knowledge of the state of each aircraft. In order to balance the need to include stochasticity in airport operations with the desire to model the physical taxi process in greater detail, a new network-based model of the airport surface is presented in this chapter.

2.1 Background

Stochasticity in airport operations is introduced by many different sources. Chief among these is the dependence on human factors. Air traffic controllers, pilots, and ground crews all influence the time taken for execution of each step in the departure process. Additional constraints can be imposed by weather at the airport, in surrounding airspace and even at destination airports. These may require aircraft to absorb delay on the surface, or introduce smaller increases in taxi times while alternative flight routes are communicated to and accepted by the pilots. Because of these factors, the network model presented in this chapter explicitly includes the stochasticity of aircraft taxi times.

2.1.1 Data used for parameter estimation and model testing

All the results presented in this chapter and pertaining to Boston Logan International Airport, are based on surface surveillance data from the entire year 2011. Data from LaGuardia airport is limited to only six weeks from June and July 2011. In instances where the data is split into a training set and a testing set, the component days are selected randomly. In all figures in this chapter, the predicted distributions are based on the training data set and are denoted by the term *model*, while the distributions of the independent test data set are denoted by the term *empirical*.

2.1.2 Model structure

Figure 2-1 shows the set of runways and taxiways on the airport surface 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 be from the time an aircraft is first detected by the ASDE-X system to the time it starts its takeoff roll from the runway threshold. Therefore, the potential source nodes in the network are the ones adjoining the gates, while the sink / destination nodes are the runway thresholds. An abstraction of the resulting model is shown in Figure 2-2. Note that Figure 2-2 (a) shows the union of the networks for all possible airport configurations (allocations of runways to landings and takeoffs). In practice, only one configuration is active at a time, and each aircraft has only one source node (gate) and one destination node (runway). Figure 2-2 (b) shows the specific network for departures from Runway 27. In any specific configuration, aircraft maintain a flow from the terminal to the runway, and generally do not taxi in cyclic paths. Consequently, the configuration-specific networks are directed acyclic graphs with random link travel times. This acyclic nature is useful when analyzing network capacity, as described in Section 3.2.3.

2.1.3 Analysis of empirical data

Aircraft on the surface taxi at fairly constant speeds, occasionally stopping because of other aircraft crossing their path, or when about to cross an active runway. Notionally, the taxi-out process can thus be classified into two modes: *unimpeded taxi*, and *stopped*. Figure 2-3 shows the procedure adopted for determining the empirical distributions of travel times for each individual link. From the instantaneous speed information in the data, the number of stops made by each aircraft on a



Figure 2-1: Layout of the airport surface at Boston Logan. Nodes in the network model are marked with white boxes.

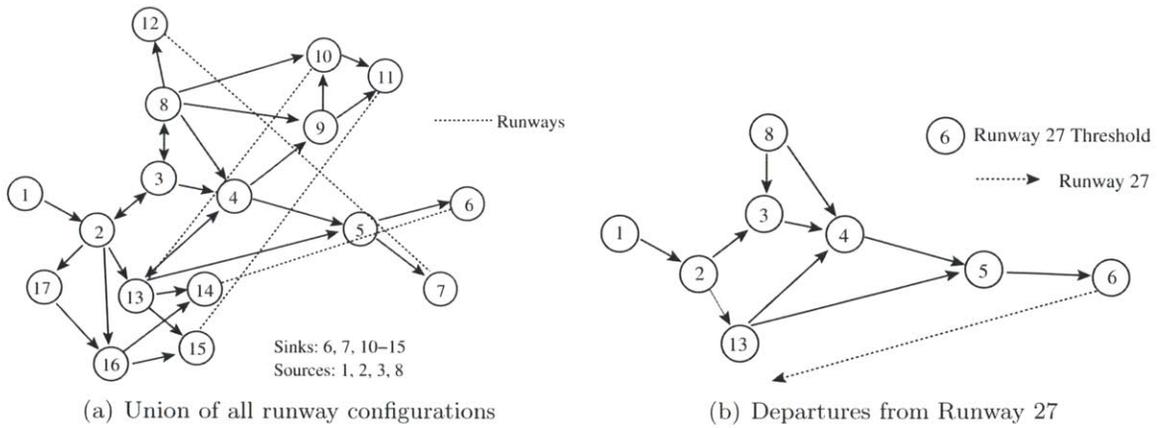


Figure 2-2: Abstraction of the network model.

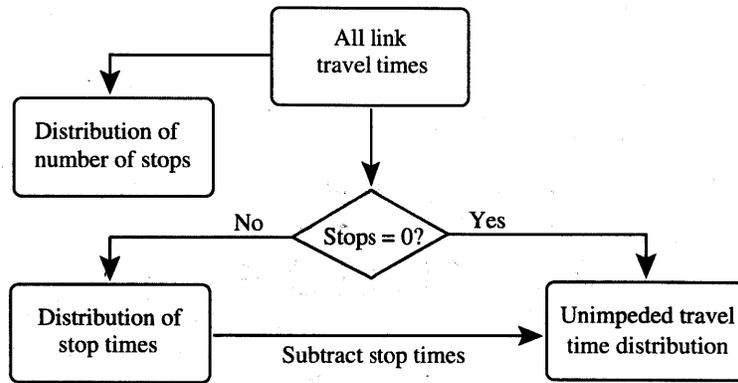


Figure 2-3: Flowchart for measuring empirical distribution of link travel times.

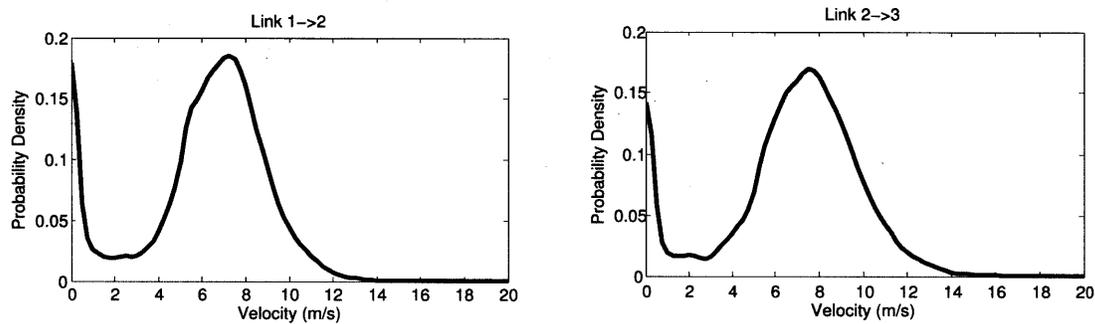


Figure 2-4: Distribution of taxi speed over the links 1→2 and 2→3.

given link can be counted. A speed threshold is defined in order to count the number of stops. From Figure 2-4, it can be seen that the typical taxi speeds of aircraft lie between 4 m/s (8 kts) and 10 m/s (20 kts). There is another peak in the distribution below 1 m/s (2 kts). A stop is therefore defined to be the number of instances when an aircraft's speed drops below 1 m/s on a given link. In order to avoid over-estimation due to slow taxiing near this value, a speed of 4 m/s has to be achieved between two logged stop events. A flight is defined to have passed *unimpeded* through a link if the speed of that flight stays above the 1 m/s threshold at all times on that particular link. The instances of unimpeded transit are used to generate the unimpeded taxi-time distribution for each link. The flights that have a non-zero number of stops are used for estimating the distributions for the number of stops and the duration of each stop. The taxi time left after subtracting the stopped times from these instances is also used for estimation of the unimpeded taxi time distribution, as shown in Figure 2-3. This ensures that the unimpeded distribution is not biased towards the high-speed instances of link travel times.

2.2 Model development

Based on the analysis of empirical data, aircraft taxi times can be modeled by a set of stochastic processes. This section progressively develops the model, starting with the component random processes and ending with the procedure for parameter estimation.

2.2.1 Random processes governing link travel times

There are three physical processes that govern the taxi time of an aircraft on each link in its path. Firstly, the unimpeded portion of the travel time is a direct result of the length of the link and the speed at which the aircraft moves. Secondly, the number of stops that an aircraft makes on each link depends on the level of congestion on the airport surface. Finally, the total time that an aircraft spends in the stopped mode depends on the duration of each stop. Three sets of theoretical distributions are therefore needed in order to explain the empirical data. Based on an inspection of the empirical distributions, the number of stops and the duration of each stop exhibit behavior similar to geometric and exponential random variables respectively (Figure 2-5). In addition to empirical observation, this behavior is also physically intuitive. It is reasonable to expect that the probability of stopping $N_{s,l}$ times on link l decreases monotonically with the number of stops $N_{s,l}$, and a similar argument can be made for the duration of each stop. The geometric and exponential distributions are the two simple analytical models that exhibit these properties. While this reasoning need not hold true for the links containing the departure queue (such as 5→6 in Figure 2-2), empirical data shows that the models are still valid for such links. As seen in Figure 2-5, the rate of decrease for link 5→6 is slower but is nonetheless monotonic.

The empirical unimpeded taxi time distribution for each link shows a marked peak, as seen in Figure 2-6. Several analytical probability distributions have this property, but are unsuitable for this problem because of the asymmetry also exhibited by the unimpeded taxi time distribution. There is a non-negligible probability of long taxi times on a link, but the minimum taxi times are limited by the physics of aircraft motion. While Gaussian models have been proposed in prior literature [112], they are unlikely to be good fits since the normal distribution is symmetric. The quality of fit for three candidate theoretical models has been measured in this thesis: Gaussian, log-normal and Erlang. The Kullback-Leibler (KL) divergence [113] is a natural measure of the ‘distance’ between two distributions, and is useful for comparison of the fit quality between each of

these models and the empirical distributions.

Let the three model families be denoted G (Gaussian), L (log-normal) and E (Erlang) respectively. Let $\chi_{\mathbb{M}}$ represent the set of parameters for family \mathbb{M} . For example, χ_G would consist of the mean and variance for the Gaussian family. Then the optimal set of parameters $\chi_{\mathbb{M}}^*$ for each model family \mathbb{M} is the one that results in the minimum KL divergence from the empirical distribution.

$$\chi_{\mathbb{M}}^* = \arg \min_{\chi_{\mathbb{M}}} D_{KL}(P||Q_{\mathbb{M}}(\chi_{\mathbb{M}})),$$

where $D_{KL}(P||Q_{\mathbb{M}}(\chi_{\mathbb{M}}))$ is the KL divergence from the empirical distribution, P , to the theoretical distribution, $Q_{\mathbb{M}}$. The optimal distribution, $Q_{\mathbb{M}}(\chi_{\mathbb{M}}^*)$, is called the information projection of P on $Q_{\mathbb{M}}$, and is denoted by $q_{\mathbb{M}}^*$ in Figure 2-9. It is now possible to compare the quality of fit possible using the three model families by comparing the KL divergence of q_G^* , q_L^* and q_E^* from P . For the current problem, the Erlang family is consistently closest to the empirical distribution for all links in the network. Therefore, the unimpeded taxi time distribution is modeled by an Erlang random variable. Parameter estimation can now be carried out for each link in the network using the three underlying random processes. Sample fits between empirical data and the theoretical models are shown in Figures 2-6, 2-7 and 2-8 for various links in the network for Boston Logan.

It is worth noting that these estimates are based on ASDE-X data from 2011, and therefore represent an average over all weather conditions and traffic levels. The variation of parameters with airport congestion levels is investigated in Section 2.2.2.

2.2.2 Variation of parameters with surface traffic

There are two primary factors which can affect the movement of aircraft on the surface. The first one is bad weather, which can potentially slow taxi operations, and also limit taxi route options in the case of snow. However, quantifying the effect of weather is difficult. Firstly, the reporting of weather is not consistent enough to obtain clear relations between severity and performance degradation. Secondly, the same weather conditions at an airport can result vastly different operational performance depending on local visibility and weather in the surrounding region. The variation of parameters with weather is not within the scope of this thesis. It is recognized that the effect can be significant, and the adaptive parameter estimation algorithm described in Section 3.2.8 can compensate for some of this variation.

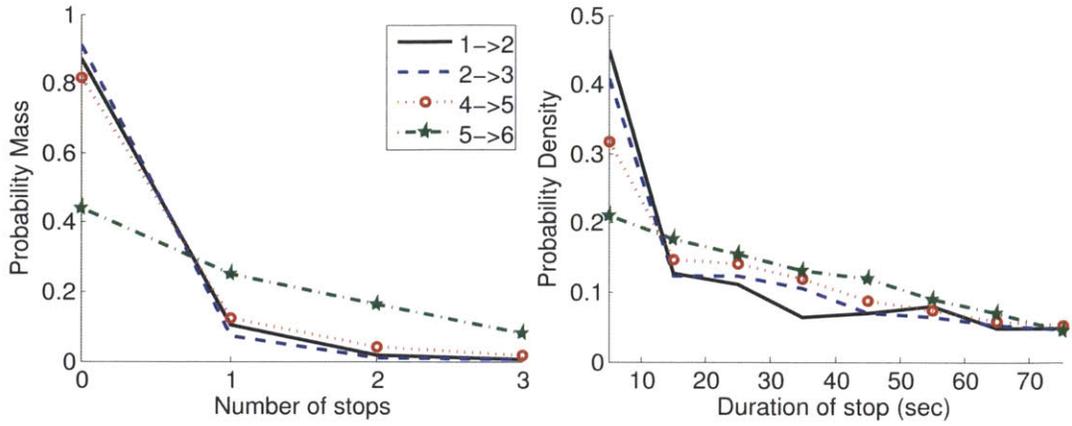


Figure 2-5: Empirical distribution of number of stops and duration of stops, for four links.

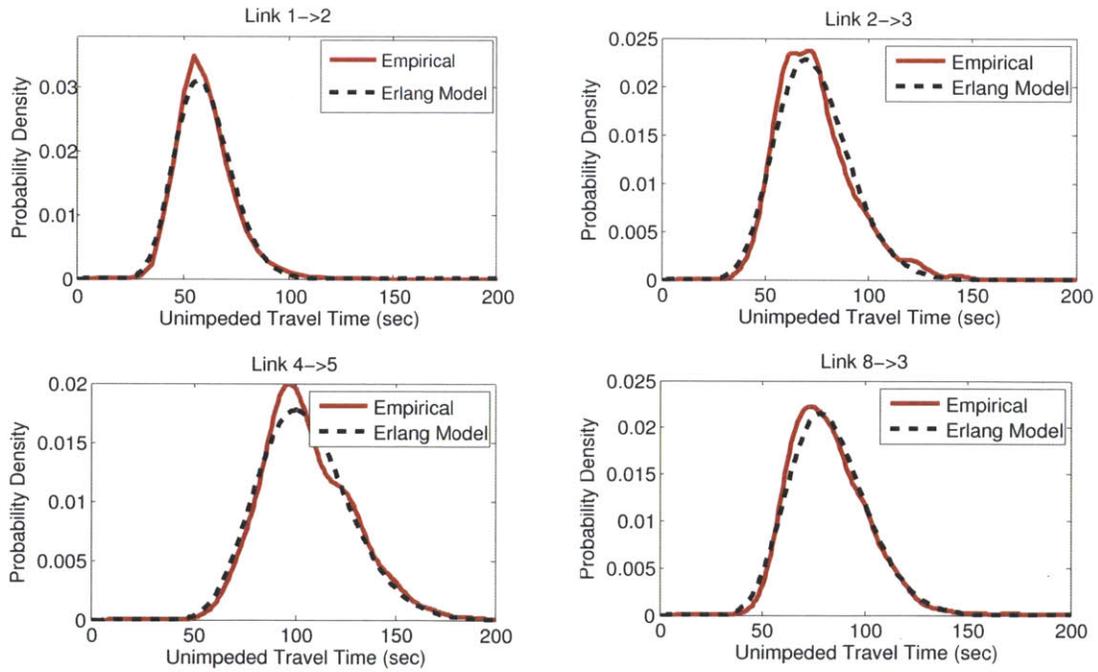


Figure 2-6: Unimpeded taxi times on various links. The model is estimated using the training data set, while the empirical distribution is based on an independent test data set.

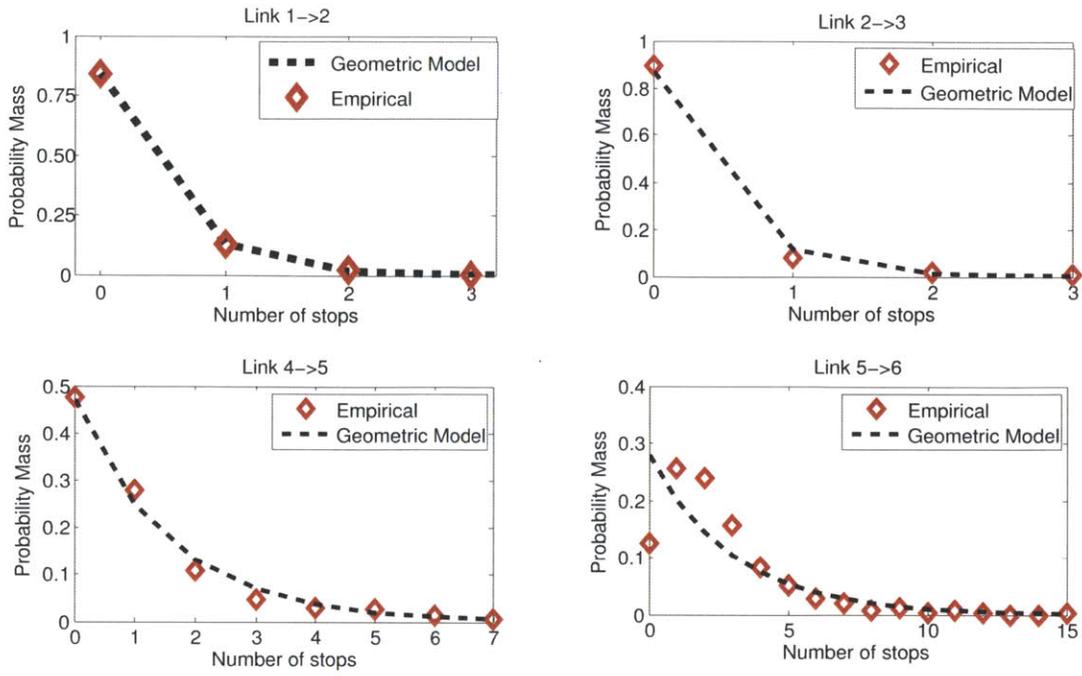


Figure 2-7: Number of stops on various links.

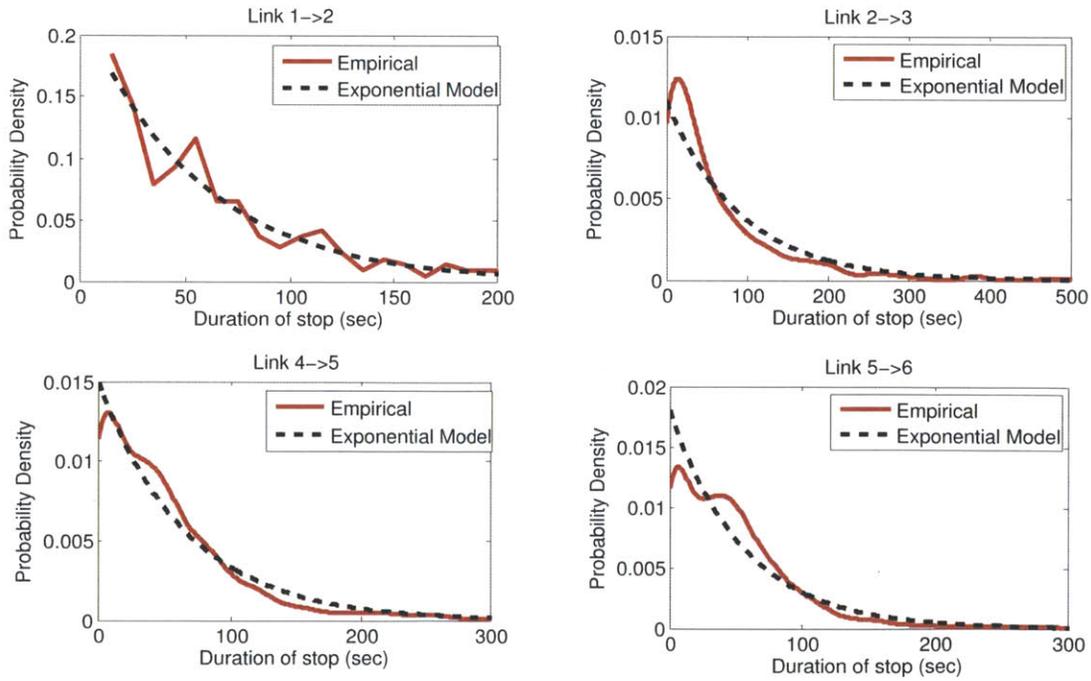


Figure 2-8: Stop durations on various links.

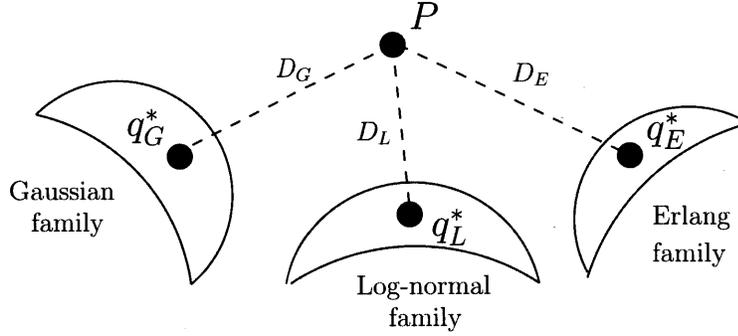


Figure 2-9: Information projection of the empirical distribution P on three candidate model families. D_G , D_L and D_E are the minimum KL divergence values from P to each family, and it is consistently seen that $D_E < D_L < D_G$.

The second factor that is known to affect taxi times is surface traffic. It is well known that taxi-out times at airports increase with increased surface congestion, and this fact needs to be accounted for by the model. The *surface traffic level*, k , is defined as the total number of departures that have pushed back from their gates but have not taken off yet (this quantity is sometimes denoted as N in prior literature [37, 61, 72]). The empirical data used in Section 2.2.1 can be divided based on the traffic level k seen by each aircraft when it pushes back from its gate. Since data from all of the year 2011 is included in this estimation process, there is a sufficient number of flights corresponding to each value of k for the results to be reliable. Empirical evidence compiled from this study shows that the unimpeded travel time parameters, n_l and λ_l , as well as the stop time parameter, μ_l , remain invariant with changes in k . The additional taxi-out time due to congestion is accounted for by an increase in the stopping probability on each link, p_{kl} . The evidence also shows that the average taxi time on each link increases approximately linearly with k . Sample plots for Links 1→2 and 4→5 are shown in Figure 2-10 to illustrate these points. The plots show that a significant number of data points are available between approximately $k = 5$ and $k = 20$. There is a steady increase in average taxi times on the link when plotted against total surface traffic. Similar behavior is exhibited by the other links in the network. Modeling this increase by a linear function results in a physically intuitive and mathematically tractable formulation. In addition, Figure 2-13 corroborates the linearity assumption by analyzing the empirical data from a different standpoint. The exact relationship between p_{kl} and k as indicated by this analysis is derived in Section 2.2.4.

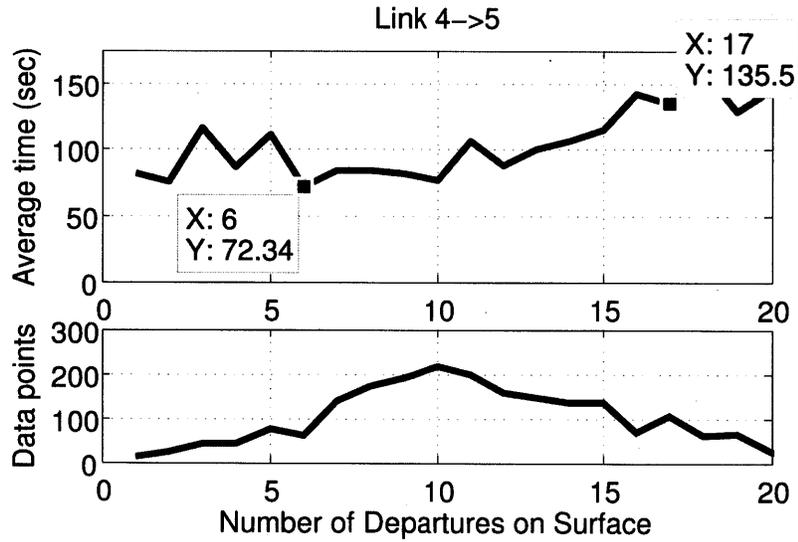
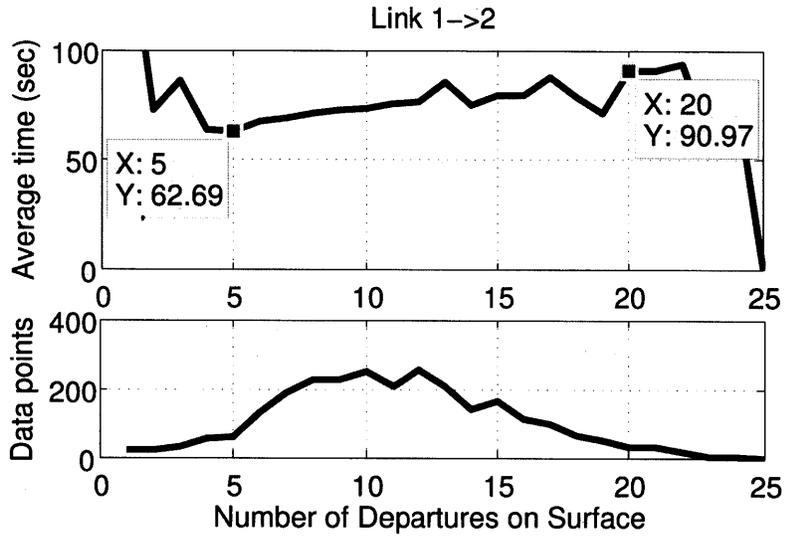


Figure 2-10: Variation of travel times over Links 1→2 and 4→5 with total traffic on the surface.

2.2.3 Model for total taxi-out time from gate to runway

Based on the discussion in Sections 2.2.1 and 2.2.2, the travel time over a link l is modeled as,

$$t_l = t_{u,l} + \sum_{i=1}^{N_{s,l}} t_{s,l,i} \quad (2.1)$$

Here, $t_{u,l} > 0$ is the unimpeded travel time over the link l , an Erlang random variable with order n_l and rate λ_l . $N_{s,l} \in \{0, 1, 2, \dots\}$ is the number of stops on the link, modeled as a geometric random variable with parameter $p_{kl} \in [0, 1]$, where k is the current level of traffic on the surface and l is the current link. 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 $\mu_l > 0$. The values of $t_{s,l,i}$ are assumed to be independent and identically distributed (*i.i.d.*).

If the number of stops is $N_{s,l} = 0$, then $t_l = t_{u,l}$. Each instance of travel time on a link is independent of all other instances, whether on the same link or on other links, conditioned on the level of surface traffic. This observation is derived from empirical data, which shows that surface traffic level as a whole has a greater influence on the taxi time of a given aircraft than aircraft in the immediate vicinity (such as on the same link). Figure 2-11 shows that the average taxi time on Link 1→2 does not increase appreciably with the number of aircraft occupying the link, in contrast to its variation with total traffic levels, as shown in Figure 2-10. The explanation for this is that most departing aircraft on the surface taxi towards a common destination (the runway), and there are few instances of interaction between successive aircraft on a given taxi path. The effect of the few such instances is captured by conditioning on the total surface traffic level.

2.2.4 Variation of stopping probability on links with surface traffic level

It was shown in Section 2.2.2 that the increase in taxi times with k was linear, and this was accounted for by an increase in p_{kl} . The relation between p_{kl} and k can be derived by taking the expectation of both sides of Equation (2.1), and noting that $N_{s,l}$ is independent of $t_{s,l,i}$:

$$\mathbb{E}[t_l | k] = \frac{n_l}{\lambda_l} + \mathbb{E}[N_{s,l} | k] \frac{1}{\mu_l} = \frac{n_l}{\lambda_l} + \frac{p_{kl}}{1 - p_{kl}} \frac{1}{\mu_l}. \quad (2.2)$$

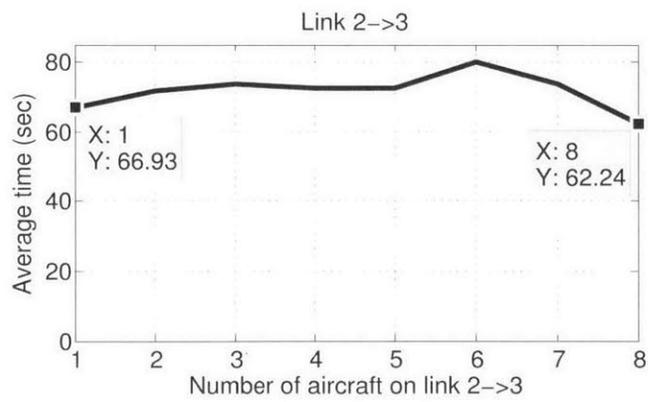
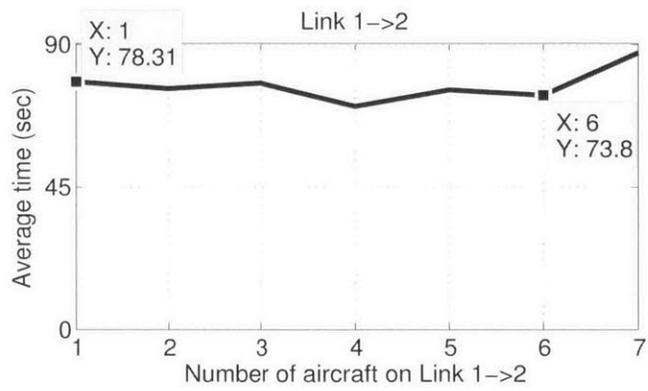


Figure 2-11: Variation of travel times over Links 1→2 and 2→3 with traffic on the same link.

Table 2.1: Summary of physical processes and stochastic models in the taxi process

t_l	Total taxi time on l	
$t_{u,l}$	Unimpeded time	Erlang(n_l, λ_l)
$N_{s,l}$	Number of stops	geometric(p_{kl})
$t_{s,l,i}$	Duration of i^{th} stop	exponential(μ_l)
k	Surface departure traffic level	
X_l	Dependence of p_{kl} on k	

Each additional aircraft on the surface adds a fixed time penalty to the expectation in Equation (2.2). Denoting this penalty as a fraction X_l of the expected stop time $\frac{1}{\mu_l}$, it may be observed that for an aircraft that pushes from its gate at a traffic level of $(k + 1)$ instead of k ,

$$\mathbb{E}[t_l | k + 1] = \frac{n_l}{\lambda_l} + \frac{p_{kl}}{1 - p_{kl}} \frac{1}{\mu_l} + \frac{X_l}{\mu_l}. \quad (2.3)$$

Comparing Equation (2.3) with Equation (2.2) evaluated at $(k + 1)$, the following recursive relationship is seen:

$$\frac{p_{k+1,l}}{1 - p_{k+1,l}} = \frac{p_{kl}}{1 - p_{kl}} + X_l.$$

This property describes a telescoping series and can be reduced to,

$$\frac{p_{kl}}{1 - p_{kl}} = \frac{p_{0l}}{1 - p_{0l}} + kX_l \quad (2.4)$$

Now solving for p_{kl} as a function of k we get,

$$p_{kl} = \frac{p_{0l} + kX_l(1 - p_{0l})}{1 + kX_l(1 - p_{0l})}. \quad (2.5)$$

The various component physical processes and their stochastic models are summarized in Table 2.1.

2.3 Model verification

There are several tests that can be performed in order to verify the taxi time model presented above. Since it is difficult to carry out standard statistical tests for a complex set of distributions

such as the ones presented in Table 2.1, the model is tested using metrics such as consistency with independent empirical data.

2.3.1 Consistency of estimated parameters

The consistency of parameter estimates for n_l , λ_l and μ_l can be tested by looking at their variation over independent sets of data. Note that these three parameters are independent of surface traffic levels. Figure 2-12 shows the estimates for the unimpeded taxi parameters n_l and λ_l , and the stop time duration μ_l , for a subset of links. The presented data consists of seven months of ASDE-X data from February to August 2011, consisting of an average of 587 departures per day. The parameter estimation process has been carried out for one month of data at a time. The estimated values for each link have fairly small standard deviations, with the exception of the unimpeded parameters for link 8→10. Further investigation reveals that there is a significant deviation for link 8→10 in only one month of data (April 2011), while the parameter estimates for the other six months are close to each other. This anomaly is therefore believed to be due to a lack of sufficient data points in that month.

2.3.2 Comparison of variable parameters with empirical data

A second check is to compare the variation of p_{kl} with k as presented in Equation (2.5) with independent empirical data. Figure 2-13 shows the comparison for two sample links, 5→7 and 1→2. It can be seen that the model-derived variation in p_{kl} is consistent with empirical data. The empirical probability of stopping depicted in the figure is based on a data set independent of that used for the parameter estimation. It has been calculated by looking at the fraction of aircraft that stopped on a link, as a function of the level of traffic on the surface when the aircraft entered the link.

