Modeling and Control of Queuing Networks: Applications to Airport Surface Operations

by

Sandeep Badrinath

Bachelor of Technology, Indian Institute of Technology Madras (2015)
Master of Technology, Indian Institute of Technology Madras (2015)
Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Aeronautics and Astronautics
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June 2021

Author . . . .
Department of Aeronautics and Astronautics
May 1st, 2021

Certified by . . . .
Hamsa Balakrishnan
William E. Leonhard (1940) Professor of Aeronautics and Astronautics
Thesis Supervisor

Certified by . . . .
Amedeo Odoni
T. Wilson Professor Emeritus of Aeronautics and Astronautics
Doctoral Committee Member

Certified by . . . .
Tom G. Reynolds
Group Leader, Air Traffic Control Systems, MIT Lincoln Laboratory
Doctoral Committee Member

Accepted by . . . .
Zoltan Spakovszky
Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

Modeling and Control of Queuing Networks: Applications to Airport Surface Operations

by

Sandeep Badrinath

Submitted to the Department of Aeronautics and Astronautics
on May 1st, 2021, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Aeronautics and Astronautics

Abstract

The significant growth in air traffic over the past few decades has led to increased congestion at major airports worldwide. Airport congestion results in increased flight delays, fuel burn and emissions. There is consequently a need to accurately model airport traffic operations and to design control algorithms that mitigate airport congestion.

In this thesis, we propose a new class of queuing network models of airport surface operations that are capable of capturing congestion at multiple locations, and that can account for the time-varying nature of demand and capacity. The proposed queuing models are based on point-wise steady state approximation that result in a simple ordinary differential equation representation for the dynamics of the ensemble queue length. Further, the models can account for propagation delays between servers in the network, and handle general service time distributions, overcoming some of the limitations of the traditional probabilistic queuing models. The queuing models are developed, adapted, and validated using actual operational data from several major airports. These models also allow us to apply techniques from reachability analysis to better understand the performance of queuing networks.

The second part of this thesis focuses on the development of airport congestion control algorithms using the proposed queuing network models. The dynamical systems representation of the queuing process allows us to use optimal and robust control techniques to regulate the queue length, and to obtain theoretical guarantees for certain special cases. We compare our algorithms with NASA's logic that was recently field-tested, using stochastic simulations. We also investigate the impact of uncertainty in airline-supplied estimates of traffic demand on the efficacy of congestion control algorithms, and quantify the benefits of reducing this uncertainty. Finally, we note that the modeling and control techniques for queuing networks developed in this thesis have broad applicability in other contexts beyond airport surface operations.

Thesis Supervisor: Hamsa Balakrishnan
Title: William E. Leonhard (1940) Professor of Aeronautics and Astronautics

Acknowledgments

First and foremost, I would like to thank my research advisor, Hamsa Balakrishnan, for giving me a wonderful opportunity to pursue a PhD at MIT and for mentoring me over the last six years. I feel very fortunate to have had such a kind and humble advisor. She took a chance on me when I was working on fluid mechanics, and let me pursue something different. Many thanks to her for believing in me and supporting me throughout my PhD. I greatly enjoyed my research because she allowed me to explore new topics, and she gave me opportunities to collaborate and mentor students. At the end of my PhD, I feel that I have not only become a better researcher, but I have also become a better person. Thank you, Hamsa!

I am very glad that Amedeo Odoni and Tom Reynolds agreed to serve on my thesis committee. I am truly thankful to both of them. Although I only had a few meetings with Amedeo, I will never forget those meetings because he always listened to my research with at most interest, provided valuable feedback and wonderful encouragement. It was great to see his passion and enthusiasm. I also had an excellent opportunity to work with Tom from the very beginning of my PhD through various collaborations with MIT Lincoln Laboratory. A good part of my thesis would not have been possible without the help from Tom. Thank you so much for providing an industry perspective and making this thesis more impactful.

I would like to thank Alexandre Jacquillat and Husni Idris for agreeing to be my thesis readers, and for providing valuable feedback. Before coming to MIT, I had little knowledge about controls, which eventually became an important topic of my thesis. I am grateful to all the faculty who taught controls at MIT. I also would like to thank my undergraduate and master's advisor, Sunetra Sarkar, for being a wonderful mentor and setting up the foundation for my research career.