2.3.3 Comparison of taxi-out times conditioned on surface traffic levels

A comparison can also be carried out between the modeled and empirically measured taxi time distribution on a link, as conditioned on the number of stops $N_{s,l}$. Figure 2-14 shows the results for the link 1→2. The theoretical distribution for taxi times conditioned on $N_{s,l} = 1$ can be calculated by convolving the unimpeded Erlang distribution with the exponential distribution for the duration

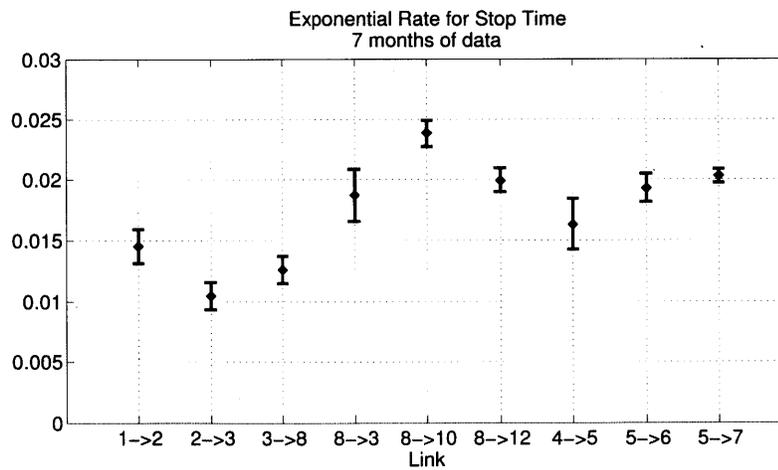
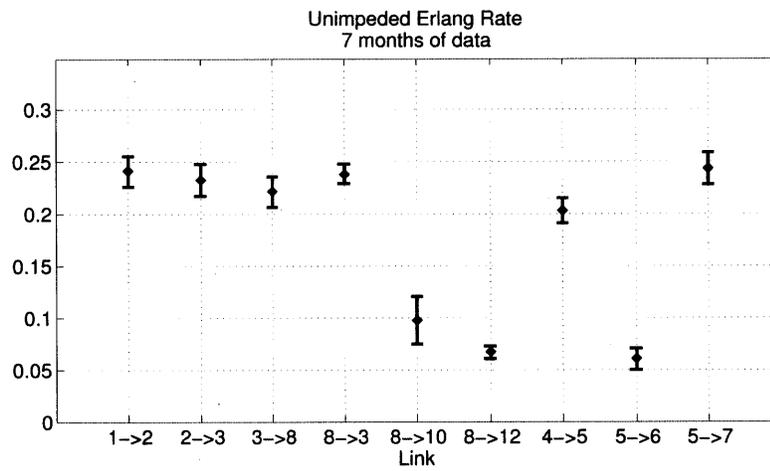
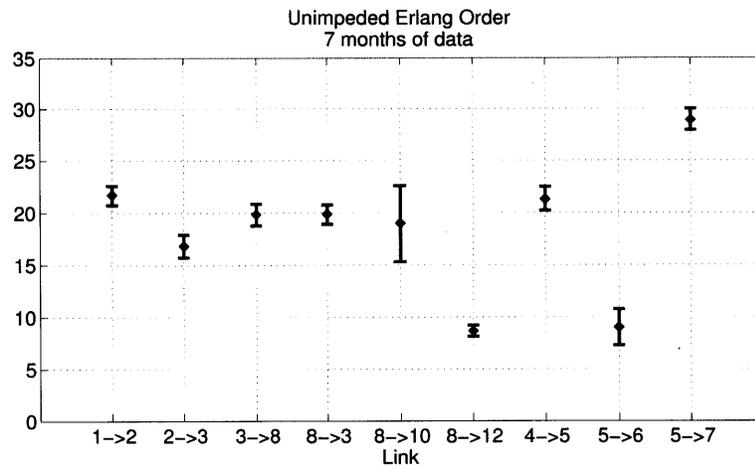


Figure 2-12: Means and standard deviations of the parameters for several links at Boston Logan.

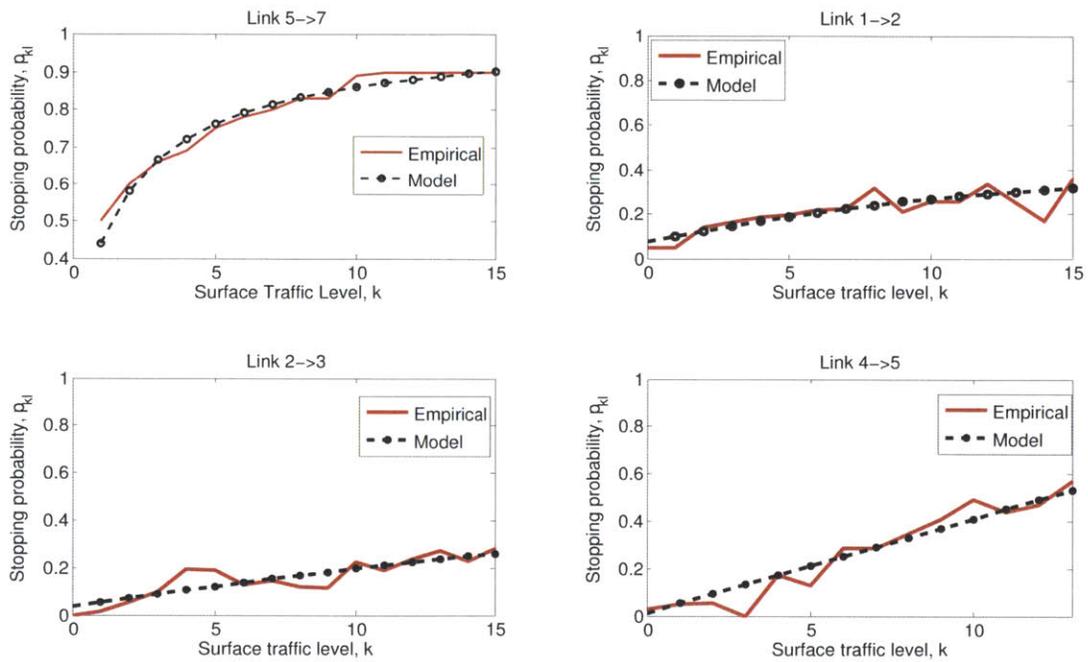


Figure 2-13: Variation of stopping probability on four links with the level of surface traffic. Note that link 5→7 holds the departure runway queue and hence has both higher stopping probability as well as the greatest sensitivity to k .

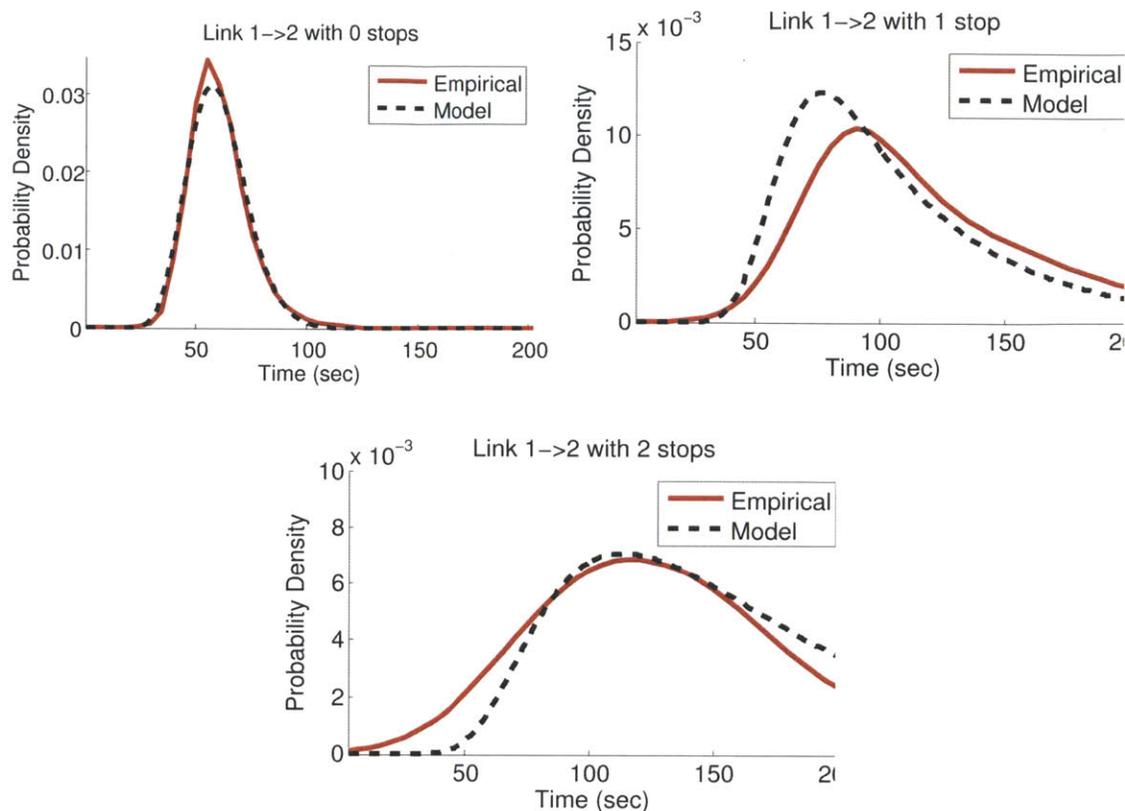


Figure 2-14: Comparison of travel time distributions conditioned on number of stops.

of stop. Similarly, the distribution for taxi times with 2 stops is the result of convolution between the unimpeded Erlang and two exponential distributions. It can be seen from the figure that there is good agreement between the theoretical prediction and empirical measurement in all three cases.

A brief observation can also be made about the dependence of the expected number of conflicts (hence stops) on the surface, as a function of the surface traffic. Note that the model represented in Equation (2.4) implies that the number of stops on the surface increases linearly with the traffic level k . Since this is true for every aircraft, and the number of aircraft themselves is trivially linear with respect to k , the total number of stops or conflicts on the surface increases quadratically with k . This is in agreement with empirical observations made in prior literature [114].

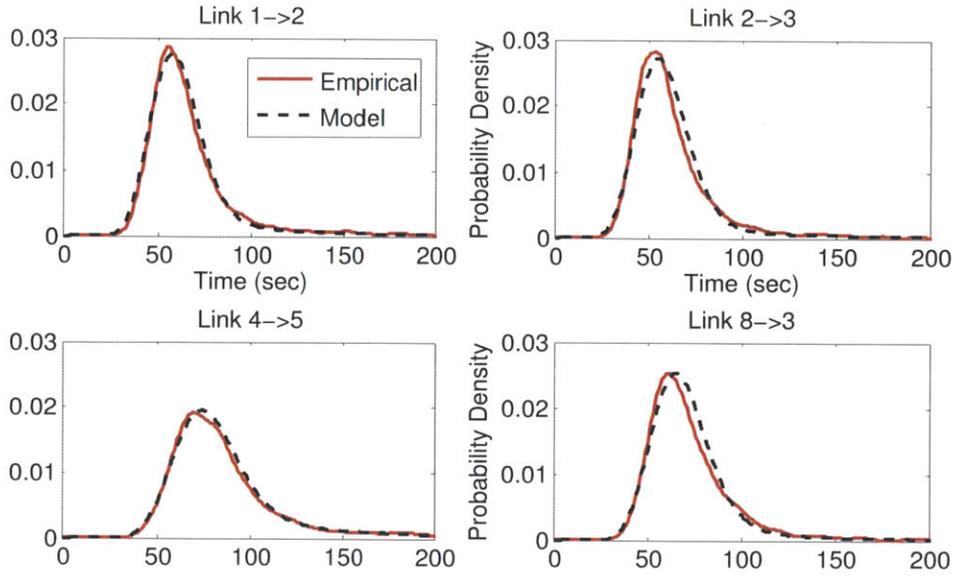


Figure 2-15: Comparison of full travel time distributions over different links in the network. The dashed curves (model) are the distributions as predicted by the taxi time model, based on parameters estimated using a training data set. The solid curves (empirical) are the observed distributions in an independent test data set.

2.3.4 Comparison of full taxi-out time distributions on links

Finally, the complete theoretical taxi time distribution can be calculated by combining the distributions for all possible values of $N_{s,l}$. This computation is carried out numerically because of the complexity of the analytical procedure. A sample comparison between the empirical and the theoretical probability densities for taxi times on several links marked in Figure 2-1, is shown in Figure 2-15. As before, the empirical distribution used is from a set of days independent of those used for training the model. Similar matches are seen between the empirical data and the model for all frequently used links.

2.4 Additional insights

The tuned values of distribution parameters for each link offer interesting insights into the operational characteristics of the airport. Some sample values are given in Table 2.2. For example, the mean unimpeded travel time over each link, given by $\frac{\mu_l}{\lambda_l}$, is an indicator of the length of link l . This interpretation is supported by the parameter estimates, with long links tending to have higher

Table 2.2: Parameter values for some sample links. A full list can be found in Appendix A. Note that the quantity $\frac{n_l}{\lambda_l}$ denotes the expected unimpeded time, while $\frac{X_l}{\mu_l}$ is the expected taxi time added for each unit increase in k .

Link	λ_l (sec ⁻¹)	n_l	$\frac{n_l}{\lambda_l}$ (sec)	μ_l (sec ⁻¹)	p_{0l}	X_l	$\frac{X_l}{\mu_l}$ (sec)
1→2	0.24	20	83	0.020	0.20	0.03	1.5
2→3	0.23	18	78	0.015	0.20	0.03	2.0
8→3	0.23	19	83	0.017	0.15	0.02	1.2
4→5	0.20	22	110	0.015	0.25	0.15	10.0
5→6	0.10	12	120	0.015	0.50	0.45	30.0

unimpeded times than short links. In addition, the final links leading to runway thresholds, which is when pilots finish the final checks before takeoff, generally have lower λ_l values than links deep inside the network. This means that the model has recognized from the training data, that taxi times on the links leading to the runway threshold tend to have greater variance than those near the gates (the variance of an Erlang distribution is $\frac{n_l}{\lambda_l^2}$). The parameters of the exponential random variables offer insights into the trend of stop times on each link. Specifically, the quantity $\frac{1}{\mu_l}$ is the mean time of a stop on the link. This tends to be between 40 and 70 sec. Again, μ_l values tend to be lower on the final links, which means that the links that contain departure queues have longer stops than others. The effect of the surface traffic level on link travel times is seen by the ratio of X_l to μ_l . Links near the terminals, such as 1→2 or 2→3, are affected to a much smaller extent than those near the runways. Note that Table 2.2 implies that for an aircraft taking the taxi path 1 → 2 → 3 → 4 → 5 → 6, each unit increase in the surface traffic level increases its expected taxi time by 43.5 sec.

2.4.1 Parameter estimation for the taxi-in process

The model for the taxi-in process is analogous to the one for taxi-out, with the time along each link depending on the three random processes described in Section 2.2.1. The unimpeded time distributions are quite close to those seen for departures (compare Figure 2-18 and Figure 2-6). However, the stop times are shorter on average, and the stopping probabilities themselves are lower. The full set of parameters is given in Appendix B. Note that in most runway configurations, arrival taxi paths are different from the departure taxi paths apart from the links closest to the airport terminals. Multiple taxiways are available on these links as well, thus ensuring that arrival taxi-in

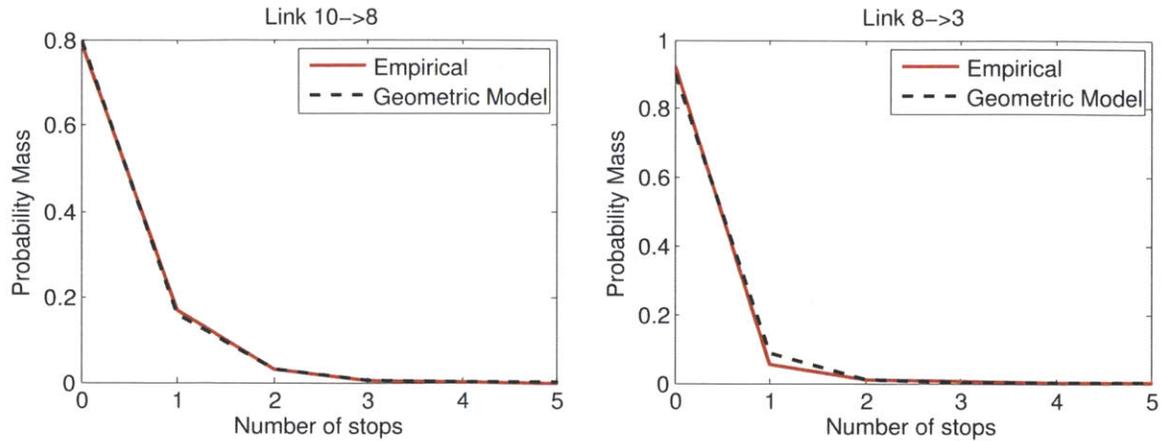


Figure 2-16: Arrival stopping probabilities.

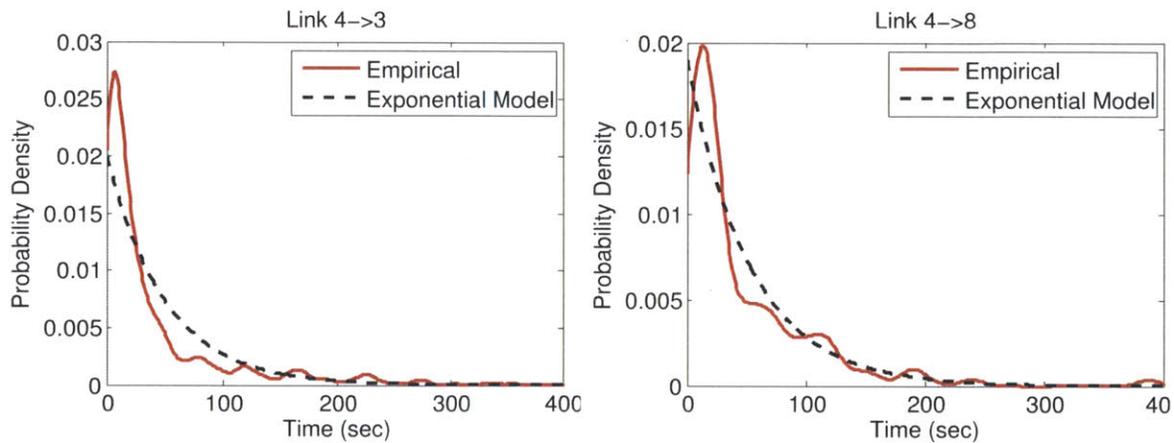


Figure 2-17: Stop time distribution for arrivals.

operations are largely separate from departure taxi-out operations. A comparison of the estimated distributions and empirically measured ones is shown in Figures 2-16, 2-17 and 2-18 for a selection of links. Note that the figures are representative of the quality of fit for all frequently used links, and are not chosen on this basis.

2.4.2 Parameter estimation for LaGuardia airport

While Boston Logan International Airport is used as a demonstrative example throughout this thesis, it is important to note that the methods proposed are extendable to any generic airport. The estimation of parameters and calculation of control policies is fully automated, once the network

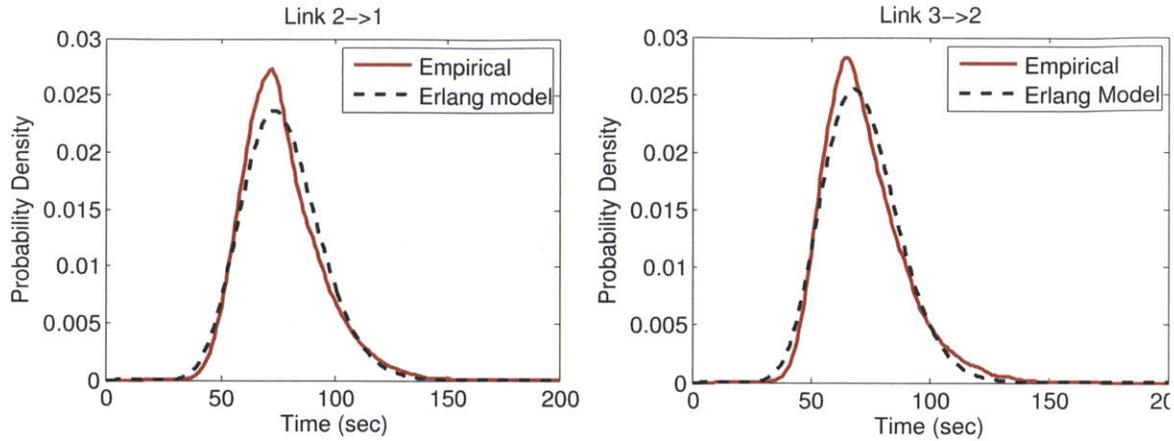


Figure 2-18: Unimpeded time distribution for arrivals.

has been defined for a given airport. An illustrative example of this is shown in Figure 2-19, which is the network as defined for LaGuardia airport in New York. The nodes are marked by white boxes as in Figure 2-1. The parameter estimation process is the same as that described in Section 2.2.1, and the resulting distribution fits are shown in Figure 2-20 for a sample set of links. The empirical data depicted in the figure is independent of the data used for parameter tuning. The quality of fit is seen to be good, which lends credence to the claim that the random processes postulated for modeling taxi times are independent of the specific airport under consideration.



Figure 2-19: Layout of the airport surface at LaGuardia airport. Nodes in the network model are marked with white boxes.

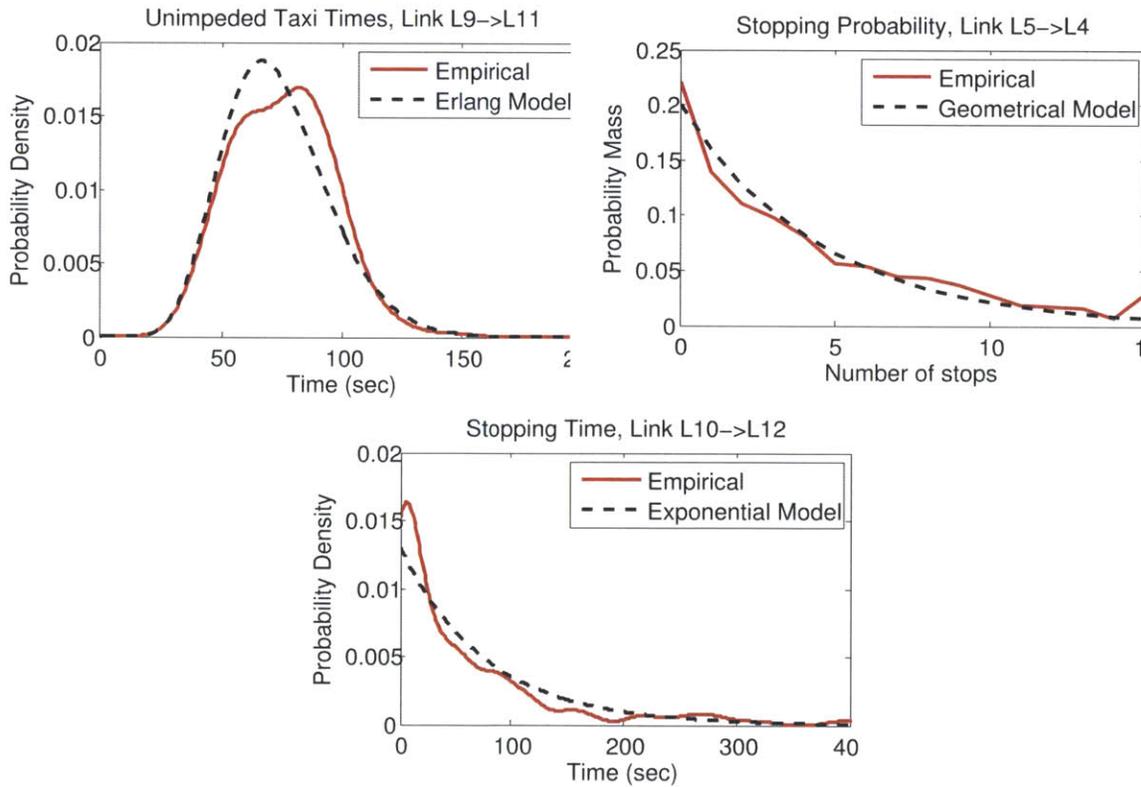


Figure 2-20: Comparison of model for LaGuardia airport with independent test data.

Chapter 3

Congestion Control of Departure Operations

This chapter presents two control algorithms for calculating the optimal time of entry into the network for each aircraft. The first algorithm aims to maintain a constant level of traffic on the airport surface, while the second algorithm simultaneously balances the objective of reduced fuel burn with the penalty of introducing gate delays and of adversely affecting airport performance.

The departure process at most airports starts with a flight plan clearance, which is relayed to the pilots on first making contact with air traffic control. At Boston Logan, pilots are then requested to contact the tower when the aircraft is ready for pushback: this happens when all passengers and cargo have been loaded and the aircraft doors are closed [115]. The controller makes visual contact with the aircraft parked at the gate, and ensures that the area around it is clear of obstructions such as ground vehicles and other taxiing aircraft. He/she then clears the aircraft for pushback, and assigns it a specific taxi route to the runway. A slightly different procedure is followed by aircraft at Terminal A. These aircraft first contact a ramp tower for permission to pushback. Their control is passed to the ATC tower when they reach the location marked as node 1 in Figure 2-1.

Other airports may have different divisions of responsibility between the ramp towers and the ATC tower. However, it is important to note that each aircraft is controlled by a ramp or ATC tower from the moment of pushback. It is operationally feasible to implement the control algorithms presented in this chapter with some amount of coordination between the various tower facilities. Several field trials are already in progress at many locations as described in Section 1.2.2, where the

implementation protocols are similar but the methods for calculating control policies are different. The advantage of the control algorithm presented in Section 3.2 is that it directly optimizes for fuel burn and airport performance, instead of achieving these benefits indirectly. In addition, it also adheres to operational constraints such as the gate capacity at the airport.

Through the remainder of this thesis, several terms are used in the context of applying the departure control algorithm at airports. These terms are defined here for the sake of clarity. A *gate* is a parking bay for aircraft attached to the airport terminal, where passengers board and deplane. *Pushback* is the process of pushing an aircraft back from the gate, in preparation for taxi to the runway. Aircraft do not start their engines until pushback is completed. *Pushback delay* is an instruction given to an aircraft, delaying the start of its pushback process. The aircraft is supposed to wait at the gate until it is cleared for pushback. Finally, a *pushback buffer* is the number and occupancy state of a collocated group of gates.

3.1 The k_{ctrl} algorithm: maintaining constant traffic levels

This section describes a control algorithm that aims to maintain a steady level of traffic on the airport surface. It has been shown that regulating traffic on the airport surface to a well-chosen level results in reduced taxi times without delaying takeoff times [61]. Since aircraft taxi times are roughly proportional to their fuel burn [108], reduced taxi times result in fuel savings. Therefore, this objective is a logical starting point for any new control strategy. Lessons learned from the analysis of this control strategy are used to develop a more comprehensive algorithm, which is described in Section 3.2. Note that both algorithms assume that control is exercised by changing the time of entry of aircraft into the network, and the taxi routes after pushback are assumed to be pre-defined.

3.1.1 Objective and implementation protocol

The objective of this control algorithm is to stabilize the airport traffic level at a user-defined operating point. The idea is based on the N-control algorithms described in literature for airport congestion control [72, 25, 23], which themselves are based on ideas such as the control-point policy for manufacturing systems [102]. Control is exercised on the network described in Chapter 2 by defining the time of entry of aircraft into the network. Aircraft are handled on a First-Come-First-

Served basis, that is, aircraft are released from the gates in the same order that they call ready for pushback. Gate delays are assigned one at a time, starting from the first aircraft due for release. When this aircraft leaves its gate, a delay is calculated for the next aircraft and so on. In this section, the assigned delay is a function of only the surface traffic level k .

It is important to note the similarities and distinctions between the N-control algorithm described in prior literature, and the k_{ctrl} algorithm described here. Both strategies have the same goal, that of stabilizing the level of surface traffic to a user-defined level. However, the way in which these strategies are implemented is different. The pure N-control algorithm [72, 25, 23] is an on-off policy where the critical events are aircraft takeoffs. As soon as one aircraft takes off, the next aircraft is allowed to pushback. This means that the strategy is predicated on past and present events only, and provides no warning of the imminent release command to the pushing aircraft. The Pushback Rate Control algorithm [61, 73] is a predictive algorithm in the sense that it forecasts the number of takeoffs in the next few minutes, and produces a suggestion for the number of aircraft that can be released in that time period. It does not specify the times at which individual aircraft should be released, and instead employs somewhat conservative policies that ensure robustness to such variations. The k_{ctrl} algorithm presented and simulated in this thesis is also predictive in nature. However, the pushback times that it calculates are much more specific, but limited to only the next aircraft waiting to pushback.

3.1.2 Definition of control strategy

Consider a scenario where an aircraft calls ready to pushback at time $t = 0$. Now consider a First-Come-First-Served (FCFS) algorithm that aims to minimize a weighted linear cost function for the aircraft, composed of *pushback delay* u and *expected taxi-out time* $\mathbb{E}[t_l|k]$ from Equation (2.3). That is, the cost function is $\mathcal{C}_k = a \cdot u + \mathbb{E}[t_l|k]$, with a being a constant weighting factor. Its relationship with the target traffic level is developed towards the end of this section. The control value u that minimizes expected cost is given by

$$\begin{aligned} u &= \arg \min_{u \geq 0} \mathbb{E}[a u + t_l] = \arg \min_{u \geq 0} \left(a u + \sum_{l=l_1}^{l_r} \left[\frac{n_l}{\lambda_l} + \frac{p_{k_{pl}}}{1 - p_{k_{pl}}} \frac{1}{\mu_l} \right] \right) \\ \Rightarrow u &= \arg \min_{u \geq 0} \left(a u + \sum_{l=l_1}^{l_r} \left[\eta_l + \frac{k_p X_l}{\mu_l} \right] \right), \end{aligned} \quad (3.1)$$

where the last two steps follow from Equation (2.2) and by defining $\eta_l = \frac{\eta_l}{\lambda_l} + \frac{1}{\mu} \frac{p_{0l}}{1-p_{0l}}$. Equation (3.1) assumes that the aircraft's route follows links $l = l_1, l_2, \dots, l_r$, and that $p_{k_p l}$ is the stopping probability on link l , when the projected surface traffic level after time u is k_p . The expected taxi time is thus also a function of u . The projected traffic level, k_p , can be calculated based on the procedure outlined in Section 3.1.3. Since entry into the network is assumed to be FCFS, the projected traffic level decreases as u increases (there can be no additional aircraft entering the network while the first aircraft is waiting). It is shown later in Equation (3.9), that the expected time between departures increases as k_p decreases. Therefore, at some value of k_p , the increase in expected cost due to the first term in Equation (3.1) outweighs the decrease in cost due to a smaller k_p in the second term. The actual value of this 'target' k_p is controlled by the weight a . In the rest of this section, the target level of traffic is denoted by k_{ctrl} .

3.1.3 Simplified model of departure process from the network

To calculate the optimal control policies, some simplifying assumptions about the departure process need to be made. The following derivation will be helpful in providing intuition about the departure process. Note that the highly simplified model works for this algorithm because projections about the future are only carried forward up to the time of the next pushback. A more sophisticated approximation procedure is required for the dynamic programming algorithm described in Section 3.2, where a much longer time scale of projection is required. For simplicity, first consider the single-link network. For moderately large values of surface traffic k , it may be assumed that departures from the link occur as exponential processes, one for each aircraft. The mean time of arrival of each process is equal to the mean taxi time $\eta_l + \frac{kX_l}{\mu_l}$. If there are k aircraft on the link, there are k racing exponential processes. Using the memoryless property [116] and Equation (2.2), the expected departure time of each aircraft (relative to the present time) is again given by $\eta_l + \frac{kX_l}{\mu_l}$. Consequently, the rate of each process is the inverse of this quantity, and the net departure rate for k independent exponential processes is

$$R_{kl} = k \cdot \frac{1}{\eta_l + \frac{kX_l}{\mu_l}} \text{ aircraft per unit time.} \quad (3.2)$$

Since each departure from the link corresponds to an expected taxi time reduction of $\frac{X_l}{\mu_l}$ for the aircraft being assigned pushback delay, the instantaneous rate of decrease of expected taxi time

with the control input u is given by

$$-\frac{d}{du} t_l = R_{kl} \frac{X_l}{\mu_l}.$$

In order to target a certain level of traffic (say $k = k_{\text{ctrl}}$), the value of the weight is chosen such that the rate of reduction of expected taxi time is equal to the rate of increase of the term $a \cdot u$, when $k = k_{\text{ctrl}}$. Since the term $a \cdot u$ increases at the constant rate a , the relationship between the weighting factor a and the target traffic level k_{ctrl} is

$$a = -\frac{d}{du} t_l = R_{k_{\text{ctrl}}} \frac{X_l}{\mu_l} = \frac{k_{\text{ctrl}} \frac{X_l}{\mu_l}}{\eta_l + k_{\text{ctrl}} \frac{X_l}{\mu_l}}. \quad (3.3)$$

3.1.4 Optimal control strategy

Since no new aircraft enter the link until the current aircraft pushes back, R_{kl} is the rate of decrease of the projected traffic level k_p . It is therefore possible to find the optimal control u^* required to maintain a specific value of k_p . An implicit assumption is that the exponential nature of the departure process for each aircraft is maintained as this projected value evolves. If k_p is the projected traffic level,

$$\begin{aligned} \frac{d}{du} k_p &= -R_{k_p l} = -\frac{k_p}{\eta_l + k_p \frac{X_l}{\mu_l}} \\ \Rightarrow u^* &= \eta_l \ln \frac{k_p(0)}{k_p(u^*)} + (k_p(0) - k_p(u^*)) \frac{X_l}{\mu_l}. \end{aligned} \quad (3.4)$$

Since $k_p(0)$ is a known quantity (the current traffic level), the optimal control for each $k = k_p(0)$ is defined by substituting $k_p(u^*) = k_{\text{ctrl}}$ in Equation (3.4). If the optimal value is negative, u is assigned a value of zero, in order to obey the constraint $u \geq 0$. Note that this happens if and only if $k_p(0) < k_{\text{ctrl}}$, which means that the control strategy calls for immediate pushback if the traffic level is below the target value. If the current traffic level is above k_{ctrl} , the pushback delay becomes progressively larger with $k_p(0)$. Another point to note is that for every value of $k = k_p(0)$, there is a unique control u^* that is commanded. Thus there is a discrete (but infinite, since the possible values of $k_p(0)$ are infinite) set of commanded control values that is generated by this control strategy.

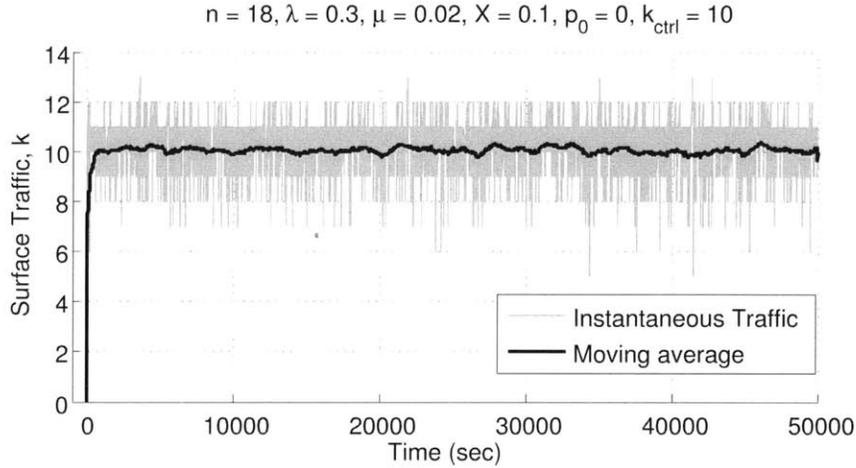


Figure 3-1: Variation of surface traffic levels seen in simulations of control algorithm for single link, with $k_{\text{ctrl}} = 10$. The assumed link parameters are listed at the top of the plot.

This property is used in Section 3.1.7, when considering the regulation characteristics of the control strategy described above.

3.1.5 Simulation of single link networks

The control strategy developed above is demonstrated through simulations in this section for a single-link network, and then for the full surface network of Boston Logan in Section 3.1.6. Since the optimal control strategy as derived above relies on a number of approximations, it is necessary to validate it using independent simulations. In Figure 3-1 and Figure 3-2, simulation results are shown for a single-link network with the control values u^* defined by Equation (3.4). It is assumed that there is an infinite buffer of aircraft waiting for release, and that the current traffic level is known at all times. In addition, aircraft are released according to a First-Come-First-Served (FCFS) policy. From Figure 3-1, the average steady-state traffic level is seen to stabilize to k_{ctrl} . Figure 3-2 shows that the average taxi times are close to the value of 105.5 sec predicted by Equation (2.2) for the assumed set of parameters. It is shown later in Equations (3.9) and (3.10) that the average taxi time penalty imposed by the presence of a finite number of aircraft on the surface is $\frac{\eta}{k_{\text{ctrl}}}$. In this example, this evaluates to 6 sec more than the theoretical minimum inter-departure times. This added average taxi time can be driven arbitrarily close to zero by increasing k_{ctrl} .

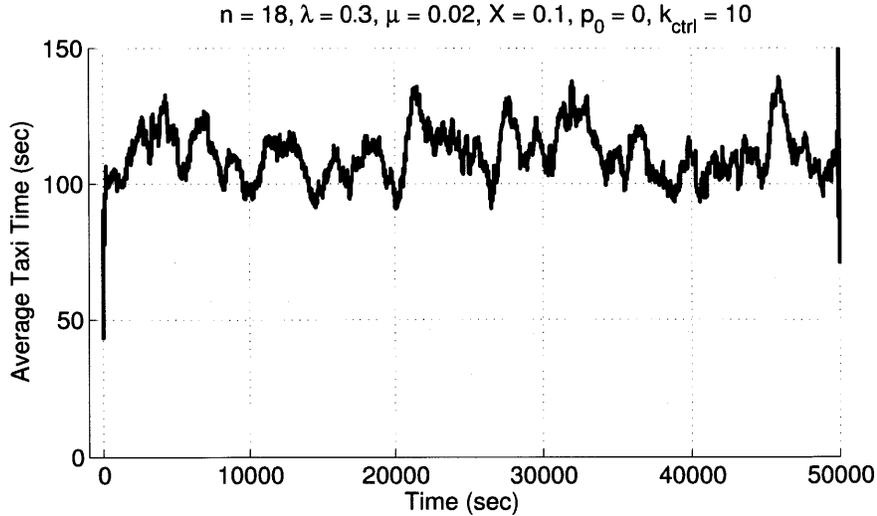


Figure 3-2: Variation of average taxi-out times seen in simulations of control algorithm for single link, with $k_{ctrl} = 10$. The assumed link parameters are listed at the top of the plot.

3.1.6 Simulation of the full airport network

Figures 3-3 and 3-4 show results from a simulation based on the full network abstraction of Boston Logan airport. All departures are assumed to happen from Runway 27 (marked as node 6 in Figure 2-1). The relevant network is depicted in Figure 2-2. It is assumed that the pushback requests from these aircraft appear at a rate based on the historical variation over a day, as derived from surface surveillance data. The x -axis shows the local time at the airport. Each curve in Figure 3-3 (a) plots the variation of simulated *average* traffic level in that 15 min interval over the course of the day, for different values of k_{ctrl} . In all the controlled cases, the algorithm successfully limits surface traffic levels to the corresponding target values in times of high departure demand. Since the same pushback schedule is used for all four tests, the total number of departures is the same when summed over the whole day. The curve for unrestricted entry (or $k_{ctrl} = \infty$) into the network clearly shows the morning and evening demand peaks, with high levels of surface congestion during these times. The cases with $k_{ctrl} = 15$ and $k_{ctrl} = 20$ mitigate this effect to a large extent, by delaying aircraft at the gate during periods of excessive demand. Note that the traffic peaks become wider with decreasing k_{ctrl} , as the control algorithm takes longer to clear out delayed aircraft. For $k_{ctrl} = 10$, the peaks are so wide that they merge with each other, and the algorithm is unable to clear the built-up demand until the end of the day. This value of k_{ctrl} is too aggressive, as seen from the

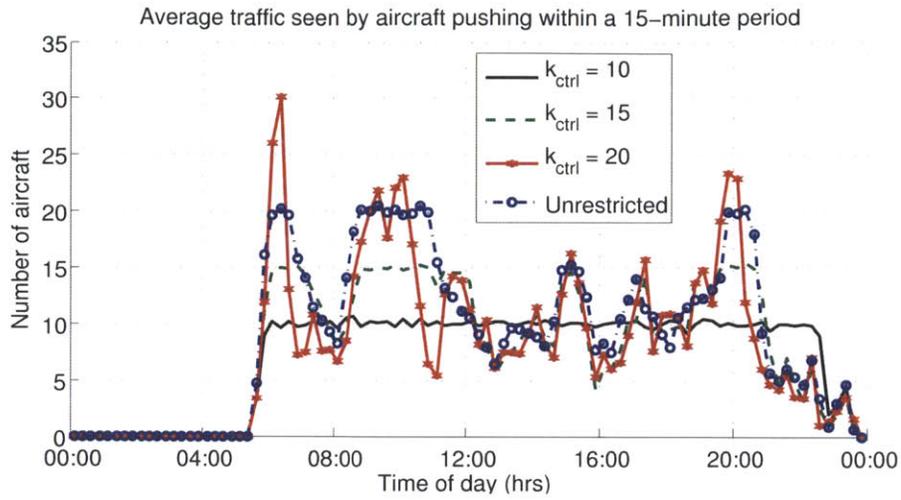
excessive pushback delays assigned in this case (Figure 3-4 (b)).

Practically, the choice of k_{ctrl} may be made on a case-to-case basis for each airport. It should be low enough to avoid gridlock on the surface, which leads to large and highly variable delays. At the same time, it should be high enough to keep up with pushback demand and the throughput capacity of the airport. For example, in the unrestricted case, a comparison of Figures 3-3 (a) and 3-3 (b) shows that the peak traffic level is higher than the maximum departure rate from the airport, leading to a buildup of traffic. For departures in this configuration, this maximum sustained takeoff rate is between 15.5 and 16.0 departures every 15 min. Note that the two peaks at 17 departures in Figure 3-3 (b) are due to takeoffs at the very start and end of the interval, and are accompanied by dips in throughput in the adjoining intervals. The maximum departure rate of around 16 aircraft per 15 min is rarely achieved by setting $k_{\text{ctrl}} = 10$. This is because the supply of aircraft is too low for 16 aircraft to reach the runway within 15 min on a sustained basis.

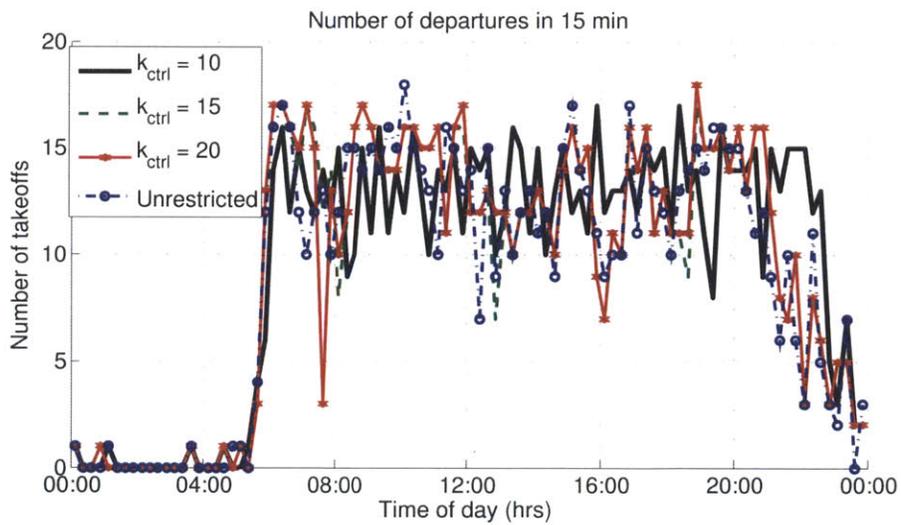
Once a feasible range of k_{ctrl} is derived from such considerations, the value can be tuned by deciding on the relative importance of one minute of taxi time reduction versus one minute of pushback delay. The parameter a in the objective function (Equation (3.1)) determines the relative weighting of taxi-out time reduction and pushback delay, assuming small perturbations from the initial state. Its value varies slightly depending on an aircraft's source node, since the expected taxi times are slightly different. Figure 3-4 (b) shows the average taxi-out times per 15-min interval by time of day. We notice that during peak demand times, taxi-out times for the unrestricted case can be as high as 24 min. By contrast, a value of $k_{\text{ctrl}} = 10$ (which corresponds to an average value of $a = 0.45$) results in average taxi-out times of around 12 min. Similarly, when $k_{\text{ctrl}} = 15$ (average $a = 0.55$), the algorithm regulates the system to the point where 1 min of additional pushback delay is traded off for 0.55 min, or 33 sec, of taxi time reduction.

3.1.7 Markov chain model with embedded control values

To analyze the effect of the control algorithm on the traffic level, it is useful to first consider the single-link case. As noted earlier, a fixed delay is assigned to each state, based on the target traffic level. Since departures from the link are stochastic, the state seen by the next aircraft to pushback is a random variable. It is possible to calculate the probability mass function of this state conditioned upon the previous state and the assigned delay. A Markov chain structure is seen to emerge, with

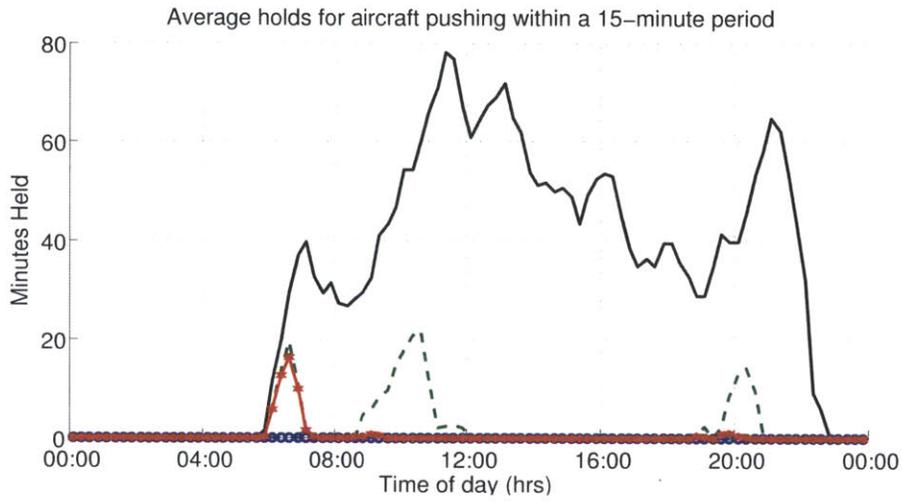


(a) Average traffic levels

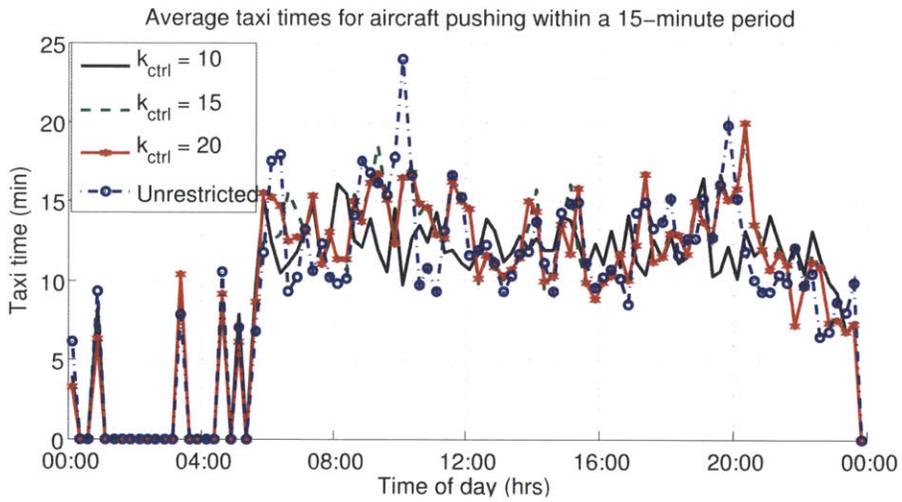


(b) Throughput per 15 min

Figure 3-3: Simulated traffic levels and throughput for the Boston Logan network, with departures from Runway 27.



(a) Average gate delays



(b) Average taxi-out times

Figure 3-4: Simulated gate delays and taxi times for the Boston Logan network, with departures from Runway 27.

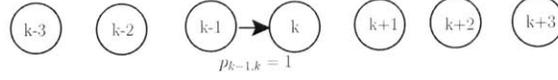


Figure 3-5: Markov chain model for transitions under the proposed control strategy, if $k < k_{\text{ctrl}}$.

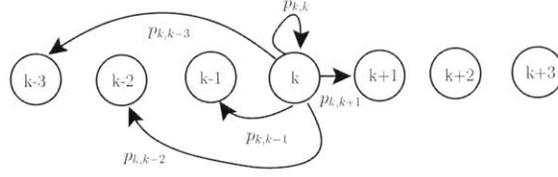


Figure 3-6: Markov chain model for transitions under the proposed control strategy, if $k \geq k_{\text{ctrl}}$.

the states corresponding to the traffic level seen by successive aircraft, and with known transition probabilities between each pair of states. Figures 3-5 and 3-6 depict this structure, assuming that the target traffic level is $k_{\text{ctrl}} = k - 1$. For states at or below k_{ctrl} , no delay is assigned according to Equation (3.4), and the transition is deterministic. For states above k_{ctrl} , the transitions are stochastic.

3.1.8 Calculation of transition probabilities

Consider the scenario depicted in Figure 3-6, in which the current aircraft sees a traffic level of k . Then, according to Equation (3.4), it is assigned a pushback delay

$$u^* = \eta_l \ln \frac{k}{k-1} + (k - (k-1)) \frac{X_l}{\mu_l}.$$

From Equation (3.2), the Poisson departure rate from the link is given by $R_{kl} = \frac{k}{\eta_l + \frac{kX_l}{\mu_l}}$. Note that a transition from state k to $(k+1)$ will occur if no aircraft depart from the link in time u^* . The probability density of the Poisson distribution results in the probability of this transition being,

$$p_{k \rightarrow k+1} = \frac{(R_{kl}u^*)^0 e^{-R_{kl}u^*}}{0!} = e^{-R_{kl}u^*}. \quad (3.5)$$

In a similar manner, the transition probabilities $p_{i \rightarrow j}$ can be calculated for each pair $\{i, j\}$. The chain may be truncated at a state k_{max} sufficiently larger than k_{ctrl} , such that the probability of ever reaching state k_{max} is very low. The matrix of transition probabilities $p_{i \rightarrow j}$ is nearly stochastic for a sufficiently large value of k_{max} . Steady state occupation probabilities for $k = 0, 1, \dots, k_{\text{max}}$

are given by the left eigenvector of this matrix. For the given system, these steady state values correspond to the proportion of traffic states seen by successive aircraft, when released according to the control algorithm defined by Equation (3.4). As shown in Figure 3-7 for $k_{\text{ctrl}} = 10$, these theoretical values match well with fractional state occupancies observed during simulation runs. The system is found to occupy one of the states k_{ctrl} , $k_{\text{ctrl}} \pm 1$ and $k_{\text{ctrl}} \pm 2$ more than 95% of the time.

Furthermore, the same result can be easily extended to more complicated networks. For the network depicted in Figure 2-2, the expected taxi time for an arbitrary aircraft can be calculated by taking a weighted average of the expected taxi times out of each source node. The weights themselves can be proportional to the fraction of traffic supplied by each source node. By noting that each component of the taxi-out process is still memoryless (geometric, exponential, or Erlang), this expected taxi time can be used to estimate the departure rate from the airport. Analogous to Equation (3.2) we have,

$$R_{k,\mathbb{C}} = k \cdot \frac{1}{\sum_{i \in \mathbb{S}} p(i) \left(\sum_{l \in \mathcal{P}_i} (\eta_l + \frac{kX_l}{\mu_l}) \right)}. \quad (3.6)$$

In Equation (3.6), \mathbb{C} is the active runway configuration, \mathbb{S} is the list of source nodes in the network, $p(i)$ is the fraction of traffic provided by source node i , and \mathcal{P}_i is the taxi route from source node i . Using $R_{k,\mathbb{C}}$ as the estimate for the Poisson rate of departures from the airport, the transition probabilities can be calculated in an analogous fashion to Equation (3.5). While this is a more crude approximation than for the single link case, a comparison with simulated fractional state occupancy shows trends that are very similar to those in Figure 3-7. In this manner, the control algorithm is able to regulate the traffic level to the desired target value.

3.1.9 Shortcomings of the k_{ctrl} strategy

The k_{ctrl} algorithm described in this section has the advantage of being simple to understand and implement, but it has several inherent drawbacks. The most important challenge with this method is the selection of a target surface traffic level. The tuning procedure for this number is highly subjective, as described in Section 3.1.6. The value of k_{ctrl} is not only influenced by the layout of the airport, but also by the runway configuration being employed and other operational

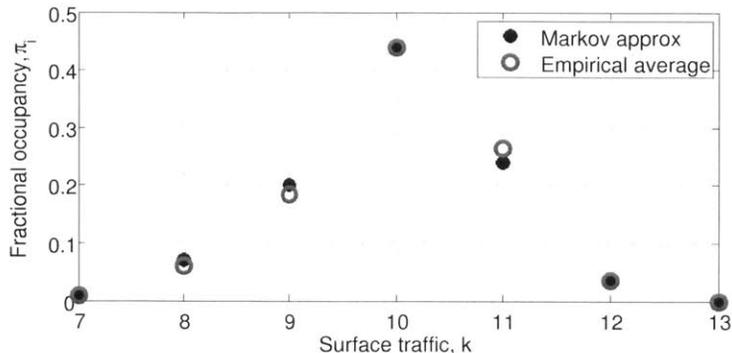


Figure 3-7: Comparison of Markov-approximated steady state occupancy and average simulated occupancy, for $k_{\text{ctrl}} = 10$.

considerations. For example, certain runways might have less space for the formation of departure queues than others, which limits the surface traffic level that can be accommodated.

Additional drawbacks of this method include the loss of all link-specific information available from the network model. For example, configuration- and airport-specific characteristics are embedded in the estimated link travel time parameters in the form of stopping probabilities and expected stop durations. Forcing a value of k_{ctrl} on the airport neutralizes the potential use of the information in these parameters. A third problem with the algorithm is the mismatch between the overall objective of implementing congestion control at airports, and the objective used for calculating the control policy. The overarching objective of airport congestion control is to reduce fuel consumption, while the k_{ctrl} algorithm merely tries to stabilize surface traffic levels. It would be far more desirable to employ a control strategy that directly targets fuel burn reduction. Finally, an operational constraint also limits the applicability of this algorithm at busy but space-constrained airports. This is the availability of gates at the airport. Since delaying departures at the gate increases average gate occupancy, it is possible for an arriving aircraft to be delayed because of the lack of empty gates. This problem is not directly addressed by the k_{ctrl} algorithm, and field demonstrations of similar strategies generally use ad-hoc techniques to circumvent this problem [61]. The issues discussed here are explained in greater detail towards the end of this chapter, where comparative simulations are carried out between this algorithm and the dynamic programming algorithm described in the next section.

3.2 Control based on dynamic programming

Based on the discussion in the previous section, it is desirable to develop a control algorithm that explicitly targets fuel burn reduction while limiting the adverse effect on airport performance. It should also adhere to operational constraints such as gate availability at the airport and user-defined levels of acceptable gate delay. For example, airline on-time reporting systems tag a flight as ‘late’ if the aircraft leaves the gate more than 15 min after its scheduled departure time. Exceeding this value of delay may be unacceptable for airlines as a result. Given the complex but information-rich nature of the taxi-out process, the constraints on the problem, and the multi-objective optimization requirement, a dynamic programming solution is the natural choice. This section describes the development of the control strategy in detail, starting from an illustrative example of control on a single link, and ending with the control strategy for the entire airport. As noted in Section 3.1.3, the highly simplified model for the departure process cannot be employed here. A more accurate estimate of the departure rate from the airport is derived below.

3.2.1 Markov process representation the taxi time model

Since the taxi time model presented in Section 2.2.3 is composed of memoryless random variables (Erlang, geometric and exponential), it can be depicted as a Markov process. Note the difference between the model presented in Section 3.1.7 and the one presented here. The former model was for transitions between surface traffic levels when the control inputs were already defined. The model being presented here is a representation of the basic taxi-out process. The model for a single link is shown in Figure 3-8, where the taxi time is the time taken to move from state 1 to the destination state D . States marked from 1 to n_l form the unimpeded portion of the taxi time, with a transition rate of λ_l each. Note that each state from 1 to $(n_l - 1)$ has only one outgoing transition. When the first $(n_l - 1)$ of the n_l hops that compose the unimpeded portion have been completed, there are two possible transitions. The system can enter the ‘stop’ state S with probability p_{kl} (and hence rate $\lambda_l p_{kl}$), or it can directly enter the destination state D with probability $(1 - p_{kl})$. State S has a geometric number of self-transitions with probability p_{kl} .

Now consider the problem of modeling the departure process from the airport for a moderately large traffic level k . This is equivalent to calculating the probability distribution of the number of departures from the airport in a given period of time, with observations starting at a random time.

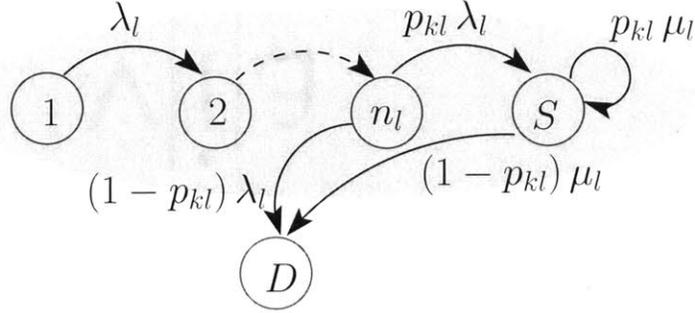


Figure 3-8: Representation of the taxi-out model as a Markov process, for a single link. The transition time for each state is exponential with rate λ_l or μ_l . State D is the final state, when the aircraft departs from the link.

The first step for calculating this quantity is finding the probability of the system being in state i ($i = \{1, 2, \dots, n_l, S\}$) when the observation starts. The solution procedure for Markov processes is outlined in [116], where it is shown that the probability of the system being in each state is proportional to the expected time spent in that state.

These expected occupancy times are depicted for the single link in Figure 3-9. There are n_l segments of length $\frac{1}{\lambda_l}$ (the unimpeded transitions), and one segment of length $\frac{p_{kl}}{\mu_l(1-p_{kl})}$ (the total expected stop time). Consider the following step-wise procedure for estimating the expected time to departure for an arbitrary aircraft already in the network, when the observation is also started at an arbitrary time.

1. The total expected taxi time from gate to runway is given by Equation (2.2) and is equal to $\left(\frac{n_l}{\lambda_l} + \frac{p_{kl}}{1-p_{kl}} \frac{1}{\mu_l}\right)$.
2. The expected time spent in each unimpeded transition (for example, from state $(n_l - 2)$ to $(n_l - 1)$) is $\frac{1}{\lambda_l}$.
3. From the two statements above, the probability of a random observer appearing when the process is in an unimpeded state $i \in \{1, 2, \dots, n_l\}$ is,

$$p_u = \frac{\frac{1}{\lambda_l}}{\frac{n_l}{\lambda_l} + \frac{p_{kl}}{1-p_{kl}} \frac{1}{\mu_l}}.$$

4. Using similar logic, the probability of the observer appearing when the process is in state S

is,

$$p_S = \frac{\frac{p_{kl}}{1-p_{kl}} \frac{1}{\mu_l}}{\frac{n_l}{\lambda_l} + \frac{p_{kl}}{1-p_{kl}} \frac{1}{\mu_l}}.$$

5. If the observer appears during an unimpeded transition out of state i , the expected time to go is,

$$t_{ui} = \frac{p_{kl}}{1-p_{kl}} \frac{1}{\mu_l} + \frac{n_l - i + 1}{\lambda_l}, \quad i \in \{1, 2, \dots, n_l\}.$$

6. If the observer appears in state S , the expected time to go (using the memoryless property) is,

$$t_S = \frac{p_{kl}}{1-p_{kl}} \frac{1}{\mu_l} + \frac{1}{\mu_l}.$$

7. The combined expected time-to-go t_{av} is the weighted sum of n_l unimpeded transition terms (each with probability p_u) and one stopped transition term (with probability p_S). The expression for t_{av} is thus,

$$t_{av} = \sum_{i=1}^{n_l} p_u t_{ui} + p_S t_S = p_u \sum_{i=1}^{n_l} t_{ui} + p_S t_S \quad (3.7)$$

Since this time corresponds to each aircraft, the net departure rate for k aircraft is $R'_{kl} = \frac{k}{t_{av}}$. The departure process from this link is approximately Poisson with rate R'_{kl} . While this is a more involved expression than Equation (3.2), it is a closer approximation to the actual rate. The departure process from the airport is composed of aircraft that are at different points on the surface and not necessarily just appearing at the source nodes. This means that the random observer assumption is more valid than the assumption in Section 3.1.3, which said that the expected remaining taxi time for each aircraft is equal to its total expected taxi time. A comparison of this theoretical model (Poisson with rate R'_{kl}) with Monte-Carlo simulations of transition probabilities is shown in Figure 3-10. The link parameters were $n_l = 20$, $\lambda_l = 0.4$, $\mu_l = 0.02$, $p_{ol} = 0.1$, and $X_l = 0.2$. Note that two types of parameters are being varied in the different plots: the starting traffic level as well as the assigned delay.

The Markov process representation for a two-link taxi path is shown in Figure 3-11. In this case, the taxi time is generated by going through each state in Link 1, with a probability $(1 - p_{k1})$ of skipping the state S_1 . The process is repeated for state 2, following which the destination state

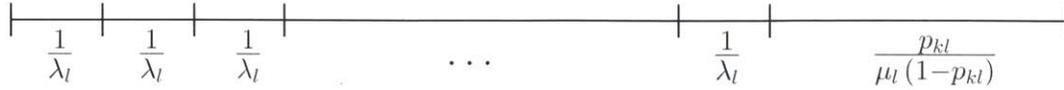


Figure 3-9: Expected length of each segment in the single link model.

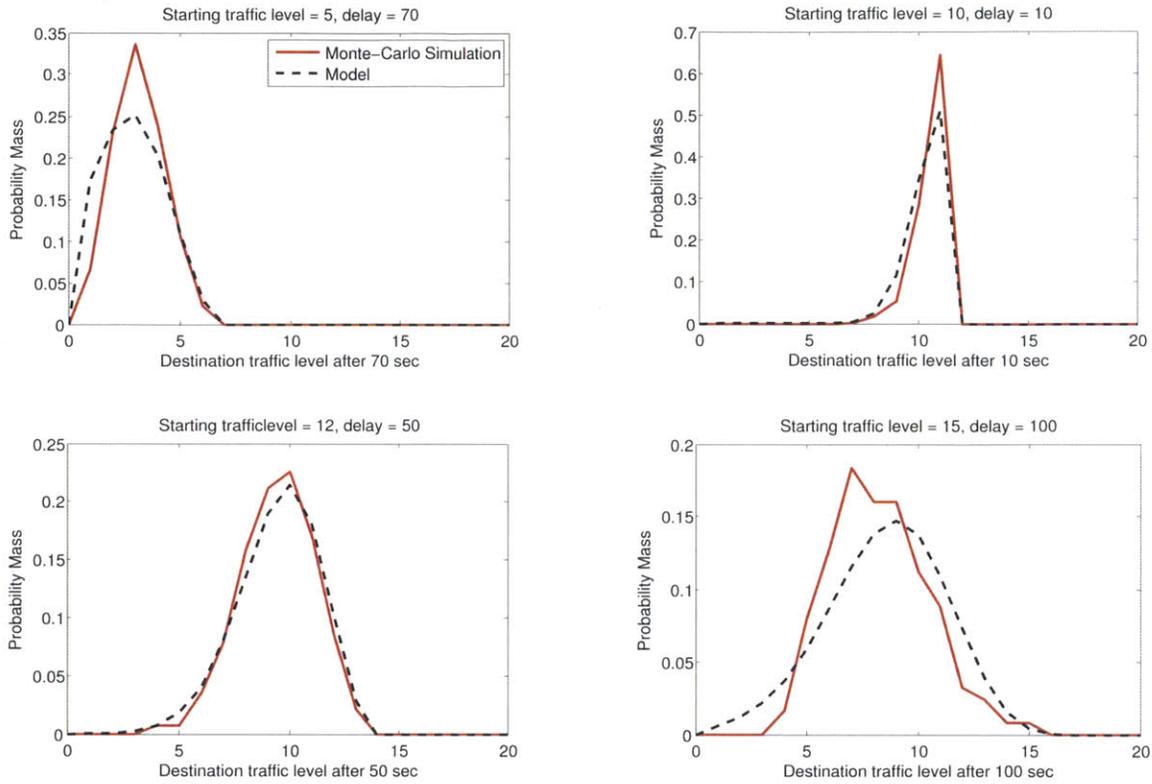


Figure 3-10: Comparison of theoretical transition probabilities and simulation-generated transition probabilities.

D is reached. The same model can be extended for multi-link paths, and the departure rate can be calculated using the same method as for the single-link case.