I was fortunate to have had several amazing collaborators both within and outside MIT. I would like to express my deepest appreciation to Emily Joback, Yashovardhan Chati, Max Li, Ji Ma, Dun Yuan Tan, Nadia Dimitrova, Azreen Zaman, James Abel, Kai Ling Neo, Gerard Degue, Yanjun Wang, Melanie Sandberg, Daniel Delahaye, and Luis Garcia. Special thanks to Emily from MIT Lincoln Laboratory for collaborating on the FAA projects and for sharing various datasets on airport operations that were an integral part of this thesis. I would like to acknowledge Yashovardhan for being the senior grad student in the lab, and for guiding me throughout my PhD. I also would like to recognize the assistance from Quentin Alexander, Ping Lee, and Pam Franklin for all the administrative support.

The work presented in this thesis was supported in part by the National Science Foundation, FAA, NASA, and Airport Cooperative Research Program. I would like to acknowledge the support from these funding agencies and industry collaborators.

I am grateful to my labmates who were very helpful and provided a wonderful atmosphere. The separation caused by the pandemic over the last year has shown how important they were. I will never forget the fun lunch conversations and the evening squash sessions. Much thanks to all of them: Patrick, Jacob, Yashovardhan, Karthik, Jackie, Joao, Kai Ling, Takuya, Baris, Max, Lauren, Gerard, and Chris.

I also would like to thank my other friends for all their support and fun trips. There were many individuals who were a part of this incredible journey, I would like to mention a few close ones: Srinivasan, Abhiram, Suhas, and Karthik. In particular, I am very grateful to my friend Karthik, who has been of much help throughout my undergrad and grad school. I also would like to thank the Toastmasters club at MIT for providing a fantastic opportunity to be a better speaker and leader. Special thanks to Sterling Crockett and Robert Forbes, who were deeply passionate and took interest in mentoring me.

Finally, a big thanks to my parents and extended family for all their support, wishes, and for letting me do whatever I want. I wouldn't be here today without their help. I don't have words to express my gratitude to my sister Soumya and her husband Suraj. They have been there for me from the moment I first landed in Boston six years back. They are some of the kindest people whom I have met and thanks to them that I have never felt away from home all these years. I was also fortunate to have had two furry friends (Coco and Pluto) during my PhD. Those two have kept my spirits high and they always put a smile on my face.

Contents

1 Introduction 21

1.1 Motivation 21

1.2 Background and Related Literature 22

1.2.1 Modeling airport surface traffic 23

1.2.2 Departure metering 25

1.2.3 Queuing network models 28

1.2.4 Control algorithms for queuing networks 30

1.3 Contributions of this thesis 32

1.3.1 Mathematical models of airport surface traffic movements 33

1.3.2 Airport congestion control in the presence of uncertainty 35

1.3.3 Robust control of queuing networks 36

1.3.4 Reachability analysis for analyzing the performance of queuing networks 37

1.3.5 Characterizing airport operations using emerging aviation data sources 38

1.4 Organization of this thesis 40

2 Queuing network model of the airport surface 43

2.1 Introduction 43

2.2 Queuing models of airport surface operations 44

2.2.1 Queuing network models of EWR, DFW, and CLT 44

2.2.2 Models of queue dynamics 47

2.2.3 Application of analytical queuing models to the airport surface 54

2.2.4 Service rates of the servers 56

2.2.5 Predictive performance of the queuing model 62
2.2.6 Applicability to other airports 67
2.3 Integration of surface and airspace models for CLT 68
2.4 Comparison to microscopic simulation 71
2.4.1 Microscopic simulator 71
2.4.2 Performace of the microscopic simulations 73
2.4.3 Discussion on the modeling philosophy 75
2.4.4 Discussion on the Poisson assumption for queuing networks 75