3.2.2 Derivation of maximum link throughput

Minimizing the effect on airport performance is one of the stated objectives of the dynamic programming formulation. Therefore, cost definition in the optimization problem should contain some measure of the network throughput for the airport. Before deriving the maximum throughput of the full network, it is necessary to find the maximum throughput of a single link l . Initially, let the link be capable of accommodating an infinite number of aircraft. Furthermore, let it be operating in

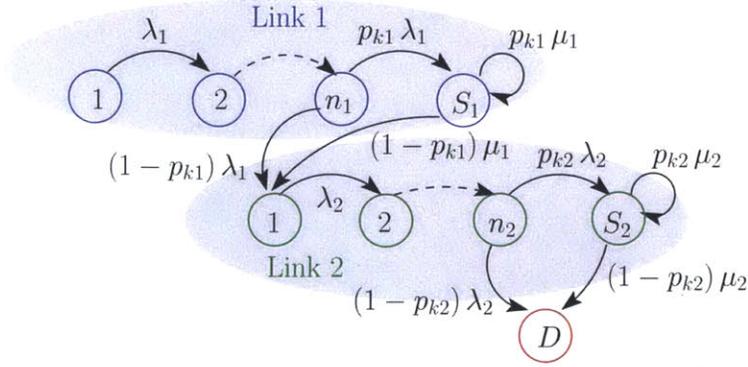


Figure 3-11: Representation of the taxi-out model as a Markov process, for a path with two links. The transition time for each state is exponential with rate λ_i or μ_i . State D is the final state, when the aircraft departs from the airport.

deterministic steady state, with the taxi time of each aircraft being equal to its expectation value. Aircraft are distributed regularly along its length, with each *departure from* the link occurring after a fixed time interval, and each *arrival to* the link happening at the same instant. If ν is the arrival rate to the link, under this assumption, a new arrival occurs every $(\frac{1}{\nu})$ sec. In steady state, the number of aircraft on the link will be maintained at some value k . From Equation (2.3), note that the (deterministic version of) taxi time on the link is given by

$$t_{kl} = \mathbb{E}[t_l|k] = \frac{n_l}{\lambda_l} + \frac{p_{0l}}{(1 - p_{0l})\mu_l} + \frac{kX_l}{\mu_l} = \eta_l + \frac{kX_l}{\mu_l}, \quad (3.8)$$

where η_l is a constant comprised of the first two terms. During this time interval in which an aircraft travels over the link, the k aircraft ahead of it depart from the link. Therefore, the inter-departure time Δt_{kl} is

$$\Delta t_{kl} = \frac{t_{kl}}{k} = \frac{\eta_l}{k} + \frac{X_l}{\mu_l}. \quad (3.9)$$

The steady state traffic level k is the result of equating the inter-arrival interval $(\frac{1}{\nu})$ to Δt_{kl} . Consequently, the minimum inter-arrival interval that can be sustained by the link is $(\frac{1}{\nu})_{\min} = \frac{X_l}{\mu_l}$, achieved as $k \rightarrow \infty$. The maximum sustained throughput of link l is the inverse of this value, that is,

$$\sigma_l \triangleq (\nu)_{\max} = \frac{\mu_l}{X_l}. \quad (3.10)$$

In the stochastic case, the average inter-departure times will be governed by the expectation in Equation (3.8). The expectation operator goes through the same derivation as for the deterministic case, which means that the result from Equation (3.10) holds for the stochastic case as well. Relaxing the assumption of infinite link capacity is also quite simple. Assuming the maximum capacity of the link to be k_{\max} , the minimum sustained inter-departure time as derived from Equation (3.9), will be $\frac{\eta_l}{k_{\max}} + \frac{X_l}{\mu_l}$. The maximum throughput will be the inverse of this quantity. If the input rate to a link l is less than σ_l , the link can be referred to as being *stable*. Once the maximum throughput values for each link are known, it is straightforward to derive the maximum network throughput, using the mincut/maxflow theorem [75, 76]. Note that in general, simply ensuring that each individual link remains stable does not ensure stability of the entire network [99, 100]. However, it is known that for the special case of directed acyclic graphs, such stability is guaranteed.

3.2.3 Derivation of maximum network throughput

The mincut/maxflow theorem [75, 76] states that the maximum capacity of a network is equal to the maximum flow rate through the most constrained cut across the network. A *cut* with respect to two terminals is defined to be “a set of branches such that when deleted from the network, the network falls into two or more unconnected parts with the two terminals in different parts” [76]. In a network containing a single link l , there is only one possible cut and the maximum throughput is thus trivially equal to σ_l .

Figure 3-12 (a) shows a network with two sources and a total of three links. As indicated in [77], multiple sources are handled by connecting a ‘supersource’ with infinite maximum link throughput to the two source nodes 1 and 2. In this case, this supersource is denoted by Σ and the extended network is depicted in Figure 3-12 (b). Any cut separates the network into two parts, one containing Σ and the other containing the sink node 4. Let the link capacities on $1 \rightarrow 3$, $2 \rightarrow 3$ and $3 \rightarrow 4$ be σ_{13} , σ_{23} and σ_{34} respectively. By assumption, the capacities of the links emanating from Σ are $\sigma_{\Sigma 1} = \sigma_{\Sigma 2} = \infty$. If aircraft are leaving source nodes 1 and 2 at rates ν_1 and ν_2 , the total flow rate restriction is given by,

$$\nu_1 + \nu_2 \leq \min(\sigma_{13} + \sigma_{23}, \sigma_{34}, \sigma_{13} + \sigma_{\Sigma 2}, \sigma_{23} + \sigma_{\Sigma 1}, \sigma_{23} + \sigma_{\Sigma 2}, \sigma_{13} + \sigma_{\Sigma 1}).$$

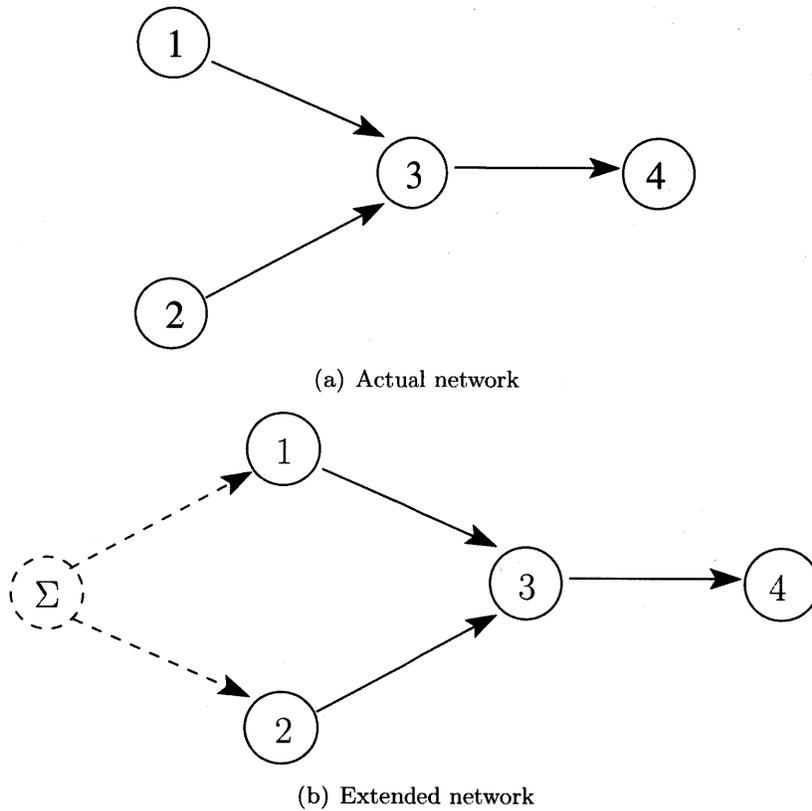


Figure 3-12: Network with two sources, three links.

Since the values of $\sigma_{\Sigma 1}$ and $\sigma_{\Sigma 2}$ are infinite, the first two terms are the only relevant ones inside the minimization operator. The constraint is thus reduced to,

$$\nu_1 + \nu_2 \leq \min(\sigma_{13} + \sigma_{23}, \sigma_{34}).$$

In addition to the total capacity of the network, there is also a limit on the maximum traffic that can be handled out of each source. Since the networks considered here are directed and acyclic, the maximum traffic rate from each source can be calculated independently of the others by looking at only the sub-network leading from one source to the sink. In cases where flow from one source can be routed through another source such as in Figure 2-2, the calculations are more involved but still

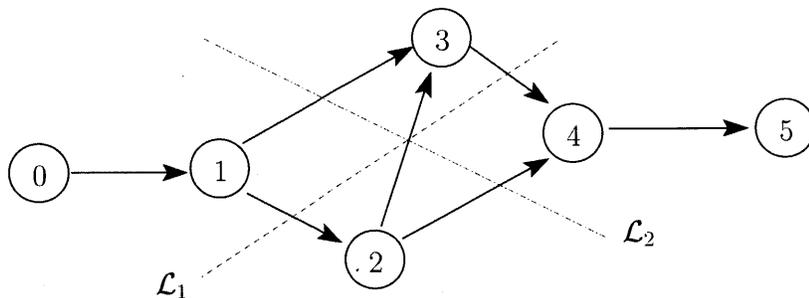


Figure 3-13: Network with one source, multiple links. All links in the cut \mathcal{L}_2 contain flows towards the destination, while there are only two such links for \mathcal{L}_1 . The maximum flow rate through this cut is equal to $(\sigma_{12} + \sigma_{34})$.

tractable. For the network shown in Figure 3-12, the additional constraints are,

$$\nu_1 \leq \min(\sigma_{13}, \sigma_{34}),$$

$$\nu_2 \leq \min(\sigma_{23}, \sigma_{34}).$$

In more complex networks such as the one in Figure 3-13, a maximum throughput of zero is assigned to branches that contain flows towards the source nodes [76]. All other branches are assigned their full throughput values. In Figure 3-13, it is assumed that the maximum throughput of link $1 \rightarrow 2$ is σ_{12} , and so on. The traffic rate generated by the source node 0 is assumed to be ν . The constraint on this value is given by,

$$\nu \leq \min(\sigma_{01}, \sigma_{12} + \sigma_{13}, \sigma_{12} + \sigma_{34}, \sigma_{13} + \sigma_{23} + \sigma_{24}, \sigma_{34} + \sigma_{24}, \sigma_{45}).$$

If the input rate to any link is less than its maximum throughput, the link is said to be in a *stable* condition. It should be remembered that the maximum throughput of link l is $\sigma_l = \frac{\mu_l}{X_l}$. This fact is used in Section 3.2.4 when defining stage costs for the optimal control problem. Furthermore, the network stability analysis may be easily extended to handle multiple sources [117]. The proposed implementation protocol in this work is to suggest pushback times to aircraft that are ready to depart from their gates. In essence, the control variables are the times of entry of aircraft into the network. Consequently, the control algorithm needs to assign a pushback time to each aircraft as it calls ready, while ensuring the stability of the network.

3.2.4 Cost definition

Any optimal control problem has two major components: the state definition and the cost definition. Assume, for the moment, that the state of the airport is represented by the quantity θ . The exact state definition depends on the formulation of the problem, and is discussed in Sections 3.2.5 and 3.2.6. The calculation of the optimal pushback policy for each state θ requires the knowledge of state transition probabilities and a definition of stage costs. The expected stage cost for the airport-aircraft system should account for three quantities: the fuel burn of each aircraft during taxi-out, the fuel burn of auxiliary motors while an aircraft is parked at the gate, and any loss of airport throughput because of the implementation of the control strategy. Note that the auxiliary motors are used to drive on-board aircraft systems such as radios and air conditioning when the engines themselves are not running. The cost is necessarily a function of the projected traffic level k_p at the assigned time of pushback, as defined by the current state θ_1 and the pushback delay u . It is shown in subsequent sections that k_p has a one-to-one correspondence with the state at the time of pushback (θ_2), and the transition probabilities from θ_1 to θ_2 (denoted by $p_{\theta_1\theta_2}(u)$) can be calculated using the method described in Section 3.2.1. With the above considerations, the expected stage cost \mathcal{C} for the system currently in state θ_1 on assigning a pushback delay u is encapsulated in Equation (3.11).

$$\mathcal{C}(\theta_1, u) = \sum_{\theta_2} p_{\theta_1\theta_2}(u) \left[c_1 k_p(\theta_2) + c_2 u + \frac{c_3}{k_p(\theta_2)} \right]. \quad (3.11)$$

As explained in Section 2.2.2, the expected taxi time increases linearly with the surface traffic level at the time of pushback (k_p). Thus the first term in Equation (3.11) is a measure of the aircraft taxi time, and therefore also a measure of the fuel burn during taxi out [108]. The second term captures the fuel cost of using auxiliary engines while at the gate. This cost is proportional to the pushback 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. This characteristic can be derived as follows. Since the total expected taxi time varies linearly with k_p ,

$$\mathbb{E} \left[\sum_{l \in \mathcal{P}} t_l \right] = \left(\sum_{l \in \mathcal{P}} \eta_l \right) + \left(\sum_{l \in \mathcal{P}} \frac{X_l}{\mu_l} \right) k_p,$$

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

$$\frac{\mathbb{E} [\sum_{l \in \mathcal{P}} t_l]}{k_p} = \left(\sum_{l \in \mathcal{P}} \eta_l \right) \frac{1}{k_p} + \left(\sum_{l \in \mathcal{P}} \frac{X_l}{\mu_l} \right). \quad (3.12)$$

Analogous to the derivation in Section 3.2.2, the minimum inter-departure time for a given path \mathcal{P} is achieved as $k_p \rightarrow \infty$. This value is equal to $\left(\sum_{l \in \mathcal{P}} \frac{X_l}{\mu_l} \right)$. The difference between this quantity and Equation (3.12) is a measure of the increase in taxi times because of the presence of only a finite number of aircraft on the surface. The quantity $\frac{1}{k_p}$ can thus be scaled using the constant c_3 to represent a time penalty. Since the first two terms in Equation (3.11) are also in units of time (properly scaled by the constants c_1 and c_2), the quantity $\frac{c_3}{k_p}$ can form the third term in the stage cost definition. Note that the stage cost increases exponentially as $k_p(u) \rightarrow 0$ and tends to drive any optimal policy away from states with very low traffic levels. This is a useful property, since very low traffic levels may lead to an undesirable event where the runway is unused even though there is high departure demand.

3.2.5 Control strategy for single links

It is logical to first develop the dynamic programming-based control strategy for a single link, and then to extend the scope to the entire airport network. Consider the setup depicted in Figure 3-14. Aircraft enter the network through the source node 1. This source is associated with a *buffer*, which corresponds to its gate capacity. In this figure, the buffer capacity is shown to be equal to 4 gates. Each gate in the buffer can be in one of three states: available (empty circle), occupied-inactive (filled square) and occupied-active (filled circle). The aircraft that is next in the queue for pushback is also occupied-active, but is denoted by a hashed circle. Available gates are empty, and can be occupied by arriving aircraft. When an aircraft pulls into a gate, it becomes inactive for a period of time, while it is being serviced. Once ready to leave, it calls the air traffic controller for permission to pushback. The gate is then tagged as being active. The purpose of the control formulation is to calculate optimal delays for aircraft that are active (ready for pushback). Aircraft are allowed to pushback on a first-come-first-served (FCFS) basis. That is, pushbacks 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.

Aircraft are assumed to arrive into the buffer at a Poisson rate β . The available buffer capacity \bar{N} is equal to the number of empty gates, and represents an implicit constraint on the delay assigned to the current aircraft by being a driver of the probability of buffer overflow. In this problem, a buffer overflow would occur if an arriving aircraft found no available empty gates ($\bar{N} = 0$). Let γ be the *buffer overflow tolerance*, or the maximum probability with which the capacity of the buffer is exceeded during the current assigned delay of duration u . This is equivalent to the probability of there being more than \bar{N} arrivals in the Poisson process with rate β during time u . An estimate of the arrival rate β can be used to determine the probability of this event as a function of the proposed delay u . In addition to the available buffer capacity \bar{N} , it has been shown in Section 3.2.4 that the surface traffic level k is an important driver of the optimal control strategy. Therefore, the state representation θ_s for the single-link network is given by $\theta_s = \{k, \bar{N}\}$.

Technically, the surface traffic level $k \in \mathbb{Z}_{\geq 0}$, the set of non-negative integers. However, it is assumed that $k \in \{0, 1, 2, \dots, k_{\max}\}$, where k_{\max} is a large but finite traffic level which is never expected to be exceeded in operation. The available buffer capacity \bar{N} is equal to the number of empty gates and is a finite, non-negative integer. If N_{\max} is the maximum gate capacity of the source, then $0 \leq \bar{N} \leq N_{\max}$. In addition, if N is the number of occupied gates (active and inactive), $N + \bar{N} = N_{\max}$ at all times. Since both the arrival and departure processes at the airport are stochastic, it is necessary to calculate the transition probabilities from one state to another, given the assigned pushback delay. The transitions between the buffer states \bar{N} are relatively straightforward, since they are governed by a Poisson process of rate β . Calculation of the transition probabilities between surface traffic levels is described in Section 3.2.1.

The calculation of optimal policies is a straightforward operation. A Bellman equation can be written for the infinite-horizon discounted cost problem using the states θ_s , and is given in Equation (3.13). \mathbb{U} is the set of allowable pushback delays, and is defined by the user. $J(\theta_s)$ is the optimal cost-to-go from state θ_s . Finally, α is the discount factor, which defines the weight on future costs with respect to the expected cost for the current transition. The infinite-horizon formulation is chosen as an approximation to the relative time scales of assigned pushback delays and airport demand variations. While pushback delays are of the order of a few minutes, demand variations at busy airports occur over a period of a few hours. The discounted-cost structure of the formulation provides numerical stability in the calculation of optimal policies [118]. In order to place sufficient

emphasis on future costs, the value of α can be set close to 1. In this work, $\alpha = 0.99$ unless otherwise stated. Equation (3.13) can be solved directly using matrix inversion, thus yielding the optimal policy $u(\theta_{s,1})$ and the optimal cost-to-go $J(\theta_{s,1})$.

$$J(\theta_{s,1}) = \min_{u \in \mathcal{U}} \left(\mathcal{C}(\theta_{s,1}, u) + \alpha \sum_{\theta_{s,2}} p_{\theta_{s,1}\theta_{s,2}}(u) J(\theta_{s,2}) \right) \quad (3.13)$$

Sample control policies can be calculated for the single-link case by defining values for the relevant parameters in the problem. While the parameter values for the full airport model described in Section 3.2.6 are derived from empirical data, the values in this illustrative case are arbitrary. It is assumed that the buffer capacity is $N_{\max} = 10$ aircraft. Thus up to 10 aircraft can be parked at any time. If an 11th aircraft arrives to the buffer, the first active aircraft needs to push immediately. With an overflow tolerance of $\gamma = 5\%$ and an arrival rate to the buffer of $\beta = 0.02$, the optimal policies are shown in Figure 3-15.

Note that the figure also shows the optimal policy calculated without consideration for the available buffer capacity. This is equivalent to the assumption that the available buffer capacity is $\bar{N} = \infty$. With respect to the state $\theta = \{k, \bar{N}\}$, this modification has the effect of flattening the cost structure along the \bar{N} dimension. The infinite-capacity dynamic programming control policy is thus a function of the same information that the k_{ctrl} algorithm used: the surface traffic level k only. Even when the buffer is nearly empty, the finite-capacity policy is more conservative than the infinite-capacity one. While this is not a direct comparison with the k_{ctrl} policy, it does show the advantage of being aware of the gate capacity constraint. Even when the buffer is nearly empty, the finite-capacity model is aware that assigning a large pushback delay to the current aircraft could take the system into a state with high buffer occupancy. This would necessitate faster pushbacks for subsequent aircraft, negating the fuel savings achieved by the current aircraft.

The effect of the buffer arrival rate on the control algorithm can be seen by comparing the policies in Figures 3-15 and 3-16. It can be seen that the policies in Figure 3-16 are much more conservative than those in Figure 3-15, as the buffer arrival rate has tripled to 0.06. Simulations carried out for this case show the advantage of using the finite buffer formulation over the infinite buffer formulation, as depicted in Figure 3-17. The finite-capacity policy only results in buffer overflow in one instance, while the infinite-capacity policy does so quite often.

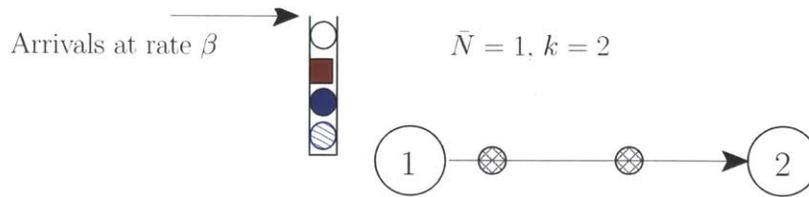


Figure 3-14: Model for a single link with finite gate capacity. Empty circles are empty gates, filled squares are occupied-inactive and filled circles are occupied-active. The circle with a one-sided hash pattern marks the aircraft that is next in line for pushback, while the double-hashed circles are aircraft that are already inside the network and are taxiing towards the runway at node 2.

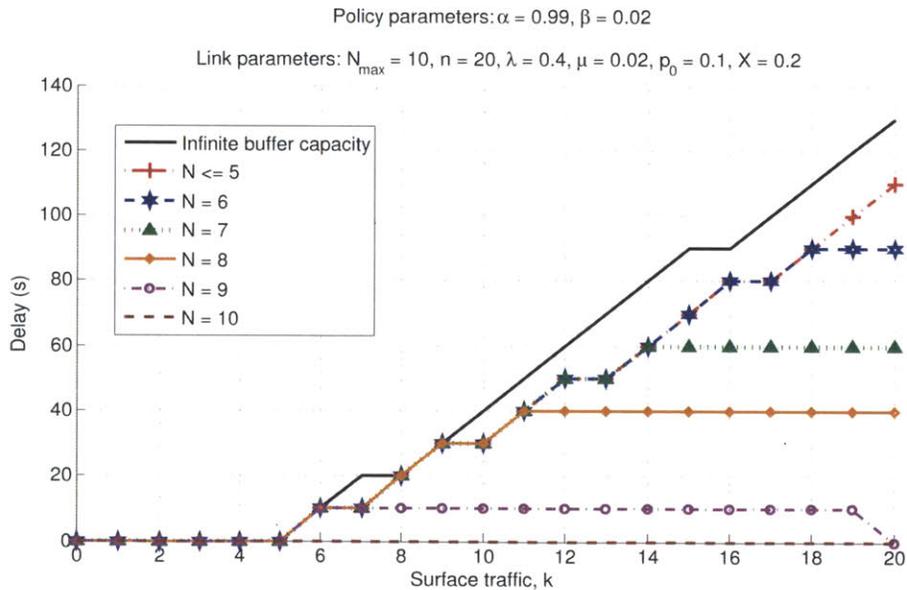


Figure 3-15: Comparison of optimal policies for a single link. The discount factor is $\alpha = 0.99$, and the arrival rate to the buffer is $\beta = 0.02$.

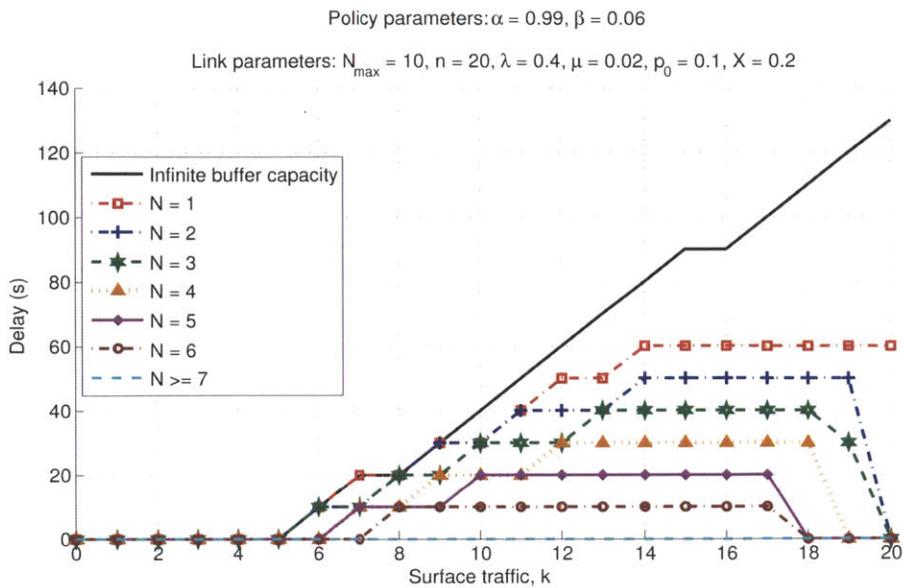


Figure 3-16: Comparison of optimal policies for a single link. The discount factor is $\alpha = 0.99$, and the arrival rate to the buffer is $\beta = 0.06$.

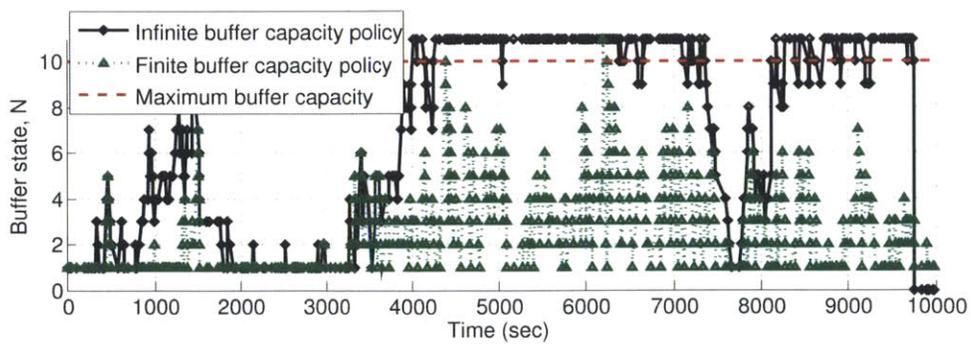


Figure 3-17: Simulation of the control strategy with and without consideration for the available buffer capacity. The discount factor is $\alpha = 0.99$, and the arrival rate to the buffer is $\beta = 0.06$.

3.2.6 Control strategy for the full airport model

The single-link control strategy described above can be extended to the full airport model with some modifications. Figure 3-18 shows the network model for departures from Runway 27, including the gate capacities available at each source. Note that the depicted gate capacities are only illustrative and are not quantitatively equivalent to the actual buffer sizes in the model. Figure 3-19 shows a functional representation of the network model, with solid arrows denoting the physical flow of aircraft and dashed arrows denoting flow of information. Since there are four sources in the network for Boston Logan International Airport (Figure 2-2), there are four buffers in the model. It is assumed that arriving aircraft have fixed destination buffers, but can occupy any gate within a given buffer. The status of each gate is denoted in the same way as in Figure 3-14: available (empty circle), occupied-inactive (filled square) and occupied-active (filled circle). Aircraft are allowed to pushback on a first-come-first-served (FCFS) order for the entire airport. That is, the order of pushback is independent of which buffer the aircraft are occupying a gate in. As before, the next aircraft to pushback is denoted by a hashed circle in Figures 3-18 and 3-19.

The arrival process to the airport as a whole is assumed to be Poisson with a rate β . This cumulative process is split into four Poisson processes with rate β_i , one for each buffer. The total rate β is estimated from historical data (average number of arrivals for the current time of day as calculated using ASDE-X data) in this chapter. A better estimate of the rate of aircraft arrivals to the airport can be obtained from the landing time predictions, and is described in Section 5.1. The individual buffer arrival rates β_i are assumed to be proportional to the gate capacities $N_{i,\max}$ at each terminal of the airport, with $\sum \beta_i = \beta$. For calculation of optimal policies, the buffer overflow tolerance γ is defined in an analogous fashion to the definition in Section 3.2.5. It is the probability with which the capacity of at least one buffer is exceeded during the current assigned gate delay u .

As before, the optimal control calculation (for pushback delay u) is a function of the state of the airport at the instant of calculation. The state in this case 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 modeled surface traffic level is capped at a large maximum value, with $k \in \{0, 1, 2, \dots, k_{\max}\}$. The control strategy is designed to operate at a level well below k_{\max} in all circumstances. The

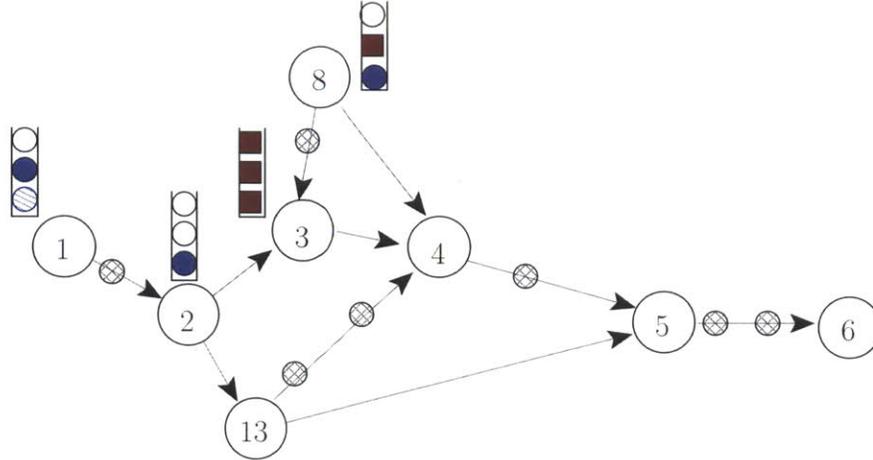


Figure 3-18: Network layout for departures from Runway 27. Note that there are few alternate taxi paths from each source. Empty circles are empty gates, filled squares are occupied-inactive and filled circles are occupied-active. The circle with a one-sided hash pattern marks the aircraft that is next in line for pushback, while the double-hashed circles are aircraft that are already inside the network and are taxiing towards the runway at node 6.

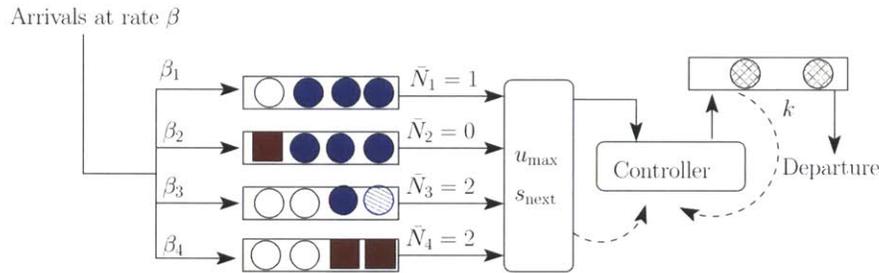


Figure 3-19: Illustration of the departure control algorithm for calculating optimal gate delays.

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 calculation of transition probabilities in this formulation is more involved than in the single-link case, but is helped by the independence of transitions between the different states. The probability of transition between the source states s_{next} is independent of delay u . If more than one aircraft is ready for pushback, the transition is essentially deterministic. This is because the next aircraft to be released is defined by the FCFS order and is already known to the control algorithm.

If there is no aircraft currently ready for departure, it is assumed that the probability mass on each source is proportional to its current gate occupancy. The transitions between the buffer states \bar{N}_i are relatively straightforward, since they are governed by Poisson processes of rate β_i . Calculation of the transition probabilities between surface traffic levels is more involved due to the aggregate nature of this part of the state. 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. Empirical estimates may be obtained using a Monte Carlo simulation procedure with randomized pushback policies. However, this is a highly time-intensive method. Approximate estimates of the transition probabilities for surface traffic levels are instead obtained using the method outlined in Section 3.2.1. Noting that each component of the full state $\phi = (k, \bar{N}_1, \dots, \bar{N}_{n_s}, s_{\text{next}})$ is independent of the others after conditioning on u , the full transition probability is equal to the product of the transition probabilities for each component.

The full state definition ϕ described above results in a very large number of states for a realistic airport. There are approximately 90 gates at Boston Logan, split into two terminals with a capacity of 25 gates each and two terminals 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 40 aircraft. Combined with four possible sources for each aircraft, the total number of states exceeds 40 million. Calculating optimal control policies for such a large number of states is prohibitively difficult. Therefore, the size of the problem is reduced by using state aggregation. Note that the set of buffer states \bar{N}_i encapsulates the risk of buffer overflow, as a function of the assigned pushback delay. An alternative parameterization of this risk is the maximum acceptable pushback delay assigned for a given instance of $\{\bar{N}_1, \bar{N}_2, \dots\}$. The state can now be re-defined to be $\theta = (k, u_{\text{max}}, s_{\text{next}})$, with u_{max} being the maximum pushback delay. The value of u_{max} is equal to that 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} can be 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 can be calculated by summing over all the transitions $\phi_1 \rightarrow \phi_2$ that correspond to transitions $\theta_1 \rightarrow \theta_2$. Combined with the stage cost definition developed in Sec. 3.2.4, it is possible to calculate the optimal pushback

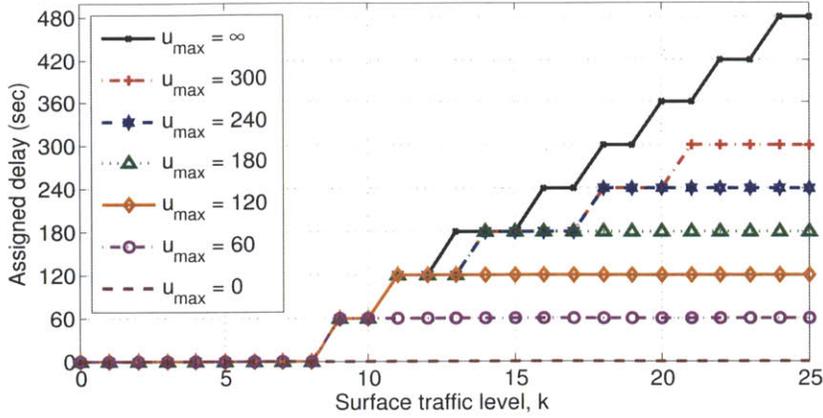


Figure 3-20: Optimal pushback delays at Boston Logan, calculated for departures from source 1. The total airport landing rate β is assumed to be 0.011, or 1 aircraft every 90 sec.

policy $u(\theta_1)$. A Bellman equation can be written for the infinite-horizon discounted cost problem using the aggregate states θ , as shown in Equation (3.14). $\mathbb{U}(\theta_1)$ is the set of available pushback delays for state θ_1 as constrained by $u_{\max}(\theta_1)$, $J(\theta_1)$ is the optimal cost-to-go from state θ_1 , and α is the discount factor which defines the weight on future costs with respect to the expected cost for the current transition.

$$J(\theta_1) = \min_{u \in \mathbb{U}(\theta_1)} \left(\mathcal{C}(\theta_1, u) + \alpha \sum_{\theta_2} p_{\theta_1 \theta_2}(u) J(\theta_2) \right) \quad (3.14)$$

The resultant optimal policies for aircraft leaving from source 1 are shown in Fig. 3-20. Each curve is for a different value of maximum pushback delay u_{\max} . As the level of surface traffic increases, the assigned pushback delay increases, up to the relevant value of maximum allowed delay. Similar policies can be calculated for the three remaining source nodes. Note that the number of states of type ϕ mapping into each of these aggregate curves changes, depending on the values assigned to c_1 , c_2 , c_3 , α and γ . However, once these quantities are defined, the control algorithm only needs to know the current surface traffic level and the gate occupancy to calculate the optimal pushback delay.

3.2.7 Comparison of k_{ctrl} and dynamic programming strategies

Figure 3-21 shows a comparison of the two methods for calculating control policies as described in this chapter. The chosen level of k_{ctrl} for the Boston Logan network is 20, while the dynamic programming strategy is the same as the one depicted in Figure 3-20. At low traffic levels ($k \leq 20$), the k_{ctrl} strategy assigns zero delay to all aircraft. For higher traffic levels, it assigns delays based on Equation (3.4). On the other hand, the dynamic programming strategy begins to assign small delays at lower traffic levels. The maximum delay assigned is upper-bounded, unlike in the k_{ctrl} case. The gradual increase in assigned delays also ensures smoother changes in surface traffic levels, as opposed to the abrupt ones in the k_{ctrl} case.

The resulting operational characteristics for both strategies are compared in Figure 3-22. The upper plot shows the traffic levels seen over the course of a simulation. It can be seen that the k_{ctrl} strategy, which is set to regulate the number of aircraft to 20, maintains a constant traffic level on the surface. However, the drawback of this strategy is seen in the lower right plot in Figure 3-22. With the delays assigned by the k_{ctrl} strategy, the buffer capacity is exceeded several times. In a real-world scenario, this would mean that a large fraction of arriving aircraft find all the gates occupied, and are thus delayed. The presence of these aircraft on the airport surface also creates operational complexity as they can block taxiways and alleyways required for the movement of departing aircraft.

The advantage of the dynamic programming based control strategy can be noted by comparing the traffic level and buffer occupancy shown in the lower left plot. It can be seen that when sufficient gate capacity is available, the traffic levels for the dynamic programming strategy are lower than those for the k_{ctrl} strategy. However when one or more of the buffers are close to full, the dynamic programming algorithm assigns smaller pushback delays, thus temporarily increasing the surface traffic level. This behavior can be noted towards the end of the simulation. In this way, a balance is struck between the fuel burn reduction obtained in times of low gate demand and the operational requirement of ensuring the availability of empty gates for arriving aircraft to occupy.

3.2.8 Adaptive algorithm for unmodeled parameter variation

It was mentioned in Section 2.2.2 that the effect of weather on the parameters in the taxi time model was difficult to quantify. The rollout algorithm described here can help with handling

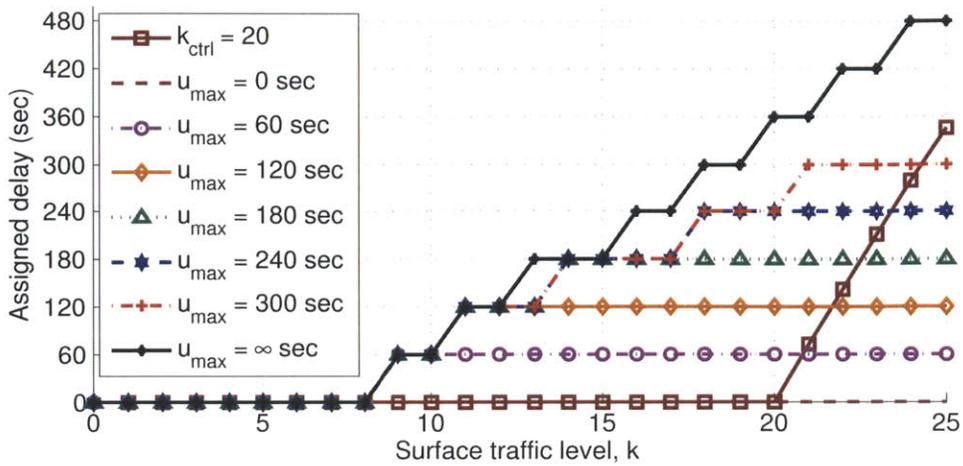


Figure 3-21: Comparison between the k_{ctrl} and dynamic programming based control strategies. It is seen that the k_{ctrl} assigns no delays for low traffic levels, and much higher delays compared to the dynamic programming strategy at high traffic levels.

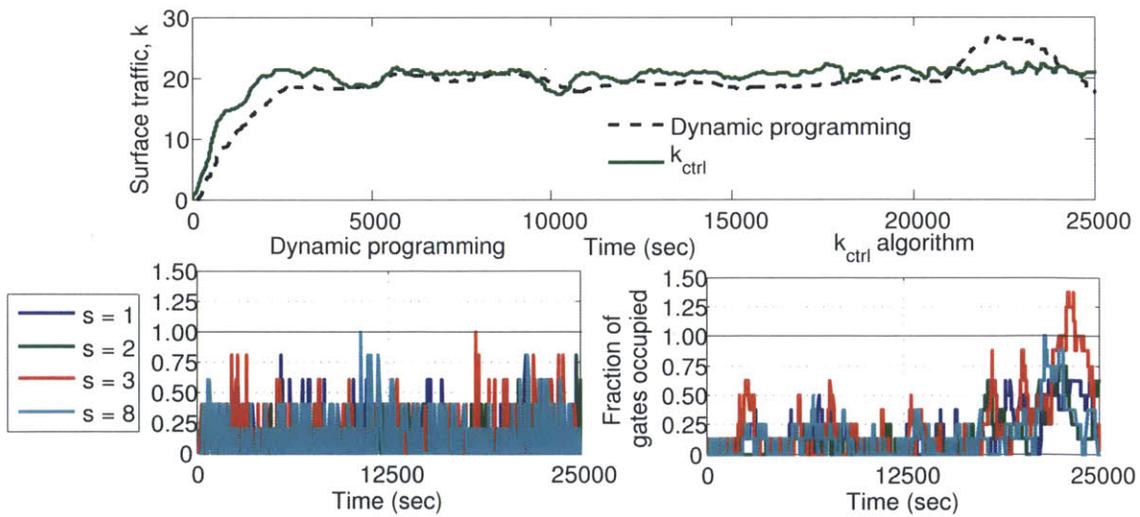


Figure 3-22: Simulation of the k_{ctrl} and dynamic programming based control strategies. The lower plots show the fraction of gates occupied at each airport terminal (source node).

the uncertainty introduced by this modeling simplification. Rollout is a method for calculating approximate control policies using dynamic programming [118]. The approximate cost-to-go is calculated offline in this method, while the current stage cost is estimated online. This allows a more accurate estimate of the current cost to be made, while the pre-calculated future costs allow for fast computation.

For the current problem, it is reasonable to assume that the effect of severe weather is not dependent on the surface traffic level. For illustrative purposes, consider only the single-link network model as described in Section 3.2.5. The expected taxi time on link l when the current traffic level is k can be calculated using Equations (2.2) and (2.5). Using the notation from Section 3.1.2, it is given by

$$\begin{aligned}\mathbb{E}[t_l|k] &= \frac{\eta_l}{\lambda_l} + \frac{p_{0l}}{1-p_{0l}} \frac{1}{\mu_l} + \frac{k X_l}{\mu_l} \\ &= \eta_l + \frac{k X_l}{\mu_l}.\end{aligned}$$

The portion of the expected taxi time independent of k is thus η_l . In the following description, it is assumed that the nominal value of η_l for link l is η_{0l} . Because of weather conditions on a given day, the actual value can vary around η_{0l} in a piecewise linear fashion. This behavior is a reasonable model for severe weather, which usually lasts for a few hours at a time. The current value of η_l can be estimated online, using past travel times over the link. The update equation for the current estimate $\hat{\eta}_l$ is based on the error between current expected taxi time and the actual observed taxi times, and is given by

$$\hat{\eta}_l^+ = \hat{\eta}_l^- + W \left(\hat{\eta}_l^- + k(r) \frac{X_l}{\mu_l} - t_{r,l} \right). \quad (3.15)$$

Here, the current estimate $\hat{\eta}_l^-$ is updated to a new value $\hat{\eta}_l^+$. The latest information is assumed to be available from the r^{th} aircraft to depart from the link, while $k(r)$ is the surface traffic level when it entered the network, $t_{r,l}$ is its actual taxi time, and W is a gain that weighs current observations against the past estimate of $\hat{\eta}_l^-$. Note that the last aircraft to depart from the link is not necessarily the last aircraft to have pushed back. The latest available estimate of η_l is used for calculating the pushback delay for each aircraft. The computation can be carried out by suitably varying the coefficient c_3 in Equation (3.11). This coefficient defines the weight on the loss of runway throughput, and Equation (3.12) shows that it is proportional to η_l . The online estimated current

stage cost is a modified version of Equation (3.11) and is given by,

$$\hat{C}(\theta_{s,1}, u) = \sum_{\theta_{s,2}} p_{\theta_{s,1}\theta_{s,2}}(u) \left[c_1 k_p(\theta_{s,2}) + c_2 u + \frac{c_3}{k_p(\theta_{s,2})} \frac{\hat{\eta}_l^+}{\eta_{0l}} \right].$$

The factor $\frac{\hat{\eta}_l^+}{\eta_{0l}}$ increases the cost of throughput loss when aircraft are experiencing longer taxi times than the average. Instead of solving the full Bellman equation for calculating the exact optimal policy, the online control policy can be calculated by assuming that the cost-to-go $J(\theta_{s,2})$ in Equation (3.13) is unchanged. The online minimization procedure is then a direct algebraic comparison of the right hand side of Equation (3.13) evaluated at each feasible control value u in the set \mathbb{U} . It should be noted that the tuning of the weight W is highly subjective in this algorithm. There are also several more update functions possible instead of the one proposed in Equation (3.15). The theory of adaptive control is well suited to the consideration of these functions, but is beyond the scope of this thesis.

Chapter 4

Control of Arrival Operations

Safety and efficiency are critical to the working of the air transportation system in general, but airborne flight management is much more safety-critical than airport surface operations. The airspace today is divided into sectors, each of which has an air traffic controller who is responsible for managing the traffic within it. Each aircraft receives specific instructions from the controller, and is expected to follow these instructions as closely as possible. The compartmentalization of airspace ensures the safe transit of all aircraft without putting excessive strain on the controllers, but frequently results in congestion near the airport. This problem is caused by the lack of coordination imposed by the division of airspace: since a controller cannot know the traffic situation in neighboring sectors, the handoffs between sectors are essentially uncoordinated. In addition, the sectors with the least available volume of airspace are the ones surrounding major airports. These sectors thus handle the most traffic. This means that aircraft arrive almost unimpeded through the outer airspace into the vicinity of the airport, but then spend a large amount of time orbiting in holding patterns while waiting for landing clearance. A high number of aircraft in a relatively small volume of airspace is a safety hazard, and also results in high fuel consumption because the aircraft have to fly for longer times at low altitudes. Thus there is a significant performance loss associated with current protocols.

Fully centralized management of arrival operations is desirable from an efficiency point of view. An central facility with access to global information can calculate optimal 4-D trajectories for all aircraft, eliminating the need for conflict detection and resolution. While this scenario is rightly recognized to be unrealistic, it nevertheless denotes one end of the spectrum of possible control

strategies. The other end of this spectrum is fully distributed control or free flight, as discussed in Section 1.2.2. A realistic goal is to find a compromise between these two strategies, with distributed control in sparsely traveled airspace and centralized control in congested airspace. In the airport arrivals context, this means that aircraft are controlled by a central facility close to the airport, and carry out their own conflict detection and resolution prior to entering this region of centralized control. This chapter focuses on developing such a control strategy, with an explicit recognition of the communications hardware and algorithms required for its implementation. It is shown that a well-chosen boundary between the regions of centralized control and distributed control allows the system to achieve efficiency close to the fully centralized case.

4.1 Characterization of arrival airspace

The current air traffic control architecture relies on ground-based radars to provide centralized surveillance. Ground radars are very large structures that are expensive to deploy and need a significant amount of maintenance [119]. Furthermore, these systems are subject to terrain blockage, and cannot provide coverage in areas where there is no line-of-sight. Instead of relying on expensive ground radar technologies, NextGen aircraft will have enhanced onboard sensing capabilities, and will carry wireless communication platforms [107, 120]. Wireless communication systems can operate beyond the line-of-sight constraints of radar, enabling longer-range aircraft detection.

4.1.1 Automatic Dependent Surveillance - Broadcast

Automatic Dependent Surveillance-Broadcast (ADS-B) is a NextGen surveillance and communication technology in which aircraft broadcast on-board flight information via datalink to ground stations or equipped aircraft within range [107, 120]. The position and velocity data is obtained using on-board satellite navigation systems. Two different types of services are offered by this technology, called ADS-B Out and ADS-B In. ADS-B Out refers to the regular broadcasts of current aircraft states made by the transponder on-board each aircraft (soon to be mandatory), while ADS-B In refers to an optional receiver which can detect broadcasts from other aircraft or ground stations.

Since ADS-B uses the same bandwidth as the replies to Secondary Surveillance Radars (SSRs), high aircraft and SSR density near airports can potentially degrade the performance of both sys-

tems. The communications algorithm used for the simulations in this chapter is taken from prior collaborative work [42], and tunes the power of ADS-B transmissions in order to minimize interference with other ADS-B transponders and ground radars. Successful implementation of control in the airspace requires message reception from one aircraft to another within a certain maximum latency period (Section 4.2.6), and thus defines the constraints for the communications layer. For the purposes of this thesis, it is sufficient to note that the utilized ADS-B transmission power is high in the distributed control region in order to maximize range, while it is low in the centralized region to minimize interference. Throughout this thesis, it is assumed that ADS-B transmissions are of the broadcast type only. The control algorithms do not require any two-way communication between aircraft. Instead, the distributed control algorithm ensures that computations carried out independently on-board different aircraft are consistent with respect to each other. A detailed description of this issue is given in Section 4.2.4.

4.1.2 Proposed system architecture

The proposed two-dimensional system architecture is depicted in Figure 4-1. While the airspace in reality is three dimensional, the concept of merging streams of air traffic remains invariant. The geometric treatment in this chapter is equally valid for the three-dimensional case, but is much easier to illustrate in two dimensions. The system is composed of an inner region of centralized control where aircraft are under the direct authority of the airport, and a distributed control region for aircraft farther out. The centralized zone is assumed to be circular, but can be of any shape without affecting the development of the control strategy. The double-hashed circles are aircraft approaching the airport at the center of the centralized zone. Aircraft C1 to C5 receive instructions from the airport, while aircraft D1 to D10 carry out their own conflict detection and resolution. The solid lines in the figure denote standard arrival procedures, which are published paths followed by all arriving aircraft. Designated intersections of two or more arrival paths in the airspace are known as fixes, while the straight-line paths between two fixes are called links. Minimal procedural modifications are desirable for easy implementation and large-scale deployment of new control algorithms, and hence the structure of the airspace is assumed to be the same as it is at present.

The development of the control strategy assumes that all aircraft are equipped with ADS-B Out as well as ADS-B In. This is a reasonable assumption to make, considering the mandate for

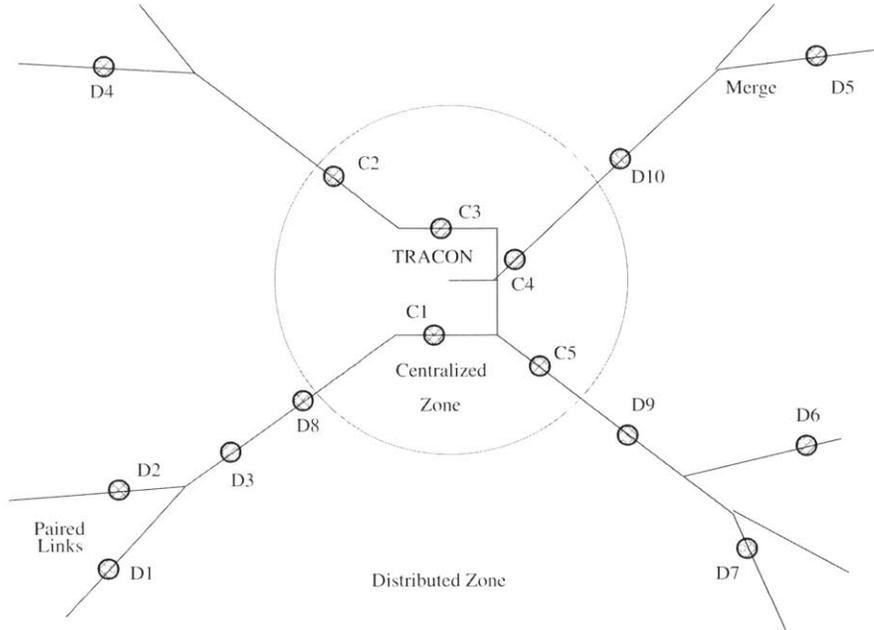


Figure 4-1: Overview of the model for arrival airspace. The airport is at the center of the figure. The distributed algorithm is in effect outside the circular region around the airport. Aircraft within the circular region receive commands from the central facility (airport).

all aircraft to be equipped with ADS-B Out by 2020 [60]. The case of mixed equipage in terms of ADS-B In is addressed in Section 4.2.7. In the ADS-B standard, each aircraft transmits position and velocity messages at the rate of 2 messages per second using a simple random access mechanism [107]. The transmission power is variable, and is tuned in order to minimize interference with other ADS-B transponders and ground radars using the method described in [42]. The centralized control algorithm calculates optimal velocities for all aircraft based on their last known state information.

In the distributed zone, each aircraft uses an adaptive channel access algorithm [42] to minimize the State Update Interval (SUI), that is, the time between two successful state vector reports. An upper bound on the maximum allowable SUI based on simple kinematic relations is derived in Section 4.2.6.

4.2 Development of control strategy for arrivals

This section develops a control algorithm that minimizes the flight times of aircraft from the time of appearance at the periphery of the airspace around an airport, to their eventual landing at

the airport. The primary control variable in this formulation is a change in aircraft velocity. A minimum separation requirement between each pair of aircraft is imposed for safety. Trajectory modifications (holding patterns) are avoided as far as possible in order to maximize safety [121]. 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 proposed control algorithm can either be automatically implemented by the aircraft involved in a potential conflict, or provide conflict resolution advisories to the pilot and the controllers.

4.2.1 Geometry of engagement between a pair of aircraft

The relative geometry between a given pair of aircraft depends on the links that they currently occupy. Broadly, any two links in the illustrative network shown in Figure 4-1 can be classified as being *paired* or *unpaired*. Two links are said to be paired if they lead to the same fix, otherwise they are said to be unpaired. This distinction is important when considering the separation requirement between aircraft. If two aircraft are on paired links, the point of closest approach between them may occur before the merge point.

The distance of closest approach is based on the relative velocity and initial positions of the two aircraft. The geometry of such a scenario is depicted in Figure 4-2, which shows a top-down view of two aircraft A and B approaching a merge point. Aircraft A is moving towards the merge point with velocity \bar{v}_A and aircraft B is doing so with velocity \bar{v}_B . Without loss of generality, let aircraft A be reaching the merge point earlier than aircraft B. The angle between the two links is δ . The initial relative position of aircraft A with respect to B in the horizontal plane is (x_0, y_0) , with \bar{r}_0 denoting the vector $x_0 \hat{i} + y_0 \hat{j}$. The velocity vectors are given by $\bar{v}_A = v_A \cos \delta \hat{i} - v_A \sin \delta \hat{j}$ and $\bar{v}_B = v_B \hat{i}$. Let the relative velocity be given by $\bar{v}_r = \bar{v}_A - \bar{v}_B$, and let the angle between \bar{r}_0 and \bar{v}_r be δ_r . Note that δ_r decreases with an increase in v_A , but increases with an increase in v_B . Finally, let \bar{r} be the relative position of A with respect to B after time t . Then the evolution of \bar{r} is given by,

$$\bar{r} = \bar{r}_0 + \bar{v}_r t. \quad (4.1)$$

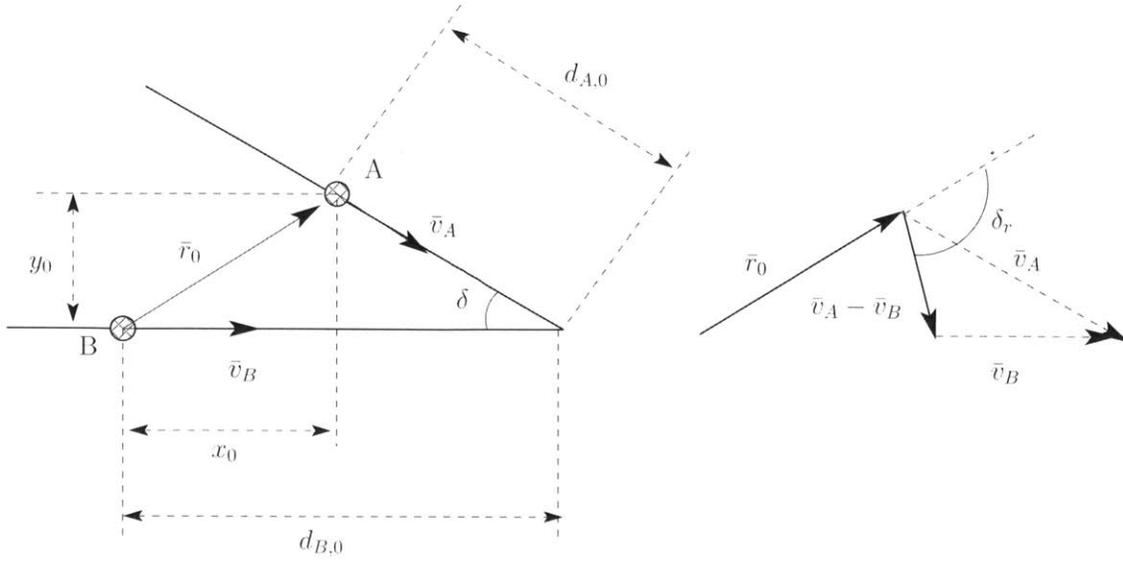


Figure 4-2: Geometry for calculating the distance of closest approach (Plan view).

The modulus of \bar{r} is given by,

$$\begin{aligned} |\bar{r}|^2 &= (\bar{r}_0 + \bar{v}_r t) \cdot (\bar{r}_0 + \bar{v}_r t) \\ &= \bar{r}_0 \cdot \bar{r}_0 + 2\bar{r}_0 \cdot \bar{v}_r t + \bar{v}_r \cdot \bar{v}_r t^2. \end{aligned}$$

Taking the derivative with respect to time and equating the result to zero, the time of closest approach t_c is given by,

$$t_c = - \left(\frac{\bar{r}_0 \cdot \bar{v}_r}{\bar{v}_r \cdot \bar{v}_r} \right). \quad (4.2)$$

If this value is less than the time at which aircraft A is projected to reach the merge point, the point of closest approach is before the merge point. Note that the value of t_c may be negative if the geometry dictates that the point of closest approach has already been crossed, and the distance between A and B is monotonically increasing. This happens when $\delta_r < \frac{\pi}{2}$, since it means that $\cos \delta_r > 0$. For the acute-angled case, the point of closest approach has already passed. Combining Equations (4.1) and (4.2), the relative position of A with respect to B at the point of closest approach is given by,

$$\bar{r}_c = \bar{r}_0 - \bar{v}_r \left(\frac{\bar{r}_0 \cdot \bar{v}_r}{\bar{v}_r \cdot \bar{v}_r} \right).$$

Taking the dot product of \bar{r}_c with itself gives the magnitude of the distance of closest approach.

$$\begin{aligned}
r_c^2 &= \left[\bar{r}_0 - \bar{v}_r \left(\frac{\bar{r}_0 \cdot \bar{v}_r}{\bar{v}_r \cdot \bar{v}_r} \right) \right] \cdot \left[\bar{r}_0 - \bar{v}_r \left(\frac{\bar{r}_0 \cdot \bar{v}_r}{\bar{v}_r \cdot \bar{v}_r} \right) \right] \\
&= r_0^2 + \left(\frac{\bar{r}_0 \cdot \bar{v}_r}{\bar{v}_r \cdot \bar{v}_r} \right)^2 \bar{v}_r \cdot \bar{v}_r - 2 \left(\frac{\bar{r}_0 \cdot \bar{v}_r}{\bar{v}_r \cdot \bar{v}_r} \right) \bar{r}_0 \cdot \bar{v}_r \\
&= r_0^2 + \frac{r_0^2 v_r^4 \cos^2 \delta_r}{v_r^4} - \frac{2 r_0^2 v_r^2 \cos^2 \delta_r}{v_r^2} \\
&= r_0^2 \sin^2 \delta_r \\
\Rightarrow r_c &= r_0 \sin \delta_r. \tag{4.3}
\end{aligned}$$

Evidently, the distance of closest approach is governed by the initial separation between the aircraft and the angle between their velocity vectors. Note that there are two cases that result in the same value of r_c in Equation (4.3), one with $\delta_r > \frac{\pi}{2}$ and the other with $\delta_r < \frac{\pi}{2}$. As discussed above, only the obtuse-angled case is interesting since it results in a future point of closest approach. Let the required minimum separation between any pair of aircraft in the airspace be s_{\min} . This generates a geometric constraint on the minimum value of $\sin \delta_r$. Since the sinusoid is a decreasing function for obtuse values of δ_r , Equation (4.4) imposes a constraint on the maximum value of δ_r .

$$r_0^2 \sin^2 \delta_r \geq s_{\min}^2 \quad \Rightarrow \quad \sin \delta_r \geq \frac{s_{\min}}{r_0}. \tag{4.4}$$

Note that the distance at initial approach should be more than s_{\min} for this relation to be relevant. Also note that \bar{v}_r is an invariant vector, while the relative position vector slowly becomes horizontal as A approaches the merge point ahead of B. Therefore, the value of δ_r decreases monotonically after initial contact. The point of closest approach occurs when $\delta_r = \frac{\pi}{2}$. If the initial value of δ_r is less than $\frac{\pi}{2}$, the distance between A and B increases monotonically. Finally, v_B should be as large as feasible while still maintaining separation, in order to minimize the time taken by the combination of A and B to clear the merge point. Since δ_r increases monotonically with v_B , this means that v_B is maximized when δ_r satisfies the constraint in Equation (4.4) with equality. Note that the relation is satisfied by two values of δ_r , of which the obtuse angle is the optimum point. The constraint in Equation (4.4) is not active when the initial δ_r is less than $\frac{\pi}{2}$, and also when the point of closest approach is projected to be beyond the merge point. In this case, aircraft A will have already turned away from B along the common link leading out of the merge point. In such cases, the relevant constraint is that of maintaining a minimum distance of s_{\min} at the merge point.

4.2.2 Optimal velocities for a pair of aircraft

Consider the setup shown in Figure 4-2. Aircraft A is initially at a relative position (x_0, y_0) with respect to B. It is also a distance $d_{A,0}$ from the merge point, while B is at a distance $d_{B,0}$ from the merge point. Let the two aircraft be traveling at velocities $v_{A,0}$ and $v_{B,0}$ respectively. An optimization problem can be set up to calculate the final commanded velocities $v_{A,f}$ and $v_{B,f}$ that minimize the time at which both aircraft clear the merge point. This derivation deals only with calculating the system-optimal velocities. The methods by which the centralized and distributed control algorithms arrive at these values in real-time are described in Sections 4.2.5 and 4.2.4.

Assuming as before that B is the trailing aircraft within this pair, the time at which both aircraft clear the merge point is given by the time at which B reaches the merge point, $\frac{d_{B,0}}{v_{B,f}}$. This is the objective function for the problem. The order of precedence between A and B (calculated by separate means) can be imposed by constraining $v_{A,f}$ to be at least as large as $v_{A,0}$. The flight envelope of the two aircraft can be captured by the maximum and minimum velocities that they can fly at. Note that the velocity of A is already lower-bounded. Therefore, the only relevant physical constraints are set by A's maximum velocity $v_{A,\max}$ and B's minimum and maximum velocities $v_{B,\min}$ and $v_{B,\max}$. The final constraint is imposed by the minimum separation requirement between A and B. This can either take the form of Equation (4.4), or be equal to s_{\min} . In general, it can be represented by an upper bound on $v_{B,f}$ as a function $v_{A,f}$, $d_{A,0}$ and $d_{B,0}$. Note that the function f increases monotonically with $v_{A,f}$, as explained in Section 4.2.1.

$$\begin{aligned}
 & \min_{v_{A,f}, v_{B,f}} \frac{d_{B,0}}{v_{B,f}} & (4.5) \\
 & \text{s.t. } v_{A,f} \geq v_{A,0}, \quad v_{B,f} \leq v_{B,\max} \\
 & \quad v_{A,f} \leq v_{A,\max}, \quad v_{B,f} \geq v_{B,\min} & \text{(Feasibility)} \\
 & \quad v_{B,f} \leq f(v_{A,f}, d_{A,0}, d_{B,0}). & \text{(Separation)}
 \end{aligned}$$

We can now set up the Lagrangian for the formulation in Equation (4.5), and find the Karush-Kuhn-Tucker necessary conditions for optimality [122, 123]. Letting ψ_i be the Lagrange coefficients

and τ_i be the slack in the inequalities,

$$\begin{aligned}\mathcal{L} = & \frac{d_{B,0}}{v_{B,f}} + \psi_1 (v_{A,f} - v_{A,0} - \tau_1^2) + \psi_2 (v_{A,f} - v_{A,\max} + \tau_2^2) + \psi_3 (v_{B,f} - v_{B,\max} + \tau_3^2) \\ & + \psi_4 (v_{B,f} - v_{B,\min} - \tau_4^2) + \psi_5 (v_{B,f} - f(v_{A,f}, d_{A,0}, d_{B,0}) + \tau_5^2).\end{aligned}$$

Finding the gradient of \mathcal{L} and setting it equal to zero gives the necessary conditions for optimality.

$$\frac{d\mathcal{L}}{dv_{A,f}} = \psi_1 + \psi_2 - \psi_5 \frac{df}{dv_{A,f}} = 0 \quad (4.6)$$

$$\frac{d\mathcal{L}}{dv_{B,f}} = -\frac{d_{B,0}}{v_{B,f}^2} + \psi_3 + \psi_4 + \psi_5 = 0. \quad (4.7)$$

The derivatives with respect to ψ_i give back the inequality constraints.

$$v_{A,f} - v_{A,0} - \tau_1^2 = 0, \quad (4.8)$$

$$v_{A,f} - v_{A,\max} + \tau_2^2 = 0, \quad (4.9)$$

$$v_{B,f} - v_{B,\max} + \tau_3^2 = 0, \quad (4.10)$$

$$v_{B,f} - v_{B,\min} - \tau_4^2 = 0, \quad (4.11)$$

$$v_{B,f} - f(v_{A,f}, d_{A,0}, d_{B,0}) + \tau_5^2 = 0 \quad (4.12)$$

The complementary slack conditions are produced by taking derivatives with respect to τ_i .

$$\psi_1 \tau_1 = \psi_2 \tau_2 = \psi_3 \tau_3 = \psi_4 \tau_4 = \psi_5 \tau_5 = 0.$$

While there are several potential solutions to this system of equations, it so happens that feasible solutions are obtained in only two cases. These cases are enumerated below.

Case 1: $\psi_5 = 0$

Assuming that $\psi_5 = 0$ allows τ_5 to be non-zero, according to the complementary slack conditions. This means that the minimum separation constraint imposed by $f(v_{A,f}, d_{A,0}, d_{B,0})$ is not active. The following description shows that this is only optimal when aircraft B is so far behind aircraft A that it cannot cause a conflict even by flying at its maximum velocity. Notice that combining

$\psi_5 = 0$ and Equation (4.6) results in the requirement $\psi_1 + \psi_2 = 0$. However, Equations (4.8) and (4.9) combined with the complementary slack equation imply that ψ_1 and ψ_2 cannot be non-zero at the same time (since it is not possible to have $v_{A,f} = v_{A,0} = v_{A,\max}$ in general). Therefore, $\psi_1 = \psi_2 = 0$ which means that $v_{A,f}$ can take any value between $v_{A,0}$ and $v_{A,\max}$. Similarly putting $\psi_5 = 0$ in Equation (4.7) results in,

$$\psi_3 + \psi_4 = \frac{d_{B,0}}{v_{B,f}^2}.$$

Since ψ_3 and ψ_4 also cannot be non-zero at the same as implied by Equations (4.10) and (4.11), one of them must be zero and the other equal to $\frac{d_{B,0}}{v_{B,f}^2}$. With the complementary slack equations, this implies that $v_{B,f}$ is equal to either $v_{B,\min}$ or $v_{B,\max}$. Noting that the Karush-Kuhn-Tucker conditions are valid at all extrema, a simple value substitution shows that $v_{B,f} = v_{B,\max}$ minimizes \mathcal{L} while the other value maximizes it. In summary, when the separation between A and B is large enough so as to not result in a conflict under any feasible velocities, it is optimal for B to fly at its maximum velocity with A flying at any feasible velocity.