3 Congestion control for airport surface operations 79
3.1 Introduction 79
3.2 Control algorithms for departure metering 81
3.2.1 Airspace Technology Demonstration-2 (ATD-2) logic 81
3.2.2 Simulation environment for evaluation 82
3.2.3 Departure metering benefits with accurate EOBTs 83
3.3 Impact of EOBT uncertainty 87
3.3.1 Evaluating EOBT uncertainty 87
3.3.2 Parametric analysis of the impacts of EOBT uncertainty 91
3.3.3 An optimal control approach for departure metering 95
3.4 Discussions 99
3.4.1 Fairness of departure metering 99
3.4.2 Incentives to improve EOBT accuracy 100
3.4.3 Other considerations for departure metering 101

4 Robust control for queuing networks: Applications to airport operations 103
4.1 Introduction 103
4.2 Control for a single queue 105
4.2.1 Case without propagation delays 105
4.2.2 Case with propagation delays 110
4.2.3 Numerical experiments 115

4.3 Control of queues in tandem 119
4.3.1 Controller for dynamics without propagation delays 120
4.3.2 Controller for tandem queues with propagation delays 128
4.4 Application to airport surface operations 129
4.4.1 Departure metering: controlling departure queue length 131
4.5 Evaluating the performance of departure metering 135
4.5.1 Benefits of departure metering 135
4.5.2 Comparison to other departure metering approaches 136
4.6 Discussions 138
4.6.1 Correcting for the prediction errors 138
4.6.2 Multiple arrival sources into the queueing network 140

5 Reachability analysis for queuing networks 143
5.1 Introduction 143
5.2 Reachability analysis 144
5.3 Mixed-monotonicity method 146
5.4 Application for queuing networks 148
5.5 Partitioning the initial set 151

6 Data analytics for airport operations: Beyond queuing models 155
6.1 Introduction 155
6.2 Environmental impact assessment of airport operations 156
6.2.1 Aircraft trajectory and fuel burn 158
6.2.2 Detecting acceleration events 159
6.2.3 Fuel burn prediction using aircraft trajectories 161
6.2.4 Fuel burn estimation using airport surface surveillance data 163
6.3 Extracting operational information from ATC voice communications 166
6.3.1 Related work 167
6.3.2 ASR model for ATC communications 168
6.3.3 Extracting operational information 169

6.3.4 Evaluating transcription accuracy in the absence of ground truth data 174

7 Conclusions 177

7.1 Summary 177

7.2 Practical implementation: opportunities and challenges 179

7.3 Opportunities for future research 181

7.3.1 Improvements to departure metering algorithm 182

7.3.2 Emergence of advanced air mobility 183

7.3.3 Data analytics to evaluate airport operations 184

7.3.4 Queuing network representations of other systems 186

A Queuing model and departure metering for Paris airport (CDG) 191

B Multi-model smoother for aircraft trajectories 197

C Automatic speech recognition model 209

List of Figures

1-1 Snapshot of traffic movements at CLT indicating queuing at different regions. Departures and arrivals are represented using black and white triangles, respectively. 26

1-2 Airport surface traffic models and congestion management techniques. 28

1-3 Queuing models and control algorithms 32

2-1 Snapshots of traffic movements at EWR, DFW and CLT, and the corresponding queuing representations under the most frequently-used configurations. The primary departure and arrival runways are indicated in the figure using blue and red arrows, respectively. Departures and arrivals are represented as black and white triangles, respectively. 45

2-2 Number of pushbacks (per 15 min), declared departure capacity (per 15 min) and sum of the queue lengths near the two departure runways for a typical day at CLT (05/22/2016). 47

2-3 The dependence of the C parameter on C_v and x_m. 50

2-4 Comparison between the approximation for $G(x)$ and the actual value ($C_v = 0.2, x_m = 10, C = 1.5$). 50

2-5 Comparison of the mean queue lengths obtained from simulations and the proposed analytical queuing model for an $M_t/M/1$ queue. Note the different x-axis limits. 51

2-6 Comparison of the mean queue lengths obtained from simulations and the proposed analytical queuing model for an $M_t/E_{10}/1$ queue. Note the different x-axis limits. 52

2-7 Mean service rate of ramp and runway servers for CLT-NF. 58

2-8 Service time distribution for departure runway server conditioned on no landings in the 5 min window for runways at three different airports. 59

2-9 Distribution of the time between successive crossings, when there is pressure on the runway crossing server for CLT-NF. 60