Case 2: $\psi_5 \neq 0$

Similar arguments as before can be used to find the optimal solution when $\psi_5 \neq 0$. Note that this implies that the geometric constraint on $v_{B,f}$ is active. At the same time, Equation (4.6) implies that the sum of ψ_1 and ψ_2 is non-zero. Since they cannot both be non-zero, either ψ_1 or ψ_2 must be equal to zero. Consequently, one of τ_1 and τ_2 must be equal to zero and thus $v_{A,f} = v_{A,0}$ or $v_{A,f} = v_{A,\max}$. Since $f(v_{A,f}, d_{A,0}, d_{B,0})$ is known to be monotonically increasing with $v_{A,f}$, the solution that minimizes the Lagrangian is obviously given by $v_{A,f} = v_{A,\max}$ with $v_{B,f}$ satisfying Equation (4.12) without any slack. Thus when A and B are expected to reach the merge point close to each other, the optimal solution is for A to fly as fast as possible while B satisfies the minimum separation constraint with equality.

4.2.3 Optimal velocities for multiple merging aircraft

The optimal velocity calculations described in Section 4.2.2 can be extended seamlessly to the multiple aircraft case. The first extension is to the case of three aircraft. Two sample scenarios are depicted in Figure 4-3 for the types of engagements possible in this case. Note that the order of precedence is A-B-C in Figure 4-3 (a), while it is A-C-B in Figure 4-3 (b). However, once this

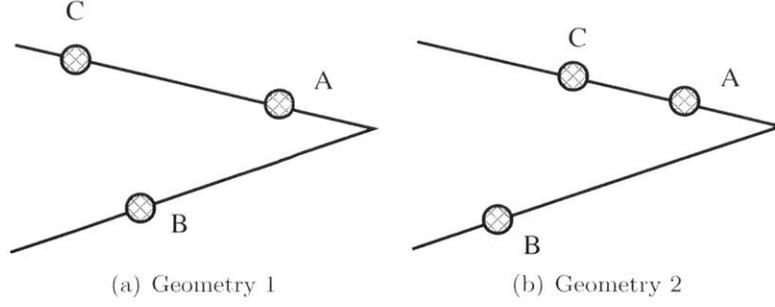


Figure 4-3: Possible geometries for multiple merging aircraft.

order is defined by the method described in Section 4.2.4, the calculation of optimal velocities is straightforward. Without loss of generality, let the order be given by A-B-C. There is no restriction on which of the two links is occupied by any of the three aircraft. The minimum separation requirement is imposed on B with respect to A, and on C with respect to both A and B. The Lagrangian in this case is given by,

$$\begin{aligned}
\mathcal{L} = & \frac{d_{C,0}}{v_{C,f}} + \psi_1 (v_{A,f} - v_{A,0} - \tau_1^2) + \psi_2 (v_{A,f} - v_{A,\max} + \tau_2^2) + \psi_3 (v_{B,f} - v_{B,\max} + \tau_3^2) \\
& + \psi_4 (v_{B,f} - v_{B,\min} - \tau_4^2) + \psi_5 (v_{C,f} - v_{C,\max} + \tau_5^2) + \psi_6 (v_{C,f} - v_{C,\min} - \tau_6^2) \\
& + \psi_7 (v_{B,f} - f(v_{A,f}, d_{A,0}, d_{B,0}) + \tau_7^2) + \psi_8 (v_{C,f} - f(v_{A,f}, d_{A,0}, d_{C,0}) + \tau_8^2) \\
& + \psi_9 (v_{C,f} - f(v_{B,f}, d_{B,0}, d_{C,0}) + \tau_9^2). \tag{4.13}
\end{aligned}$$

The optimal values for $v_{A,f}$, $v_{B,f}$ and $v_{C,f}$ can be calculated using the same method as for a pair of aircraft, and yields similar results. If the three aircraft are sufficiently close to each other so as to activate the separation constraints in the last three terms of Equation (4.13), the optimal velocity for aircraft A is $v_{A,f} = v_{A,\max}$. Aircraft B and C satisfy the separation constraint to the preceding aircraft with equality. This result greatly simplifies the optimization procedure for multiple aircraft. If the optimization is carried out *in a pairwise fashion in the order of precedence*, the resulting velocities are optimal. If there are n_a aircraft ordered $(1, 2, \dots, i, \dots, n_a)$, the optimal velocity for aircraft i can be calculated by solving the two-aircraft problem with aircraft $(1, 2, \dots, i - 1)$.

4.2.4 Distributed control strategy

The mathematical formulation for the calculation of optimal velocities can be implemented as described in this section and the next. The distributed region requires autonomous calculation of the aircraft's own optimal velocity. Unlike the centralized algorithm, the distributed protocol uses only local information received from ADS-B transmissions. Consider a scenario where aircraft C in the distributed region receives an ADS-B broadcast from aircraft B for the first time. If aircraft B is ahead of aircraft C on the same path, it has precedence over C by default (there is no overtaking on the same path). If they are on merging paths, the conflict detection and resolution protocol observes the relative geometry between the two aircraft. Assume that aircraft B is closer to their eventual merge point and is flying at a velocity lower than its declared maximum velocity. Aircraft C cannot know why aircraft B is not flying at maximum velocity, but is aware of the possibility of information asymmetry in the distributed control region. For example, consider the scenario depicted in Figure 4-4. Aircraft B may be flying at less than its maximum velocity in order to avoid a conflict with aircraft A. However, aircraft C is out of range for broadcasts from aircraft A. It receives an indication of potential traffic in front of aircraft B only by noting that B is not flying at its maximum velocity. Based on this assumption, the distributed algorithm concedes priority to aircraft B.

If aircraft B is flying at its maximum velocity when aircraft C receives its broadcast, the pairwise order is based on the projected arrival times of both aircraft at their eventual merge point. If aircraft C 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 based on the pairwise algorithm in Section 4.2.2. If B is also in the distributed control region, it carries out a complementary set of calculations on detecting C for the first time. Even if aircraft B is under centralized control, it does not affect the computations carried out by aircraft C. Finally, in addition to the detection of a new aircraft, an aircraft recalculates its velocity if there is a change in state (link, velocity or hold) of another aircraft already being tracked. 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 is developed.

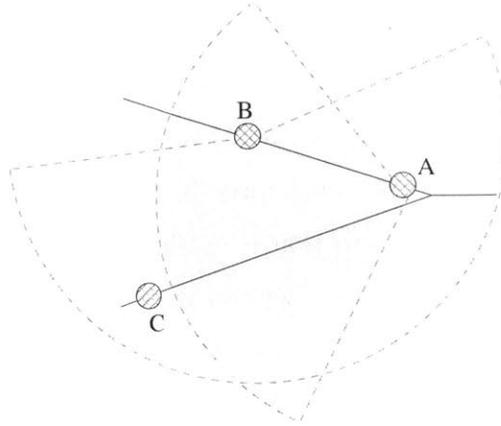


Figure 4-4: Information asymmetry in the distributed zone. The dashed arcs represent range of ADS-B transmissions. Aircraft B is visible to both A and C, but aircraft A and C are not visible to each other.

4.2.5 Centralized control strategy

Each aircraft moves towards the airport through the distributed control region, eventually entering the region of centralized control. This boundary may be defined by the range of ground radars located near the airport, or by the presence of ADS-B ground stations connected to the airport (as explained in [107]). Note that the update rates for secondary surveillance radars are in the 4.8-12.0 sec range, while ADS-B updates happen at an average interval of 0.5 sec [42]. Combined with the cheaper cost of deploying and maintaining ground-based ADS-B receivers, the use of ADS-B as the primary method of surveillance is highly attractive. This does not, however, mean that radar surveillance is obsolete. In addition to providing a backup to the ADS-B system, radars are also the only means of positive surveillance in the airspace. While aircraft can be detected by primary radars independently of their transponder status, ADS-B relies completely on transmissions from aircraft. This is a possible security risk which can be handled using the methods outlined in [43].

Once an aircraft has been detected by the central facility, it no longer needs to carry out its own conflict detection and resolution for efficiency purposes. It is expected that its collision avoidance systems such as TCAS are still active. The status messages transmitted by ADS-B Out are assumed to include the aircraft's estimated position, velocity, feasible velocity range and intended arrival path. Part of this information is included in the standard ADS-B message. The remaining is assumed to be embedded in the empty bits available for miscellaneous purposes, as available in the ADS-B specifications [107]. Note that these state updates can also be carried out by a combination

of secondary surveillance radars and voice communications with the accompanying added latency.

The central facility 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, and the priority order is established according to these predictions. Conflict detection is carried in a pairwise fashion for each pair of aircraft, starting with the aircraft that has the highest priority. Resolution maneuvers (if required) are commanded for the aircraft that are lower in the priority order. Consequently, an aircraft that is i^{th} in the priority order for landing could have up to $(i - 1)$ downward adjustments of its commanded velocity while the control algorithm is processing data. If the commanded velocity is less than the least feasible velocity for that aircraft, it is commanded to enter a holding pattern. Once the computation is completed, the final velocity and / or hold commands are transmitted to each aircraft. With current technology, these commands can be relayed by voice or datalink. Since the time horizon to a possible conflict is expected to be of the order of several minutes, making manual speed adjustments is feasible. In the future, it might be possible to communicate instructions directly to the aircraft's flight management system.

The pairwise conflict resolution procedure is described in Section 4.2.2. If successive aircraft are on paired links, the optimal velocities are calculated by solving the problem formulated in Equation (4.5). If they are on unpaired links, the algorithm allows for a minimum separation of s_{\min} at their projected merge point. If two successive aircraft are on the same link, a separation of s_{\min} is ensured at all times, subject to the physical velocity constraints of each aircraft. If no feasible velocity is found for an aircraft, it is sent to a holding pattern, and resumes its original trajectory after a period of 2 min. Finally, optimal velocities are recalculated for the entire centralized region based on two trigger events: the entry of a new aircraft into the zone of centralized control, or the start or end of a holding pattern by at least one aircraft. The rest of the time, the central facility operates in passive monitoring mode.

4.2.6 Constraint on communication performance

The State Update Interval (SUI) of the communications system is defined to be the time between two successful state vector reports delivered from one aircraft to another. For the two-aircraft case, it is possible to derive the maximum SUI that allows conflict resolution to take place without the

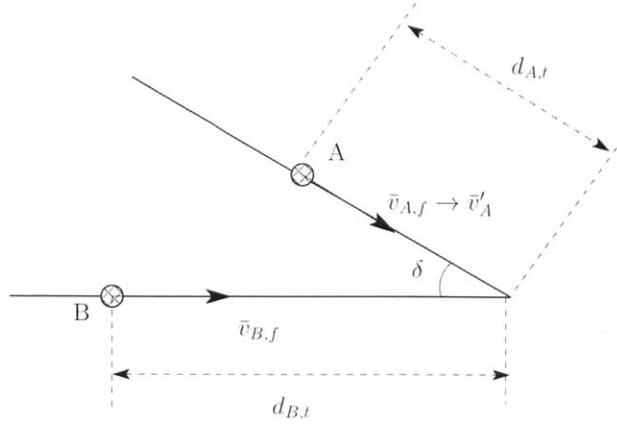


Figure 4-5: Setup for derivation of maximum allowable SUI.

necessity of holding patterns. Let the system consist of two aircraft A and B as shown in Figure 4-5. Aircraft A is scheduled to cross the merge point before aircraft B. Both aircraft have received ADS-B broadcasts from each other, and are flying at their optimal velocities $v_{A,f}$ and $v_{B,f}$ as derived in Section 4.2.2. One simplifying assumption made in the following derivation is that both aircraft can make instantaneous changes to their velocities. Additionally, it is assumed that there is no error in the estimation of position and velocity. Uncertainty in state estimation can be easily added later, as explained in Section 4.2.7.

The maximum allowable SUI that retains network stability is derived below. It is assumed that aircraft arriving earlier at the merge point have higher priority, and that they can change their velocities unconstrained by the aircraft behind them. Suppose aircraft A, flying at velocity $v_{A,f}$, and B, flying at $v_{B,f}$ (Figure 4-5) have previously made contact while at distances $d_{A,0}$ and $d_{B,0}$ from the merge point, and aircraft A has priority. Aircraft A now reduces its velocity to $v'_A \leq v_A$ while at a distance $d_{A,t}$ from the merge point. Aircraft B, which is at distance $d_{B,t}$ from the merge point, needs to adjust its own velocity to maintain separation with aircraft A. Nominally, aircraft A would reach the merge point after a further time $t_A = \frac{d_{A,t}}{v_{A,f}}$, which is changed to $t'_A = \frac{d_{A,t}}{v'_A} \geq t_A$. Thus aircraft B needs to extend its own flight time by an amount equal to $(t'_A - t_A)$. Let Γ_A denote the maximum allowable SUI after which aircraft B can receive an update from aircraft A, and still not have to enter a holding pattern. In other words, aircraft B flies at its original velocity for a further time Γ_A , at which time it receives the velocity update from aircraft A. It then slows to $v_{B,\min}$ until aircraft A is at the merge point. At this time, aircraft B needs to be the same distance

away from the merge point, as it would have been in the original scenario. Let this distance be denoted by $d_{B,f}$. Then the total distance covered by B during the time t'_A is,

$$d_{B,t} - d_{B,f} = \underbrace{v_{B,f} \Gamma_A}_{\text{Up to time of msg reception}} + \underbrace{v_{B,\min} \left(\frac{d_{A,t}}{v'_A} - \Gamma_A \right)}_{\text{After msg reception}}. \quad (4.14)$$

The same distance would have been covered by B in time t_A if aircraft A hadn't changed its velocity,

$$d_{B,t} - d_{B,f} = v_{B,f} \frac{d_{A,t}}{v_{A,f}}. \quad (4.15)$$

Comparing Equations (4.14) and (4.15) yields,

$$d_{B,t} - d_{B,f} = v_{B,f} \Gamma_A + v_{B,\min} \left(\frac{d_{A,t}}{v'_A} - \Gamma_A \right) = v_{B,f} \frac{d_{A,t}}{v_{A,f}}.$$

Simplifying the above equation, the maximum allowable SUI for communication from aircraft A to aircraft B is,

$$\Gamma_A = \frac{\frac{d_{A,t}}{v_{A,f}} v_{B,f} - \frac{d_{A,t}}{v'_A} v_{B,\min}}{v_{B,f} - v_{B,\min}}. \quad (4.16)$$

Equation (4.16) suggests that as $d_{A,t}$ decreases, that is, as aircraft A approaches the merge point, it needs to broadcast any velocity updates with lower latency. It also suggests that if aircraft B is already flying at its minimum speed ($v_{B,f} = v_{B,\min}$), then the only feasible v'_A is equal to $v_{A,f}$, that is, aircraft A cannot slow down without causing aircraft B to enter a holding pattern to maintain separation. The flip side of this insight is to consider the nominal case where aircraft A flies at its original speed, $v_{A,f} = v'_A$. Equation (4.16) then implies that $\Gamma_A = \frac{d_{A,t}}{v_{A,f}}$. Aircraft A only needs to transmit an update when it reaches the merge point, supporting the assumption that control computations need only be rerun when aircraft transition from one link to another. For any $v'_A < v_{A,f}$, the maximum allowable SUI is less than $\frac{d_{A,t}}{v_{A,f}}$, that is, there must be an update before aircraft A arrives at the intersection. This pairwise result can be extended to the case of several aircraft. The communication constraint on aircraft A is the minimum of all the allowable SUIs, as imposed by pairwise calculations with respect to all other aircraft in the airspace.

4.2.7 Challenges to control implementation

While the algorithm described so far is likely to work well under ideal conditions, there are several issues to overcome before it can be implemented in practice. These potential issues and their solutions are described below.

Asynchronous state updates

Due to stochastic transmission times and possible packet loss, state updates between two aircraft, or between an aircraft and the central facility are asynchronous in general. The consistency of control computations can be ensured by including the time stamp of broadcast within each ADS-B message. This feature allows the estimation of the current state of each aircraft based on its last update. It also reduces the likelihood of inconsistent calculations in the distributed algorithm by guarding against a mismatch caused by the clocks on board two aircraft not being synchronized. As long as all aircraft use the transmitted time stamps for state estimation, computations will be consistent regardless of the accuracy and latency of each message.

Uncertainty in state estimation

Uncertainty, both in state measurement and in velocity, is also a challenge to practical implementation. The proposed algorithm can account for uncertainty by appropriately buffering the minimum separation constraint. Once the error model for state measurement has been defined, it is possible to calculate a confidence interval for the actual position and velocity of each aircraft. The control algorithm can then ensure that a conflict cannot occur in the worst case scenario.

Guarding against deadlock

A scenario may arise where two aircraft are projected to reach their merge point at exactly the same time. In this case, the asynchronous nature of ADS-B transmissions proves beneficial. The distributed control algorithm is set to give precedence to the other aircraft in case of deadlock. Since it is very likely that one aircraft receives a state update before the other, it will already have slowed down by the time the other aircraft begins its computations. Even if message delivery is nearly simultaneous and both aircraft reduce their own velocities, a small time difference between the adjustments will be sufficient to resolve the deadlock in the next computation cycle.

Mixed equipage and uncooperative behavior

The control algorithm also gives precedence to non-cooperative aircraft in the airspace, which could be present because of a lack of ADS-B equipage, equipment failure, or some other on-board emergency. Actual non-cooperative behavior can be differentiated from message reception failure by using the SUI to calculate the probability of no messages being received by the aircraft in a given time window.

4.3 Simulations of control strategy

The simulation results presented in this section are based on a model of the Los Angeles International Airport (LAX) Terminal Radar Approach and Control (TRACON), depicted in Figure 4-6. This location was chosen because of the availability of Performance Data Analysis and Reporting System (PDARS) data for the Southern California sector. Using this data, it is possible to empirically measure the traffic rates on various arrival routes, thus providing more realistic simulation results. Operational flight data from this region was also used to verify that the published routes were followed accurately. An equivalent model for the approach procedures to Boston Logan International Airport (BOS) is used in Chapter 5, and is depicted in Figure 5-1. For both LAX and BOS, the airspace model was generated from published Standard Terminal Arrival Routes (STARs).

Simulation procedure

Once the layout of arrival paths has been defined for a given airport such as LAX in Figure 4-6 or BOS in Figure 5-1, the simulator generates aircraft on the periphery of the modeled airspace. For the LAX simulations presented in this section, the PDARS data is useful for defining the rate of incoming traffic and also its distribution along the different arrival paths. Since the generated aircraft are several hundred miles away from the airport, it is assumed that they are initially flying at their maximum feasible velocity. Aircraft-to-aircraft and aircraft-to-ground communication, such as ADS-B broadcast rates, the accompanying range of transmission and the probability of message loss is modeled using the methods described in [42]. The control algorithm is constrained by the best performance that the communications layer is able to deliver. In the distributed zone, aircraft perform their own conflict detection and resolution. An aircraft detects its own entry into the

centralized zone when its broadcast reaches the central facility for the first time, thus prompting the central facility to start issuing commands to it. The aircraft then switches to following centralized control commands. In the simulation, it is assumed that these commands are implemented instantaneously and automatically. However, this is not a greatly restrictive assumption, since the resolution maneuvers are generated several minutes in advance while even manual implementation of the commands should take no more than a few seconds. When holding patterns are commanded in the simulation, they are implemented in the form of trajectories orthogonal to the current link that the aircraft is on: one minute outbound and one minute inbound, with the aircraft resuming its flight from the same point that it left the link from. The simulation is seeded by Poisson processes for traffic generation on the periphery, with the rate for each route being defined using historical data. In the following discussion, the net traffic generation rate is denoted by β' , and the average time between two aircraft appearing on the periphery by $\frac{1}{\beta'}$.

Effect of different radii of centralized control on holding patterns

Holding patterns in the airspace are an indicator of congestion and instability within the network. These holds are necessary when a pure velocity change by an aircraft cannot guarantee satisfaction of the minimum separation constraint. In dense traffic, one holding pattern typically causes a cascade of holding patterns upstream, affecting a large section of the airspace. Figure 4-7 shows the generation of holding patterns for three different radii of centralized control. Each simulation is run with the same schedule of arrivals at the periphery of the airspace. Note that there are two bursts of arriving traffic as highlighted in the lower left plot. The distributed control algorithm is efficient enough to carry these aircraft through the outer part of the airspace. However once they enter the congested central part of the airspace (after an approximate flight time of ΔT as marked in the figure), they necessitate hold commands in all three cases. When the centralized radius is small, the more efficient centralized control algorithm is unable to recover from the first burst of traffic before the second burst arrives. Therefore, the number of holding patterns exhibits unstable behavior. For larger radii of centralized control (65 nm and 110 nm), recovery can be seen. Note also that the number of active aircraft in the airspace for the latter two cases is nearly the same - a centralized radius of 110 nm does not have an advantage over the smaller radius of 65 nm in this regard.

Comparison of control performance for different scenarios

Figure 4-8 shows the average number of holds commanded per hour for traffic loads $\frac{1}{\beta T} = 45$ sec, 65 sec and 90 sec, as a function of the radius of the centralized region. The reduction in the number of holds on moving from a radius of 65 nm to 110 nm is seen to be quite small. As the traffic arrival rate increases, there is a much more marked change in moving from a 30 nm radius to a 65 nm radius.

Trends in total traffic in airspace for different scenarios

Figure 4-9 further emphasizes the unstable nature of the network for high traffic rates and small centralized zones. It shows a time series of traffic for two different sizes of the centralized zone. Centralized control applied to the larger region (110 nm) is seen to stabilize the traffic in all three cases $\frac{1}{\beta T} = 45, 65$ and 90 sec. On the other hand, the smaller region (30 nm) cannot cope with higher traffic loads, and experiences a continuous increase in the number of active aircraft in the airspace, most of which have been delayed in the central region. While holding patterns are generated in bursts, low to moderate traffic loads allow the airspace to recover and resume smooth operations. However, traffic accumulates if more holds are generated before this recovery is complete, as can happen with high traffic loads.

Performance comparison with current ATC procedures

Current air traffic control procedures rely heavily on human supervision, and are difficult to model exactly. However, it is reasonable to assume that aircraft are only deconflicted up to the next merge point, and downstream conflicts are resolved as they emerge. Figure 4-10 compares this approach to the proposed control strategy for $\frac{1}{\beta T} = 65$ sec. It shows the average amount of time required by aircraft at the periphery to land at the airport. Since all the simulations start with an empty airspace model, the initial flight times for all cases are similar (approximately 2.75 hrs). However, as time progresses and the airspace congestion increases, the difference in performance becomes significant. The proposed control algorithm performs significantly better than the current operations for all values of the central radius. Increasing the radius of the centralized region increases the efficiency of the control algorithm up to a point (seen to be at 65 nm for the current model), beyond which the marginal benefits are minimal.

Decreasing the radius of the centralized zone reduces the number of ground radars near a terminal area and the associated ground infrastructure cost. However, a smaller centralized zone degrades both communication and control performance. An arbitrarily large region of centralized control not only entails large costs, but also fails to show significant improvement in performance. The traffic density far away from an airport is small enough for the distributed control algorithm to perform nearly as well as the centralized algorithm. In the simulations presented in this section, it appears that a centralized radius of 65 nm is sufficient for attaining a major portion of these benefits. The optimal size of the centralized region can be determined for other locations using similar methods.

Performance degradation with measurement uncertainty

As stated in Section 4.2.7, measurement uncertainty can be handled by suitably padding the minimum separation requirement. For example, it may be assumed that the error in position and velocity measurement is independent and Gaussian with a known standard deviation. Optimal velocities for each aircraft pair can be calculated by assuming that the leading aircraft is flying slower and is farther from the merge point than its measured position and velocity. Similarly, it can be assumed that the trailing aircraft is flying faster and is closer to the merge point than its measurement. Note that the priority order is still calculated based on measurements: this ensures that the computations on board both aircraft are consistent. By solving for the optimal velocities in this worst-case scenario, a conflict can be avoided with a certain user-defined probability. This solution necessarily makes the control algorithm more conservative than for the deterministic case. For example, a trailing aircraft in a paired merge may be asked to enter a holding pattern to increase separation, while this would not have been necessary in the deterministic case. An idea of the resulting performance degradation can be obtained by measuring the number of holding patterns commanded per unit time, and also the total traffic level in the simulation. These results are presented in Table 4.1 for a traffic generation rate of $\frac{1}{\beta'} = 90$ sec.

Position Unc. (one std. dev., nm)	Velocity Unc. (one std. dev., kts)	Av. holds (per hr)	Max traffic
0.1	0	0	100
0.1	1	0	100
1	2	20	110
1	3	60	110
1	5	750	180

Table 4.1: Variation in performance with position and velocity uncertainty.

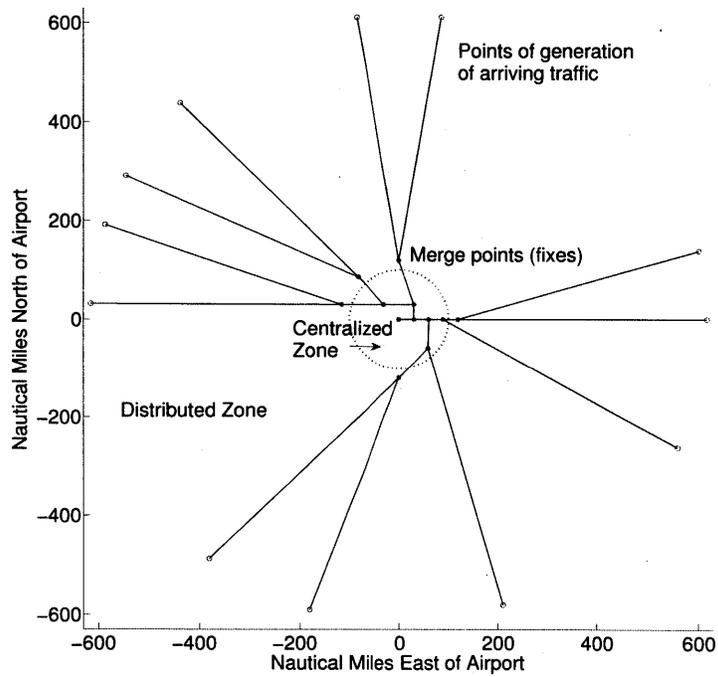


Figure 4-6: Model of the airspace around Los Angeles International Airport, with procedures for arrivals on Runway 25L. The airport is located at the origin.

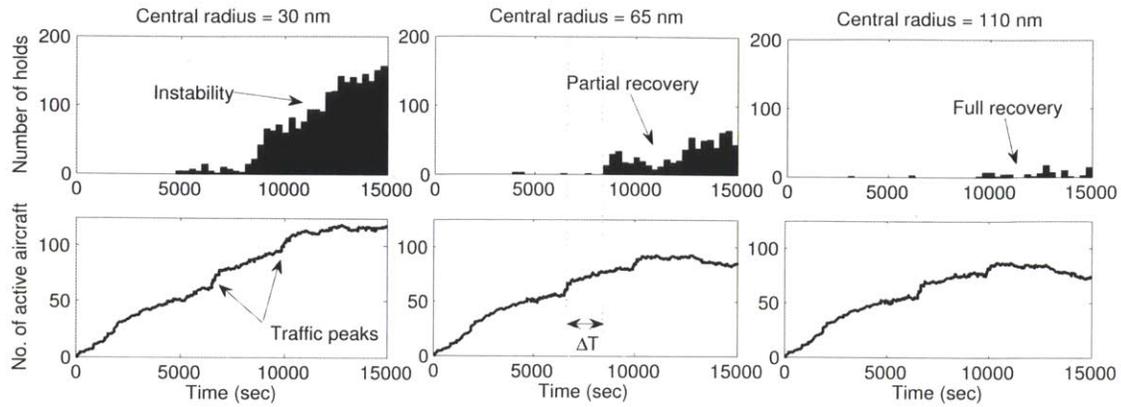


Figure 4-7: The onset of holding patterns with different radii of centralized control. The traffic arrival rate is one aircraft every 65 sec on average. ΔT is the approximate flight time through the outer part of the airspace.

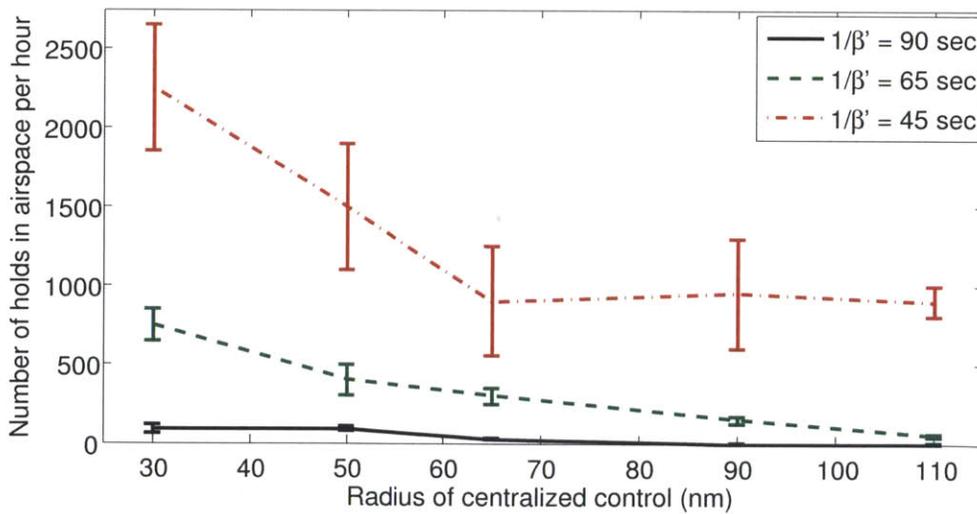


Figure 4-8: Number of holding patterns generated by the control strategy for different levels of incoming arrival demand.

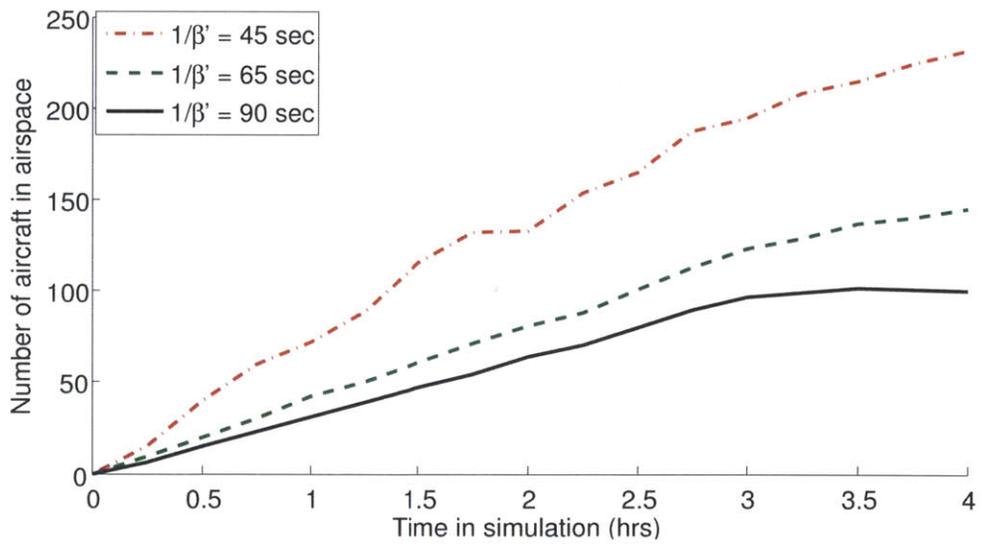
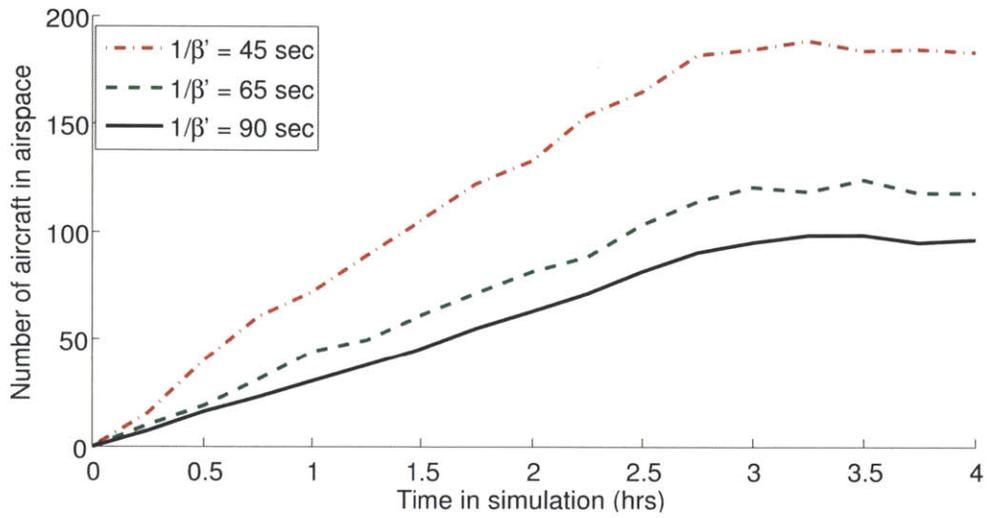


Figure 4-9: Traffic levels in the simulated airspace for different radii of centralized control.

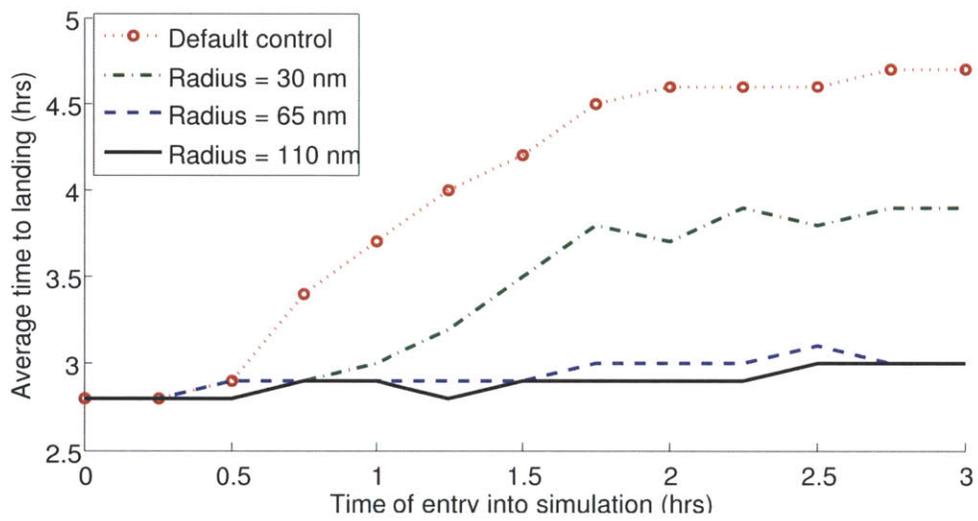


Figure 4-10: Average time required for aircraft on the periphery of the modeled airspace to land at the airport. Traffic is being generated at the average rate of 1 aircraft every 65 sec.

Chapter 5

Integrated Control of Arrivals and Departures

The preceding two chapters have described the control algorithms for departures on the surface (Chapter 3) and arrivals in the approach airspace (Chapter 4). The purpose of this chapter is to combine the two algorithms into a single, integrated strategy for managing airport operations. This proposed strategy is described in Section 5.1. A logical procedure for formulating and calculating the integrated control algorithm for a generic airport is described in 5.3.

5.1 Combining the arrival and departure algorithms

This section develops the integrated control strategy for arrivals and departures, using Boston Logan International Airport (BOS) as an example. The arrival paths around BOS for landing on Runway 33L are shown in Figure 5-1. This runway is commonly used for arrivals when departures are using Runway 27 [124]. The layout depicted in Figure 5-1 is thus complementary to the departure control policy developed in Section 3.2.6. The modeled airspace is a circle around the airport with a radius of 475 nm. A major assumption in the development of the model is the absence of interaction with other airports in the vicinity. This is not a realistic assumption to make for BOS, but is useful in developing the control algorithms described in this thesis. While a more realistic model is required for calculating the optimal policies for actual implementation, the simplifications used in this chapter allow the algorithm to be demonstrated without the effect of externalities. With

this objective in mind, the arrival traffic at BOS originating from nearby airports such as those in the New York area is instead generated on the periphery of the modeled airspace. Unlike the case of LAX where the split of incoming traffic could be based on operational data, the split at BOS is based on the estimated from various regions of origin.

Based on simulation runs of the model depicted in Figure 5-1, it is possible to obtain an estimate of the airspace capacity. This can be done by testing the model at various traffic generation rates β' , and noting the landing rate β at the airport. If the incoming traffic rate results in a stable airspace network, steady state is achieved with $\beta = \beta'$. The throughput of the airspace model can be characterized by plotting an analogous curve to that in Figure 1-2. The number of arrivals at the airport as a function of the number of airborne aircraft within the modeled airspace is shown in Figure 5-2. A reasonable estimate of the steady state achieved in this case, is an average of 10 aircraft every 15 min. This is equivalent to an average separation of 90 sec between landings, and is equal to the traffic generation rate. This means that 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 happens to be 1 aircraft every 70 sec. Note that the capacity estimate may not correspond to the real-world airspace capacity of the Boston TRACON, because of the simplifications listed above. Should an accurate estimate of the capacity be desired, a similar setup can be simulated but with more realistic traffic characteristics. For the purposes of this thesis, it is sufficient to develop an idea of the maximum rate at which the airborne portion of the model is likely to deliver aircraft to the airport surface. This estimate informs the set of policies that are calculated offline for the departure control algorithm.

Based on the range of possible landing rates β , a set of departure control policies can be calculated offline. The policy used in real-time is selected from this set of policies as a function of the number of expected arrivals over a fixed time horizon. This ensures that the best possible fuel savings are obtained for that current arrival rate β , as explained in Section 3.2.6. The risk of gate conflict is concurrently minimized. The control policy is more aggressive (assigns larger gate delays) when the expected β is low, and vice versa.

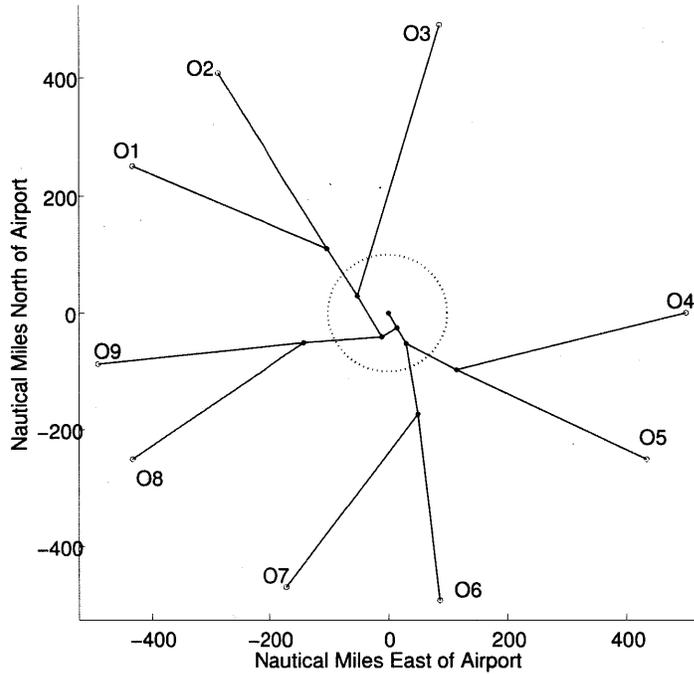


Figure 5-1: Model of the airspace around Boston Logan International Airport with procedures for arrivals on Runway 33L. The airport is located at the origin.

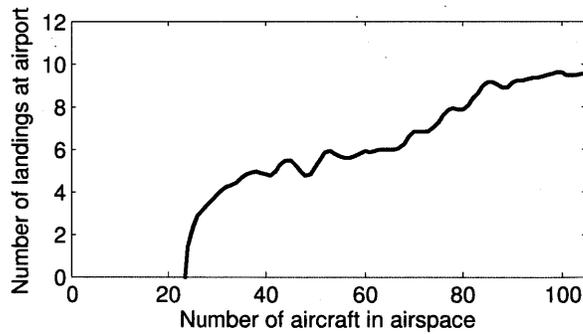


Figure 5-2: Arrival throughput characteristics at Boston Logan. The y axis shows the average number of landings in a 15 min interval, when the corresponding number of active aircraft is on the x axis. The radius of centralized control in this plot is 90 nm, and the traffic generation rate is 1 aircraft every 90 sec on average.

Time	Av. interval between generation of two aircraft (min)
00:00 - 06:00	8.3
06:00 - 10:00	1.0
10:00 - 18:00	1.6
18:00 - 21:00	1.0
21:00 - 00:00	6

Table 5.1: Rates used for generation of arriving traffic at the periphery of the modeled airspace.

5.2 Performance evaluation

The integrated control strategy for departures and arrivals and departures is evaluated in this section through simulation. A bespoke discrete event simulator has been developed for this purpose. The simulator propagates aircraft from the periphery of the arrival airspace, through the taxi-in, gate service and taxi-out processes, to the instant of takeoff.

5.2.1 Simulation procedure

The arrival airspace model described in Section 4.1.2 and the surface taxi model described in Section 2.2.3 are the primary components of the combined arrival/departure simulation procedure. The schematic is shown in Figure 5-3. Incoming airborne traffic is generated at the periphery of the airspace, and is propagated to the airport. In the following simulations, the rate of traffic generation is Great Lakes and Canada (O1-O2 in Figure 5-1) 10%, Great Circle Transatlantic (O3) 5%, Southern Europe (O4) 5%, Other international flights and US East Coast (O5-O6-O7) 40%, Midwest and US West Coast (O8-O9) 40%. The airport predicts landing times for all aircraft that are within broadcast reception range, i.e., aircraft in the centralized control region. These landing time predictions are fed to the departure control algorithm, which also adds predictions of the taxi-in times and forms an estimate of the imminent arrival rate to each buffer, β_i . These arrival rates, in combination with the departure surface traffic level k , drive the pushback control policy. The surface traffic simulation separately generates actual taxi-in and taxi-out times for arrivals and departures respectively.

Aircraft in the arrival airspace are assumed to be using approach paths for landing on Runway 33L at the airport, as shown in Figure 5-1. The surface movement simulator 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 Figure 2-1). This runway has been chosen because it is used for

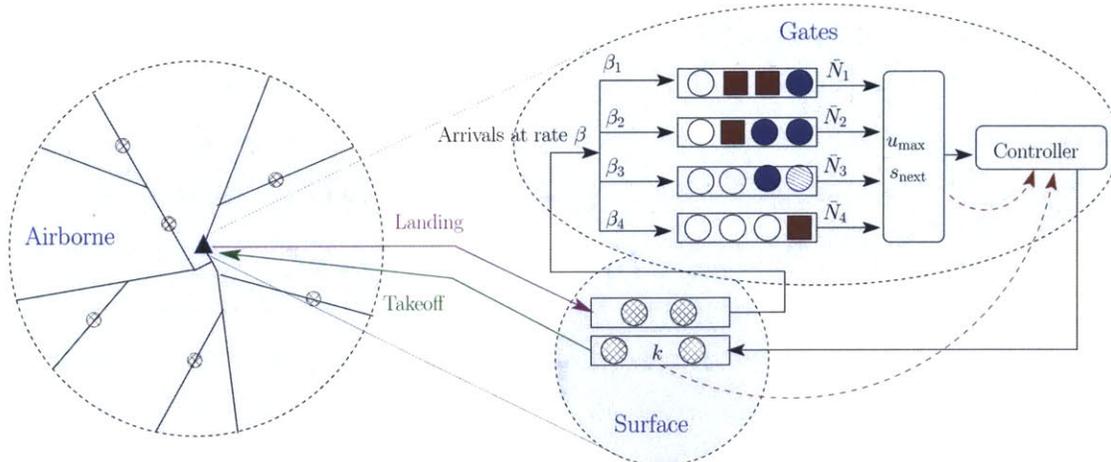


Figure 5-3: Schematic of the combined control algorithm and simulation procedure.

departures at Boston when aircraft are arriving on Runway 33L (node 7 in Figure 2-1). The four source nodes at the airport (nodes 1, 2, 3 and 8 in Figure 2-1) have capacities of 25, 20, 25 and 20 gates respectively. This results in a total of 90 gates at the airport. For the purpose of this work, all gates at a given terminal are assumed to be equivalent, that is, an arriving aircraft can park at any gate at its assigned terminal. This assumption is convenient for demonstrating the proposed algorithm, but is not accurate in reality as each airline has access to only a specific subset of gates at a terminal. The current formulation can accommodate this level of fidelity by defining separate nodes and corresponding buffers for each set of gates owned by an airline.

5.2.2 Simulation results

A simulation of one day's operations at Boston Logan is shown in Figures 5-4 and 5-5. Aircraft appear at the periphery of the 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. The uppermost plot in Figure 5-4 shows that the peak morning demand appears at the periphery of the airspace at approximately 03:00 local time, consisting mainly of transcontinental and transatlantic traffic. These aircraft reach the airport just before 06:00, when the number of landings increases to about 10 aircraft every 15 min. This corresponds to an average separation of 90 sec 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 2.2.3. A gate service

time for each arrived aircraft is generated from a uniform distribution between 30 and 45 min. The choice of this distribution is based on the general turnaround times seen for commercial aircraft in empirical studies such as [125]. During this time, the gate is marked occupied-inactive. Once servicing is completed, the aircraft is ready for pushback and the gate is marked occupied-active. The occupied-active gates are assigned pushback delays on a first-come-first-served basis.

After pushback, the simulator propagates each aircraft to Runway 27 for departure. As seen in Figure 5-4, the morning departure demand trails the arriving one by a period of approximately 45 min. The average departure rate at the airport is also seen to stabilize at 10 aircraft every 15 min. The middle plot shows the average taxi times for arrivals and departures, as generated by the simulator. Since arriving aircraft are not subject to the higher traffic levels (shown in the bottom plot) seen by departures, average arrival taxi times are significantly lower than departure taxi times. This is consistent with empirical data. An interesting feature of Figure 5-4 is the spike in the departure traffic level seen in the bottom plot at 20:00. This can be correlated with the gate occupancy plotted in Figure 5-5. 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, thus temporarily increasing the surface traffic level. This behavior is a useful feature of the proposed control algorithm, where surface congestion is balanced against the risk of buffer overflow.

5.2.3 Effect of control strategy on taxi-out times

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

A comparison of the distribution of aircraft taxi-out times seen in the three simulations, is shown in Figure 5-6. The base schedule used in this figure is the same as that generated in Section 5.2.2. It should be emphasized that this schedule reflects the pattern and volume of demand empirically seen

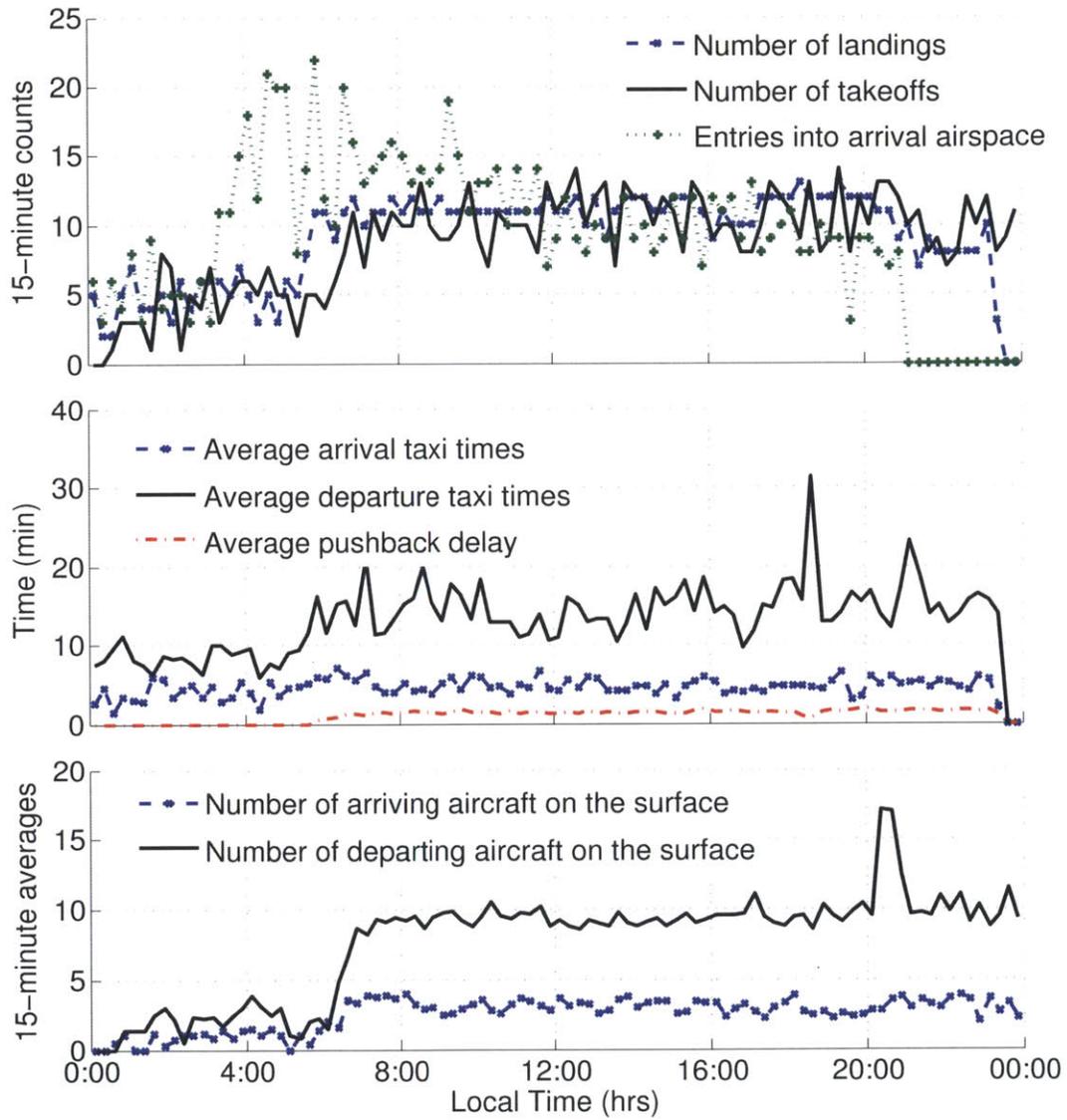


Figure 5-4: Operation counts, average taxi and pushback delay times, and surface counts in a full day's simulation.

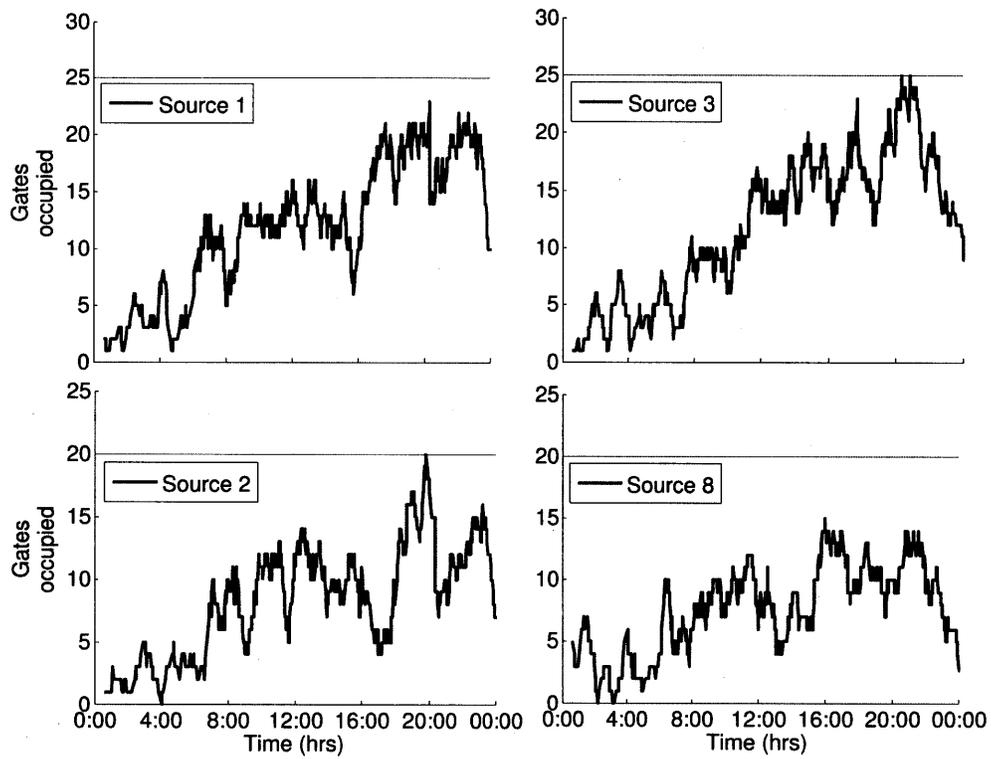


Figure 5-5: Gate occupancy over the course of the simulated day, at each of the four sources (airport terminals).

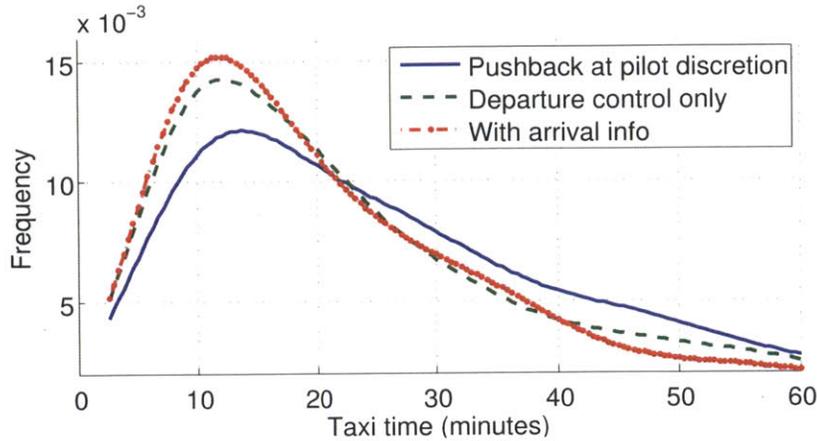


Figure 5-6: Comparison of the distribution of taxi-out times for three different control strategies.

at BOS. It can be seen that the frequency of long 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 taxi-out time reduction per aircraft in this simulation is 0.85 min for the departure control algorithm. Using arrival information further reduces the mean taxi-out time by 0.75 min per aircraft. Using the procedure described in [108], the estimated fuel burn reduction per medium-sized aircraft is 12.5 kg (4.5 gallons) for the departure control algorithm, and a total of 22.5 kg (8 gallons) per aircraft after including arrival information.

A comparison of the average taxi-out times and frequency of gate conflicts seen for the three strategies is given in Table 5.2. These averages are calculated over 10 simulation days, with all three control strategies using the same pushback 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 Boston Logan by 5.2%. The departure control strategy with arrival information reduces taxi-out times by 10.2% compared to current procedures. Since holding aircraft at the gate increases the buffer occupancy, we see an increase in the number of gate conflicts (instances of buffer overflow). The fraction of aircraft with gate conflicts is well within the tolerance level γ . This added conservativeness could be because of the simplifying assumptions made while calculating state transition probabilities in Section 3.2.1. It should also be noted that including arrival information reduces the number of gate conflicts by 30% in addition to the reduction in

Control strategy	Av. taxi time (min)	Average gate conflicts in 24 hrs	% flights with gate conflicts
Pilot's discretion	17.2	14	0.5%
Dep. control only	16.3	33	2.7%
With arrival info	15.4	24	1.9%

Table 5.2: Comparison of taxi-out times and frequency of gate conflicts. The averages are calculated over 10 simulation runs. The buffer overflow tolerance is $\gamma = 5\%$.

mean taxi-out times. Depending on user preference, the expected number of gate conflicts can be traded off with the reduction in taxi-out times, using the tolerance level γ .

It is important to note that the actual values of the various savings presented here, are specific to simulations of one configuration at Boston Logan International Airport. These values will change based on the runway configuration, the airport, and the simulation assumptions, but the principle behind the increase in efficiency will remain invariant.

5.2.4 Effect of control strategy on takeoff delay

As stated before, one major objective of the control strategy is to not increase aircraft takeoff times beyond those attained using current procedures. This will ensure that the benefits of reduced fuel burn are not at the cost of airport performance. Figure 5-7 shows the distribution of simulated takeoff delay under the two control strategies. Here, takeoff delay is the difference between the takeoff time under a pushback control strategy and the takeoff time under current procedures. It can be seen that the relative takeoff times are 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. This means that on average, aircraft are taking off earlier than they do under current procedures. The average decrease in takeoff times is 0.37 min and 0.62 min under departure control without and with arrival information respectively.

5.3 From modeling to implementation: A summary

The last four chapters have provided a detailed description of the logical steps involved in the implementation of congestion control at airports. Boston Logan International Airport has been used as a representative example throughout the development. The aim of this section is to provide a concise description of the entire process involved with implementing the proposed strategy at a

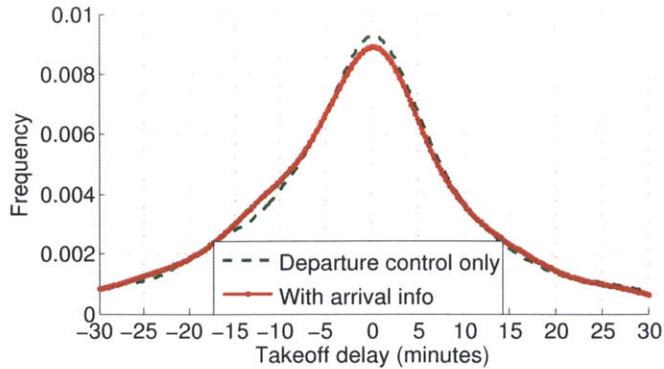


Figure 5-7: Comparison of the distribution of takeoff times relative to those attained using current procedures. Note that all three cases (current procedures, departure control only, and combined arrival/departure control) are based on simulations. Large differences in taxi times could arise in a few cases due to the stochasticity of the underlying processes, and the overall distribution is more significant than individual outliers.

new location. The schematic of this proposed strategy is shown in Figure 5-8. Note that the arrival control strategy can be implemented independently of the departure control strategy, but not vice versa. However, the departure control strategy only depends on the arrival control strategy to the extent of receiving estimates of the expected arrival rates. These rates can also be estimated using other means. Both strategies can thus be implemented stand-alone or in a combined fashion.

Control of airborne arrivals

The procedure for instantiating the arrival strategy is depicted in the top half of Figure 5-8. The offline simulation portion in the top-left quadrant is necessary for two reasons. Firstly, it can be used to estimate the capacity of the airspace as it is currently laid out. This capacity estimate is used by the departure control algorithm for generating policies that can handle the range of potential landing rates at the airport. Secondly, the simulation procedure can also be used to compute the number and locations of ADS-B ground stations required for managing the arrival traffic.

For modeling the arrival airspace more accurately, the model presented in Section 4.1.2 should be modified to handle all the different runway configurations that may be in regular use at the airport. Each arrival runway may have different arrival procedures depending on orientation, airspace layout and inter-dependence with other airports [126]. The effect of airports in close vicinity can also

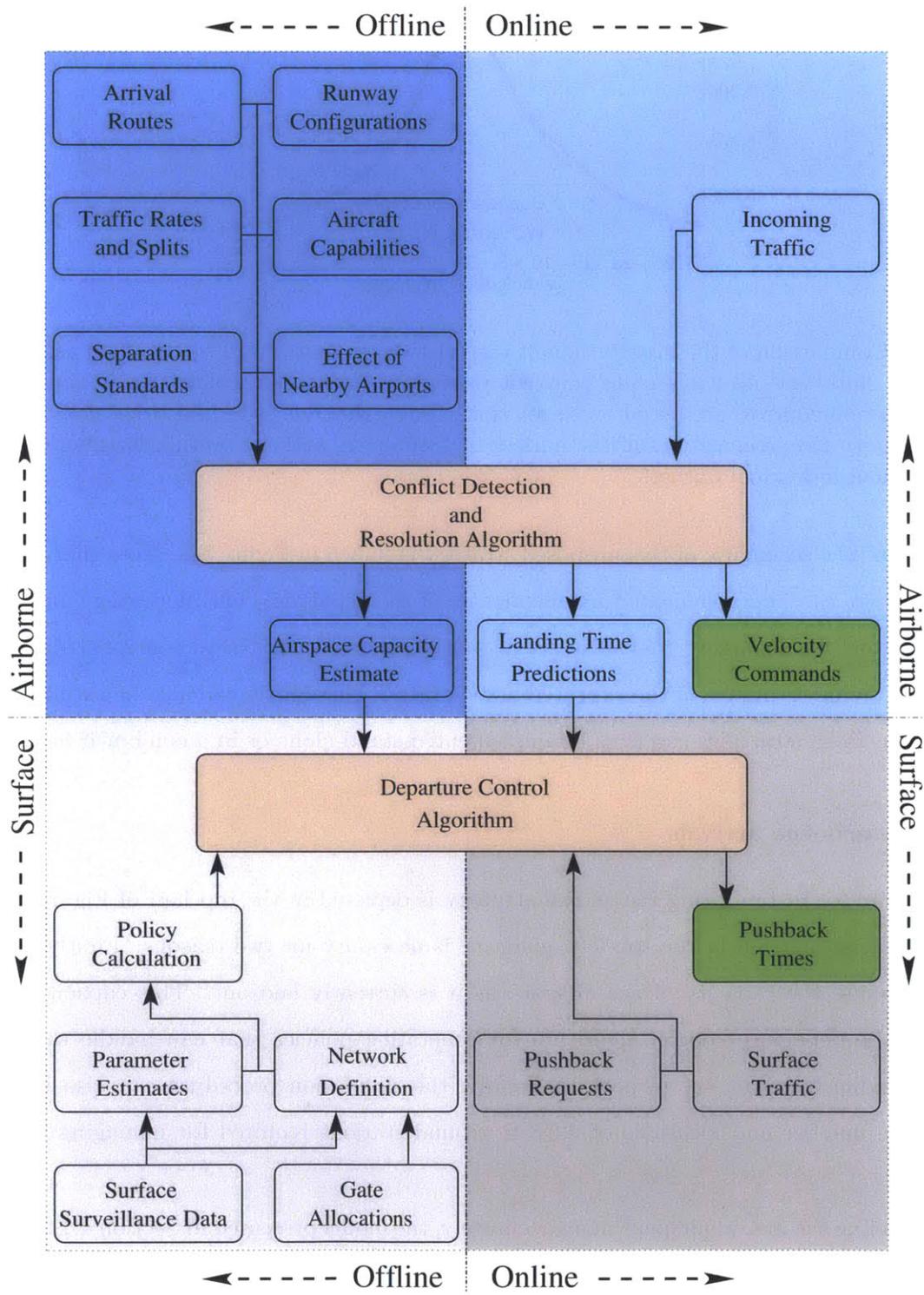


Figure 5-8: Schematic of proposed combined control strategy for arrivals and departures.

include the generation of arrival traffic within the modeled airspace. This traffic will also exhibit climbing and descending trajectories, and the control algorithm will need to be modified slightly to include the possibility of vertical separation in addition to horizontal separation standards. The basic structure of the strategy remains unchanged if the different altitude ‘layers’ are deconflicted separately. It should be noted that aircraft that are separated vertically are not a factor for the control algorithm, but are significant for the communications algorithm. Similarly, airborne departures affect the communications algorithm but are not a direct factor for the control algorithm. These routes are usually separated from the arrival routes in terms of heading or altitude. A detailed study of their impact on the system is described in [42].

Once the airspace structure is modeled, the constraints imposed by minimum separation requirements can be introduced. Note that both the separation standards and the allowed minimum/maximum velocities can change depending on flight altitude and proximity to the airport. Another factor which may change the separation standard is the possibility of multiple runways being used for arrivals, especially for simultaneous parallel approaches. Since the pairwise conflict resolution algorithm is solved repeatedly by the proposed control strategy, changing the constant terms in these constraints depending on location should not pose major challenges. The introduction of position and velocity uncertainty is also desirable in order to obtain realistic estimates from the model. This problem has been addressed in Section 4.3 and in [43], where it is shown that uncertainty can be accounted for by suitably padding the minimum separation value s_{\min} . The extent of the added separation is a function of the level of uncertainty. The final piece required for completing the offline portion of the arrival control strategy is an estimate of the expected rates of incoming traffic seen along each of the possible arrival paths. This can be obtained from historical data and/or scheduled flight data.

Control of surface movement

The lower half of Figure 5-8 illustrates the procedure for developing the departure control strategy. The nodes and links of the network can be defined using insights from surface surveillance data. Estimating the importance of taxiways and intersections in terms of usage levels, is generally not possible using only airport maps. Air traffic controllers tend to develop preferences for certain taxi routes depending on an aircraft’s departure fix, likelihood of being affected by traffic management

initiatives, and so on. Viewing surface surveillance data from as many days as possible helps pick out the most heavily loaded taxiways. Gate allocations for the different airlines operating at the airport are also an important factor in developing the surface network. The model developed in Section 2.1.2 contains only one node per airport terminal, with the assumption that all gates at a given terminal are equivalent. For a more realistic model, the ownership/leasing of gates by certain airlines, as well as aircraft size restrictions will need to be taken into account. This will result in additional source nodes at each terminal, with corresponding buffers. With this change, the usefulness of the aggregation method described in Section 3.2.6 becomes even more pronounced.

Just as additional nodes may be required to represent the sources in the network, there may also be more than one sink node. This can happen in the case of multiple departure runways. The only configuration at Boston Logan where more than one departure runway is used for jet aircraft is (4R, 4L | 9, 4R), where runways 9 and 4R are used for departure. Figure 2-1 marks these runways by nodes 14 and 15 respectively. It can be seen that the departure thresholds are very close together, with the result that the taxi routes used by aircraft heading to these runways are largely the same. Therefore, the taxi time parameters as well as their dependence on surface traffic levels remain unchanged. At airports where the taxi routes to different departure runways are greatly separated, some modifications may be necessary for modeling the variation of stopping probability with surface traffic. In particular, it should be ascertained whether a single value of surface traffic is sufficient for explaining the variation in taxi times. It might be necessary to split the traffic level by departure runway, if the networks are largely independent.

Finally, the calculation of optimal departure policies can be carried out based on the method outlined in Section 3.2.6. The choice of allowable control (pushback delay) values can be defined using a combination of simulations and operational constraints. The maximum allowable might be set to a large value for the initial policy calculations. The highest value used for typical operations can be determined using simulations, since this will be driven by the gate capacity and arrival rates for the given airport. Once this is determined, the higher values of allowable delay can be eliminated for simplicity. A constraint might also be imposed by user preferences, if the airlines are unwilling to accept delays greater than a certain value.

Protocols for implementation

With the policies pre-calculated and the control algorithm in place, it remains to define the protocols for implementation in real-world situations. In this context, the departure control algorithm can be implemented sooner and with fewer modifications to current procedures than the arrival control algorithm. Controlling arrivals using the procedures outlined in Chapter 4 first requires significant penetration of ADS-B technology. This is expected to take place by the year 2020 [60]. In addition to the mandatory ADS-B Out technology, a sufficient number of aircraft should also be equipped with ADS-B In for the control strategy to deliver benefits. The proposed protocol is able to account for some fraction of non-equipped traffic as explained in Section 4.2.7, but it is reasonable to expect that significant benefits can be delivered only when the majority of aircraft are equipped with ADS-B In. This is especially true in the distributed control region, where it is not possible for air traffic controllers to transmit commands via voice communication.

Assuming that every aircraft flying near a major airport is equipped with ADS-B Out and that a large majority is also equipped with ADS-B In, the arrival control strategy can be implemented in a few different ways. The most efficient and simultaneously the longest term idea is to implement both the distributed and centralized algorithms automatically, with the pilots only performing a supervisory role. This will require significant advances in technology as well as extensive testing for safety. On a near term basis, the on-board or centralized computations can be implemented automatically, and the velocity commands can be delivered via data-link and cockpit displays to the pilots. This implementation can be very similar to current TCAS technology. As explained in Chapter 4, the time scales are favorable enough for the generated recommendations to be implemented within a time period of several seconds, enough for human pilots to process the advisories and make suitable velocity adjustments.

The departure control algorithm can be similarly implemented on a fully or partially automated basis. The fully automated procedure would require the air traffic controller to log a change in gate status (free / inactive / active) depending on unfolding events (pushback, pull-in and push-ready call respectively). In case of a push-ready call, the control algorithm would then measure the number of active departing aircraft and the current gate status based on a surface surveillance data feed. The expected arrival rate can be fed in based on predictions from the arrival control algorithm. Even if this is unavailable, a reasonably reliable estimate of the expected number of arrivals over

a small time horizon (typically the next 15 min) is already available to traffic managers in the control towers of most busy airports [127]. The control algorithm would then deliver a pushback delay recommendation which could be transmitted to the aircraft by voice or data-link. Based on the Collaborative Decision Making (CDM) efforts already in place at several airports, this level of automation is expected to be achievable with current technology. Alternatively, the current traffic level, gate occupancy and expected arrivals can be fed in manually into a hand-held device by an air traffic controller, as has been implemented in an earlier study at Boston Logan airport [73].

Chapter 6

Conclusions

6.1 Summary

The goal of this thesis was to develop a paradigm for the management of aircraft operations in and around airports, with the focus on making the strategy feasible for practical implementation in the mid- to long-term. 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 introduced several novel ideas to the realm of surface congestion control, chief among which was modeling the airport surface as a network. This was enabled by the availability of surface surveillance data from the Airport Surface Detection Equipment (ASDE-X) system. For the first time, a detailed model of airport operations could be developed. This included an explicit consideration for the physical events constituting the taxi process. The departure control algorithm also included relevant constraints such as limited gate capacity at the airport. Since the problem was nonlinear and was constrained, the optimal control policies were calculated using dynamic programming. An additional state aggregation procedure was introduced, in order to enable fast policy calculation.

The arrival control algorithm combined the communication and control aspects of air traffic management, and proposed a strategy for improving terminal airspace operations. It was assumed that the primary communications system would be Airborne Dependent Surveillance Broadcast (ADS-B). The hybrid centralized/distributed strategy was shown to have performance comparable to fully centralized strategies, while requiring significantly lesser ground infrastructure. The

strategy can be implemented by transmitting commands over a data-link and displaying them as advisories to pilots, thus potentially allowing near-term implementation in the real world. An integrated control strategy was then described, that combined the management of arrival airspace and surface operations. This strategy creates a synergy between the various aspects of airport operations, with departures being released from their gates as a function of surface congestion as well as expected aircraft arrival rates. A balance is struck between the objective of fuel burn reduction and the constraint 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 (Table 5.2). At the same time, the implementation requires only the knowledge of the surface traffic level and the gate occupancy at each terminal. Both these quantities are easily obtainable in real-time. It is thus postulated that the proposed control strategy will be an effective tool for reducing aircraft fuel burn, without adversely affecting airport performance.

It is important to note that the actual values of the various savings presented in Chapter 5, are specific to simulations of one configuration at Boston Logan International Airport. These values will change based on the runway configuration, the airport, and the simulation assumptions. However, the fundamental mechanism of airport efficiency improvement will remain invariant. Given the layout of the arrival airspace, the taxiway system on the surface, and a sufficient amount of surveillance data, the modeling and control procedure presented in this thesis is potentially applicable to any airport or runway configuration.

6.2 Future research opportunities

The focus of thesis has been on the application-oriented development of control algorithms. Consequently, there are several future research directions that can be pursued in the realm of practical implementation of these ideas. There are also opportunities for extending the theoretical approaches presented for modeling and control policy calculation.

Improvements to arrival airspace model

The implementation plan described in Section 5.3 contained elements that are not present in the current formulation. These additions will help to reduce the uncertainty in the expected performance of the control strategy. For the control of arrival airspace, an extension of the basic method to include interaction with nearby airports is likely to be useful. This entails the addition of a third dimension to the modeled airspace, as well as including aircraft that first appear in the middle of the region rather than the periphery.

Departure control for multiple runway configurations

For departure control, the most obvious extension is to multiple departure runway instances. Depending on the independence between the two (or more) runways, different measures of surface traffic levels could be necessary. As explained in Section 5.3, the runways in the (4R, 9 | 4L, 4R) configuration at Boston are so tightly coupled that no modification is necessary to the current taxi model. However, this is unlikely to hold true for locations such as Dallas Fort-Worth or Atlanta Hartsfield, where the taxi paths to the two runways are largely separate.

Inclusion of weather effects

It was noted in Section 2.2.2 that weather can have a significant effect on aircraft operations, both in the air and on the ground. The current formulation handles weather uncertainties by including them in the stochastic models or by adaptive methods such as the one described in Section 3.2.8. However, there is an opportunity to include weather more explicitly in the predictive models.

Coordinated arrival / departure runway operations

An interesting extension that involves both the arrival and departure control strategies, is to introduce the possibility of coordinated runway operations. The current implementation assumes that arrivals and departures are controlled separately, with their interactions captured by the stochastic process representations. Some of the stochasticity can be reduced by explicitly coordinating arrivals and departures on crossing runways.

Higher fidelity simulation algorithms

The current version of the arrival/departure simulation is sufficient for estimating the average savings in taxi time / fuel, but there is scope for improving the fidelity in this simulation. In particular, if a single day's operations are to be simulated exactly, it will be necessary to include explicit overtaking constraints, runway occupancy times, etc.

Large-scale stability analysis for arrival airspace

The stability results presented in Section 4.2.6 focus on the two-aircraft system, with the assumption that all conflict resolution is to be carried out in a pairwise fashion. This is a reasonable assumption to make for current and/or moderately elevated levels of traffic. However, the simulations presented in Section 4.3 show that there is a cascade effect in the onset of holding patterns for high traffic levels. In order to analytically estimate the maximum capacity of arrival airspace as it is currently set up, it will be necessary to develop a more comprehensive stability analysis for the full network. This will enable a case to be made for airspace restructuring as and when necessary.

Effect of uncertainty on the potential benefits offered by the control algorithms

Many of the extensions that have been proposed in this section are focused on reducing the uncertainty in aircraft operations. It is important to remember that this is a realistic goal only for the type of uncertainty that is introduced due to modeling assumptions. In the case of aircraft operations, some of the uncertainty is inherent in the process itself. In this case, the focus should be on accurate quantification of its properties. This is the core principle of the proposed methodology for modeling and control of airport processes.

The extensions proposed in this section will improve upon both aspects of stochasticity-dependent inefficiency. Better models of the arrival airspace and of the effect of weather on airport operations will help to characterize the inherent stochasticity in airport operations, while improved simulation procedures will provide more accurate estimates of potential benefits. Following these avenues of research is likely to be a worthwhile pursuit.

Appendix A

Parameter Values for Departures at BOS

The purpose of this appendix is to include all the parameter values for the network model representing Boston Logan International Airport. Note that these values are used for departures only.



Table A.1: Unimpeded Erlang rates λ_i

End →	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Start ↓																	
1		0.24															
2			0.23	0.20									0.22			0.24	0.20
3		0.14		0.46				0.22									
4					0.20				0.15				0.14				
5						0.10	0.10										
6																	
7																	
8			0.23	0.23					0.15	0.10		0.06					
9										0.15	0.15						
10											0.08						
11																	
12										0.12							
13													0.30			0.21	
14																	
15																	
16														0.22	0.23		
17		0.11															0.11

Table A.3: Exponential rate for stop times μ_i

End →	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Start ↓																	
1		.020															
2			.015	.005									.018			.014	.011
3		.020		.010				.012									
4					.015				.005				.013				
5						.015	.015										
6																	
7																	
8			.017	.01					.008	.017		.020					
9										.009	.008						
10											.010						
11																	
12										.010							
13														.020		.015	
14																	
15																	
16														.009	.012		
17		.010														.005	

Table A.4: Link stopping probability with no surface traffic, p_{0i}

End →	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Start ↓																	
1		0.20															
2			0.20	0.22									0.10			0.24	0.24
3		0.15		0.10				0.20									
4					0.25				0.31				0.10				
5						0.50	0.50										
6																	
7																	
8			0.15	0.20					0.26	0.40		0.50					
9										0.38	0.42						
10											0.25						
11																	
12										0.50							
13													0.50			0.42	
14																	
15																	
16														0.65	0.50		
17		0.15															0.75

Table A.5: Sensitivity to surface traffic, X_l

End →	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Start ↓																	
1		0.03															
2			0.03	0.07									0.03			0.14	0.05
3		0.10		0.01				0.05									
4					0.15				0.14				0.26				
5						0.45	0.45										
6																	
7																	
8			0.02	0.06					0.31	0.50		0.60					
9											0.44						
10											0.40						
11																	
12										0.40							
13														0.08		0.22	
14																	
15																	
16														0.13	0.08		
17		0.16														0.25	

Appendix B

Parameter Values for Arrivals at BOS

The purpose of this appendix is to include all the parameter values for the network model representing Boston Logan International Airport. Note that these values are used for arrivals only. Since arrival traffic is usually limited, it is not possible to get statistically significant estimates of the sensitivity of the number of stops to arrival traffic. Therefore, only the mean stopping probabilities for each link are listed in this appendix.



Table B.1: Unimpeded Erlang rates λ_l

End →	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Start ↓																	
1																	
2	0.26		0.22														
3		0.28						0.25									
4			0.25					0.23					0.35				
5				0.25									0.22				
6																	
7					0.25												
8			0.28														
9			0.17	0.20				0.15									
10								0.18				0.13					
11									0.16	0.10							
12								0.14									
13		0.31		0.30													
14													0.33				
15													0.33		0.10		
16		0.29															
17		0.24															

Table B.2: Unimpeded Erlang order n_i

End →	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Start ↓																	
1																	
2	20		15														
3		20						19									
4			11					29					19				
5				19									22				
6																	
7					23												
8			21														
9			30	23				30									
10								30				30					
11									11	5							
12								13									
13		14		15													
14													10				
15													30			5	
16		27															
17		15															

Table B.3: Exponential rate for stop times μ_l

End →	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Start ↓																	
1																	
2	.018		.009														
3		.015						.015									
4			.020					.019					.012				
5				.014									.013				
6																	
7					.019												
8			.010														
9			.017	.014				.021									
10								.033				.010					
11									.011	.012							
12								.015									
13		.030		.007													
14													.018				
15													.015			.014	
16		.014															
17		.014															

Table B.4: Average link stopping probability

End →	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Start ↓																	
1																	
2	0.18		0.15														
3		0.10						0.14									
4			0.21					0.16					0.10				
5				0.12									0.11				
6																	
7					0.17												
8			0.15														
9			0.18	0.21				0.17									
10								0.18				0.46					
11									0.22	0.31							
12								0.14									
13		0.08		0.18													
14													0.07			0.11	
15													0.13				
16		0.18															
17		0.41															

Bibliography

- [1] Global Aviation Industry, “A sustainable flight path towards reducing emissions,” in *UN-FCCC Climate Talks*, (Doha, Qatar), November 2012.
- [2] “Revolutionising air traffic management,” tech. rep., Air Transport Action Group, Geneva, Switzerland, November 2012.
- [3] Federal Aviation Administration, “The operations network.” <http://aspm.faa.gov/OPSNET>. Accessed February 2013.
- [4] Federal Aviation Administration, “Aviation System Performance Metrics database.” <http://aspm.faa.gov/main/aspm.asp>. Retrieved December 2010.
- [5] R. de Neufville and A. Odoni, *Airport Systems: Planning, Design, and Management*. McGraw-Hill, 2003.
- [6] Y. Jung, “Fuel consumption and emissions from airport taxi operations,” in *Green Aviation Summit*, (Moffett Field, CA), September 2010.
- [7] C. Cros and C. Frings, “Alternative taxiing means - engines stopped.” Presented at the Airbus workshop on “Alternative taxiing means - engines stopped”, November 2008.
- [8] I. Simaiakis and H. Balakrishnan, “Impact of congestion on taxi times, fuel burn and emissions at major airports,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2184, pp. 22–30, December 2010.
- [9] US General Accounting Office, “Free flight tools show promise, but implementation challenges remain,” in *Report to Congressional Requesters*, August 2001.
- [10] D. Bertsimas and S. Stock-Patterson, “The air traffic flow management problem with enroute capacities,” *Operations Research*, vol. 46, pp. 406–422, May 1998.
- [11] L. Le, G. Donohue, and C.-H. Chen, “Auction-based slot allocation for traffic demand management at Hartsfield Atlanta International Airport: A case study,” *Transportation Research Record, Journal of the Transportation Research Board*, vol. 1888, pp. 50–58, 2004.
- [12] V. Vaze and C. Barnhart, “A multi-stakeholder evaluation of strategic slot allocation schemes under airline frequency competition,” in *USA/Europe Air Traffic Management Research and Development Seminar*, (Berlin, Germany), June 2011.

- [13] E. Gilbo, "Airport capacity: representation, estimation, optimization," *IEEE Transactions on Control Systems Technology*, vol. 1, pp. 144–154, September 1993.
- [14] D. Gillen and A. Lall, "Developing measures of airport productivity and performance: an application of data envelopment analysis," *Transportation Research Part E*, vol. 33, no. 4, pp. 261–273, 1997.
- [15] E. Pels, P. Nijkamp, and P. Rietveld, "Inefficiencies and scale economies of European airport operations," *Transportation Research Part E*, vol. 39, no. 5, pp. 341–361, 2003.
- [16] V. Ramanujam and H. Balakrishnan, "Estimation of arrival-departure capacity tradeoffs in multi-airport systems," in *Conference on Decision and Control*, (Shanghai, China), December 2009.
- [17] N. Pyrgiotis, K. M. Malone, and A. Odoni, "Modelling delay propagation within an airport network," *Transportation Research Part C*, pp. 60–75, February 2013.
- [18] M. Matthews, M. Wolfson, R. DeLaura, J. Evans, C. Reiche, H. Balakrishnan, and D. Michalek, "Measuring the uncertainty of weather forecasts specific to air traffic management operations," in *Special Symposium on Weather-Air Traffic Management Integration*, (Phoenix, AZ), January 2009.
- [19] 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.
- [20] C. E. van Daalen and T. Jones, "Fast conflict detection using probability flow," *Automatica*, vol. 45, no. 8, pp. 1903–1909, 2009.
- [21] A. Alonso-Ayuso, L. F. Escudero, and F. J. Martin-Campo, "A mixed 0-1 nonlinear optimization model and algorithmic approach for the collision avoidance in ATM: velocity changes through a time horizon," *Computers & Operations Research*, vol. 39, no. 12, pp. 3136–3146, 2012.
- [22] I. Simaiakis and H. Balakrishnan, "Queuing models of airport departure processes for emissions reduction," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, (Chicago, IL), August 2009.
- [23] H. Idris, J.-P. Clarke, R. Bhuva, and L. Kang, "Queueing model for taxi-out time estimation," *Air Traffic Control Quarterly*, vol. 10, pp. 1–22, January 2002.
- [24] F. Carr, A. Evans, J.-P. Clarke, and E. Feron, "Modeling and control of airport queueing dynamics under severe flow restrictions," in *American Control Conference*, (Anchorage, AL), May 2002.
- [25] 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.
- [26] "Free flight implementation," tech. rep., Task Force 3, RTCA Inc., Washington, DC, 1995.

- [27] J. Košecká, C. Tomlin, G. Pappas, and S. Sastry, “Generation of conflict resolution maneuvers for air traffic management,” in *International Conference on Intelligent Robots and Systems*, (Grenoble, France), September 1997.
- [28] P. K. Menon, G. D. Sweriduk, and B. Sridhar, “Optimal strategies for free-flight air traffic conflict resolution,” *Journal of Guidance, Control and Dynamics*, vol. 22, no. 2, pp. 202–211, 1999.
- [29] A. Bicchi and L. Pallottino, “On optimal cooperative conflict resolution for air traffic management systems,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 4, pp. 221–232, 2000.
- [30] J. Hu, M. Prandini, A. Nilim, and S. Sastry, “Optimal coordinated maneuvers for three dimensional aircraft conflict resolution,” *Journal of Guidance, Control and Dynamics*, vol. 25, no. 5, pp. 888–900, 2002.
- [31] P. K. Menon, G. D. Sweriduk, and K. D. Bilimoria, “New approach for modeling, analysis, and control of air traffic flow,” *AIAA Journal of Guidance, Control and Dynamics*, vol. 27, pp. 737–744, September-October 2004.
- [32] P. K. Menon, G. D. Sweriduk, T. Lam, G. M. Diaz, and K. D. Bilimoria, “Computer-aided Eulerian air traffic flow modeling and predictive control,” *AIAA Journal of Guidance, Control and Dynamics*, vol. 29, pp. 12–19, January-February 2006.
- [33] A. Bayen, R. Raffard, and C. Tomlin, “Adjoint-based control of a new Eulerian network model of air traffic flow,” *IEEE Transactions on Control Systems Technology*, vol. 14, pp. 804–818, September 2006.
- [34] C. Brinton, J. Krozel, B. Capozzi, and S. Atkins, “Improved taxi prediction algorithms for the Surface Management System,” in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, (Monterey, CA), August 2002.
- [35] H. Balakrishnan and Y. Jung, “A framework for coordinated surface operations planning at Dallas-Fort Worth International Airport,” in *AIAA Guidance, Navigation, and Control Conference*, (Hilton Head, NC), August 2007.
- [36] H. Lee, I. Simaiakis, and H. Balakrishnan, “A comparison of aircraft trajectory-based and aggregate queue-based control of airport taxi processes,” in *Digital Avionics Systems Conference*, (Salt Lake City, UT), October 2010.
- [37] N. Pujet, *Modeling and Control of the Departure Process of Congested Airports*. PhD thesis, Massachusetts Institute of Technology, December 1999.
- [38] F. Carr, *Robust Decision-Support Tools for Airport Surface Traffic*. PhD thesis, Massachusetts Institute of Technology, February 2004.
- [39] H. Khadilkar and H. Balakrishnan, “Analysis of airport performance using surface surveillance data: A case study of BOS,” in *AIAA Conference on Aviation Technology, Integration and Operations*, (Virginia Beach, VA), September 2011.

- [40] H. Khadilkar and H. Balakrishnan, "Metrics to evaluate airport operational performance using surface surveillance data," *to appear in Air Traffic Control Quarterly*.
- [41] H. Khadilkar and H. Balakrishnan, "A network congestion control approach to airport departure management," in *American Control Conference*, (Montréal, Canada), June 2012.
- [42] P. Park, H. Khadilkar, H. Balakrishnan, and C. Tomlin, "Hybrid communication protocols and control algorithms for NextGen aircraft arrivals," *submitted to IEEE Transactions on Intelligent Transportation Systems*.
- [43] P. Park, H. Khadilkar, H. Balakrishnan, and C. Tomlin, "High confidence networked control for next generation air transportation systems," *submitted to IEEE Transactions on Automatic Control*.
- [44] C. Tomlin, G. Pappas, and S. Sastry, "Conflict resolution for air traffic management: A study in multiagent hybrid systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 4, pp. 509–521, 1998.
- [45] Z.-H. Mao, E. Feron, and K. Bilimoria, "Stability and performance of intersecting aircraft flows under decentralized conflict avoidance rules," *IEEE Transactions on Intelligent Transportation Systems*, vol. 2, no. 2, pp. 101–109, 2001.
- [46] H. Erzberger and K. Heere, "Algorithm and operational concept for resolving short-range conflicts," *Institution of Mechanical Engineers, Journal of Aerospace Engineering*, vol. 224, pp. 225–243, 2009.
- [47] J. K. Kuchar and L. C. Yang, "A review of conflict detection and resolution modeling methods," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 4, pp. 179–189, 2000.
- [48] J. Kuchar and A. Drumm, "The traffic-alert and collision avoidance system," *Lincoln Laboratory Journal*, vol. 16, no. 2, pp. 277–296, 2007.
- [49] "Introduction to TCAS II version 7.1," tech. rep., Federal Aviation Administration, Washington, DC, 2011.
- [50] "Annex 10 - volume 4," tech. rep., International Civil Aviation Organization, Montréal, Canada, 2007.
- [51] F. Netjasov, A. Vidosavljevic, V. Tomic, and H. Blom, "Systematic validation of a mathematical model of ACAS operations for safety assessment purposes," in *USA/Europe Air Traffic Management Research and Development Seminar*, (Berlin, Germany), June 2011.
- [52] M. Prandini, J. Lygeros, A. Nilim, and S. Sastry, "A probabilistic framework for aircraft conflict detection," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, (Portland, OR), August 1999.
- [53] H. Swenson, J. Thipphavong, A. Sadovsky, L. Chen, C. Sullivan, and L. Martin, "Design and evaluation of the terminal area precision scheduling and spacing system," in *USA/Europe Air Traffic Management Research and Development Seminar*, (Berlin, Germany), June 2011.

- [54] D. Bertsimas, M. Frankovich, and A. Odoni, "Optimal selection of airport runway configurations," *Operations Research*, vol. 59, pp. 1407–1419, December 2011.
- [55] H. Erzberger, "CTAS: computer intelligence for air traffic control in the terminal area," tech. rep., National Aeronautics and Space Administration, Moffett Field, CA, 1992.
- [56] D. Denery and H. Erzberger, "The Center-TRACON Automation System: simulation and field testing," tech. rep., National Aeronautics and Space Administration, Moffett Field, CA, 1995.
- [57] D. Isaacson, J. Robinson, H. Swenson, and D. Denery, "A concept for robust, high density terminal air traffic operations," in *AIAA Aviation Technology, Integration and Operations Conference*, (Fort Worth, Texas), September 2010.
- [58] H. Idris, B. Delcaire, I. Anagnostakis, W. Hall, N. Pujet, E. Feron, R. 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.
- [59] 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.
- [60] Federal Aviation Administration, *NextGen Implementation Plan*, March 2012.
- [61] I. Simaiakis, H. Khadilkar, H. Balakrishnan, T. G. Reynolds, R. 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.
- [62] I. Simaiakis and H. Balakrishnan, "Dynamic control of airport departures: algorithm development and field evaluation," in *American Control Conference*, (Montréal, Canada), June 2012.
- [63] 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.
- [64] 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.
- [65] A. Nakahara, T. Reynolds, T. White, and R. Dunskey, "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.
- [66] S. Stroiney, H. Khadilkar, B. Levy, and H. Balakrishnan, "Assessing the impacts of the JFK ground management program," in *Digital Avionics Systems Conference (submitted)*, (Syracuse, NY), October 2013.

- [67] Y. Jung, T. Hoang, J. Montoya, G. Gupta, W. Malik, and L. Tobias, "Performance evaluation of a surface traffic management tool for Dallas/Fort Worth International Airport," in *USA/Europe Air Traffic Management Research and Development Seminar*, (Berlin, Germany), June 2011.
- [68] D. Böhme, "Tactical departure management with the Eurocontrol/DLR DMAN," in *USA/Europe Air Traffic Management Research and Development Seminar*, (Baltimore, MD), June 2005.
- [69] M. Schaper, G. Tsoukala, R. Stavрати, and N. Papadopoulos, "Departure flow control through takeoff sequence optimisation: Setup and results of trials at Athens airport," in *Digital Avionics Systems Conference*, (Seattle, WA), October 2011.
- [70] "European Airport Collaborative Decision Making (A-CDM)." <http://www.euro-cdm.org>.
- [71] E. Lagios, "Airport CDM - lessons learnt & challenges ahead," in *META-CDM workshop, European Community's Seventh Framework Programme*, (London, England), January 2013.
- [72] E. Feron, R. 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, Massachusetts Institute of Technology, December 1997.
- [73] I. Simaiakis, M. Sandberg, H. Balakrishnan, and R. J. Hansman, "Design, testing and evaluation of a pushback rate control strategy," in *International Conference on Research in Air Transportation*, (Berkeley, CA), May 2012.
- [74] S. Ravizza, J. Atkin, M. Maathuis, and E. Burke, "A combined statistical approach and ground movement model for improving taxi time estimations at airports," *Journal of the Operational Research Society*, pp. 1–14, October 2012.
- [75] L. Ford and D. Fulkerson, "Maximal flow through a network," *Canadian Journal of Mathematics*, vol. 8, 1956.
- [76] P. Elias, A. Feinstein, and C. Shannon, "A note on the maximum flow through a network," *IRE Transactions on Information Theory*, vol. 2, pp. 117–119, December 1956.
- [77] L. Ford and D. Fulkerson, *Flows in networks*. Princeton University Press, 1962.
- [78] R. Ahuja, T. Magnanti, and J. Orlin, *Network flows: theory, algorithms and applications*. Upper Saddle River, NJ: Prentice Hall, 1993.
- [79] M. Beckman, C. McGuire, and C. Winsten, *Studies in the economics of transportation*. New Haven, CT: Yale University Press, 1956.
- [80] D. Boyce, "Forecasting travel on congested urban transportation networks: Review and prospects for network equilibrium models," *Networks and Spatial Economics*, vol. 7, no. 2, pp. 99–128, 2007.
- [81] P. Ferrari, "Capacity constraints in urban transport networks," *Transportation Research Part B*, vol. 31, no. 4, pp. 291–301, 1997.

- [82] H. Yang, M. Bell, and Q. Meng, "Modeling the capacity and level of service of urban transportation networks," *Transportation Research Part B*, vol. 34, no. 4, pp. 255–275, 2000.
- [83] L. Adacher, M. Flamini, and G. Nicosia, "Decentralized algorithms for multiple path routing in urban transportation networks," in *Symposium on Transportation Analysis*, (Phuket, Thailand), June 2007.
- [84] E. Nikolova, M. Brand, and D. Karger, "Optimal route planning under uncertainty," in *International Conference on Automated Planning and Scheduling*, (Lake District, England), June 2006.
- [85] E. Nikolova and D. Karger, "Route planning under uncertainty: The Canadian traveller problem," in *Conference on Artificial Intelligence*, (Chicago, IL), July 2008.
- [86] F. Paganini, J. Doyle, and S. Low, "Scalable laws for stable network congestion control," in *Conference on Decision and Control*, (Orlando, FL), December 2001.
- [87] R. Johari and D. Tan, "End-to-end congestion control for the internet: delays and stability," *IEEE/ACM Transactions on Networking*, vol. 9, pp. 818–832, December 2001.
- [88] S. Low, F. Paganini, J. Wang, S. Adlakha, and J. Doyle, "Dynamics of TCP/RED and a scalable control," in *INFOCOM*, (New York, NY), June 2002.
- [89] Z. Wang and F. Paganini, "Global stability with time-delay in network congestion control," in *Conference on Decision and Control*, (Las Vegas, NV), December 2002.
- [90] D. Katabi, M. Handley, and C. Rohrs, "Congestion control for high bandwidth-delay product networks," in *SIGCOMM*, (Pittsburgh, PA), August 2002.
- [91] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Transactions on Automatic Control*, vol. 37, pp. 1936–1948, December 1992.
- [92] L. Tassiulas, "Scheduling and performance limits of networks with constantly changing topology," *IEEE Transactions on Information Theory*, vol. 43, pp. 1067–1073, May 1997.
- [93] A. Eryilmaz and R. Srikant, "Joint congestion control, routing, and MAC for stability and fairness in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, pp. 1514–1524, August 2006.
- [94] R. Srikant, *The mathematics of internet congestion control*. Boston, MA: Birkhäuser, 2004.
- [95] M. Welzl, *Network Congestion Control*. Chichester, England: John Wiley and Sons, Ltd, 2005.
- [96] P. Kawka and A. Alleyne, "Stability and feedback control of wireless networked systems," in *American Control Conference*, (Portland, OR), June 2005.
- [97] A. Eryilmaz and R. Srikant, "Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control," in *INFOCOM*, (Miami, FL), March 2005.

- [98] M. Neely, E. Modiano, and C.-P. Li, "Fairness and optimal stochastic control for heterogeneous networks," *IEEE/ACM Transactions on Networking*, vol. 16, pp. 396–409, April 2008.
- [99] J. Perkins and P. Kumar, "Stable, distributed, real-time scheduling of flexible manufacturing/assembly/disassembly systems," *IEEE Transactions on Automatic Control*, vol. 34, pp. 139–148, February 1989.
- [100] P. Kumar and T. Seidman, "Dynamic instabilities and stabilization methods in distributed real-time scheduling of manufacturing systems," *IEEE Transactions on Automatic Control*, vol. 35, pp. 289–298, March 1990.
- [101] S. Gershwin, "Design and operation of manufacturing systems: control- and system-theoretical models and issues," in *American Control Conference*, (Albuquerque, NM), June 1997.
- [102] S. B. Gershwin, "Design and operation of manufacturing systems: the control-point policy," *IIE Transactions*, vol. 32, no. 10, pp. 891–906, 2000.
- [103] H. Papadopoulos and C. Heavey, "Queueing theory in manufacturing systems analysis and design: A classification of models for production and transfer lines," *European Journal of Operational Research*, vol. 92, no. 1, pp. 1–27, 1996.
- [104] Sensis Corporation, East Syracuse, NY, *ASDE-X brochure - ASDE-X 10/05.qxd*, 2008.
- [105] M. Sandberg, "Applications of ASDE-X data to the analysis of airport surface operations," Master's thesis, Massachusetts Institute of Technology, 2012.
- [106] H. Khadilkar and H. Balakrishnan, "A multi-modal unscented Kalman filter for inference of aircraft position and taxi mode from surface surveillance data," in *AIAA Conference on Aviation Technology, Integration and Operations*, (Virginia Beach, VA), September 2011.
- [107] RTCA, Inc., *Minimum Aviation System Performance Standard for Automatic Dependent Surveillance Broadcast (ADS-B)*, 2002.
- [108] 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.
- [109] H. Khadilkar and H. Balakrishnan, "Network congestion control of airport surface operations," *to appear in Journal of Guidance, Control and Dynamics*.
- [110] H. Khadilkar and H. Balakrishnan, "Control of aircraft pushbacks at an airport using a dynamic programming formulation," in *AIAA Guidance, Navigation, and Control Conference*, (Minneapolis, MN), August 2012.
- [111] H. Khadilkar and H. Balakrishnan, "Optimal control of airport operations with gate capacity constraints," in *European Control Conference*, (Zurich, Switzerland), July 2013.
- [112] P. Burgain, *On the control of aircraft departure operations*. PhD thesis, Georgia Institute of Technology, November 2010.

- [113] S. Kullback and R. Leibler, "On information and sufficiency," *Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.
- [114] A. Barnett, G. Paull, and J. Iadeluca, "Fatal US runway collisions over the next two decades," *Air Traffic Control Quarterly*, vol. 8, 2000.
- [115] R. Clewlow and D. Michalek-Pfeil, "Logan control tower: Controller positions, processes and decision support systems," tech. rep., Massachusetts Institute of Technology, Cambridge, MA, 2010.
- [116] R. Gallager, *Discrete Stochastic Processes*. Springer, 1996.
- [117] G. Heineman, G. Pollice, and S. Selkow, *Algorithms in a Nutshell*. Cambridge, MA: Oreilly Media, 2008. Chapter 8: Network flow algorithms.
- [118] D. Bertsekas, *Dynamic Programming and Optimal Control, Volume I*. Cambridge, MA: Athena Scientific, 3 ed., 2005.
- [119] E. Lester and R. Hansman, "Benefits and incentives for ADS-B equipage in the National Airspace System," tech. rep., Massachusetts Institute of Technology, Cambridge, MA, 2007.
- [120] K. Sampigethaya, R. Poovendran, S. Shetty, T. Davis, and C. Royalty, "Future e-enabled aircraft communications and security: The next 20 years and beyond," *Proceedings of the IEEE*, vol. 99, no. 11, pp. 2040–2055, 2011.
- [121] 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.
- [122] W. Karush, "Minima of functions of several variables with inequalities as side constraints," Master's thesis, University of Chicago, 1939.
- [123] H. Kuhn and A. Tucker, *Nonlinear programming*. Berkeley, CA: University of California Press, 1951.
- [124] H. Khadilkar, "Analysis and modeling of airport surface operations," Master's thesis, Massachusetts Institute of Technology, 2011.
- [125] C.-L. Wu and R. Caves, "Modelling and optimization of aircraft turnaround time at an airport," *Transportation Planning and Technology*, vol. 27, no. 1, pp. 47–66, 2004.
- [126] A. D. Donaldson, "Improvement of terminal area capacity in the New York airspace," Master's thesis, Massachusetts Institute of Technology, 2011.
- [127] S. Smith and E. Gilbo, "Analysis of uncertainty in ETMS aggregate demand predictions," tech. rep., Volpe National Transportation Systems Center, Cambridge, MA, 2005.