2-10 Service time of the departure runway server 36R at CLT in VMC conditions. 61

2-11 Queue length predictions for taxi-out ramp and runway queue for a typical day (Jun 25, 2016) at CLT. 63

2-12 Queue length predictions for taxi-in ramp and runway crossing queue for a typical day (Jun 25, 2016) at CLT. 63

2-13 Taxi times averaged over 15 min interval for a typical day at CLT in North-Flow configuration (Jun 25, 2016). 64

2-14 Comparison of model predictions and observed queue lengths on a typical good weather day at DFW (07/20/2016). 65

2-15 Taxi-out times averaged over 15-min intervals on 07/20/2016 at DFW. . . . 65

2-16 Comparison of the predicted and actual taxi-out time distributions. 67

2-17 High-level system diagram of the integrated surface-airspace model for predicting gate to final departure fix transit times (Image courtesy of Max Z. Li [24]). 69

2-18 Airport layout and corresponding node-link representation for CLT-NF. . . . 72

2-19 Plot of distance vs unimpeded time for departures. 73

2-20 Safe separation rules applied in the simulator. 73

2-21 Taxi-times averaged over 15-min windows for a typical day at CLT-NF (07/12/2015) 74

2-22 Illustration of queues in series and service time distribution. 76

2-23 Comparison of the mean queue lengths obtained from simulations and the proposed analytical queuing network model for different service time distributions. $\lambda(t) = 1 + 0.5\sin(0.1t), \mu_1 = \mu_2 = 2$. 77
2-24 Comparison of the mean queue lengths obtained from simulations and the proposed analytical queuing network model for different service time distributions. $\lambda(t) = 1.9 + 0.5\sin(0.1t), \mu_1 = \mu_2 = 2$. 77

3-1 Illustration of the ATD-2 logic for departure metering. 82
3-2 Metering simulation results for different excess queue time buffer values. The plots show averaged values over a 15 min time window for a typical day at DFW in North-Flow configuration (07/20/2016). 84
3-3 Departure metering statistics for DFW-SF. The statistics were calculated over 9,212 flights in a 15 day period. 84
3-4 Cumulative distribution of the time between the actual pushback time and the next arrival at a gate. 87
3-5 Message time of the EOBT information relative to the EOBT time (Airline A). 88
3-6 Time between EOBT updates for lookahead time less than 40 min (Airline A). 89
3-7 Distribution of number of EOBT updates recieved by a flight which had its EOBT published (Airline A). 89
3-8 EOBT error distribution for different lookahead horizons (Airline A). 90
3-9 EOBT information update statistics for Airline B at EWR. 91
3-10 Optimal excess queue time buffer and taxi-out time reduction for DFW-SF. . 93
3-11 Annualized taxi-out time reduction and fuel burn reduction per flight using the ATD-2 logic for different levels of EOBT uncertainty. 94
3-12 Relative frequency of the number of updates for the release time for the flights that were held at the gate using the optimal control policy. 99

4-1 Mean queue length for $M_t/M/1$ queue ($\lambda/\mu = 0.9 + 0.45\sin(0.1t);\ \mu = 2$). . 104
4-2 Queue length obtained when the delay-free controller is applied on the time-delay system. Parameters: $x_d = 8 + 2\sin(0.1t))$, $\tau = 5,\ C = 1,\ \mu = 1$. 110

4-3 Output queue length when there are no model uncertainties. 117

4-4 Tracking accuracy when the actual service rate (μ_a) is different from the nominal service rate assumed in the model ($\mu = 1$). 117

4-5 Queue length from the queuing simulation with exponential service time distribution. 117

4-6 Schematic of two queues in tandem. 119

4-7 Controller performance for tracking the length of the second queue. 123

4-8 Single realization and statistics over multiple realizations of the queue length. Simulation parameters: $\mu_1 = 2,\ \mu_2 = 1,\ \lambda = 23,\ \eta = 0.5,\ \Delta t = 0.1, x_{2,d} = 8 + 2\sin(0.1t)$ 123

4-9 Markov chain representation for tandem queue 124

4-10 Control policy and queue length obtained from simulation ($x_d = 8,\ \mu_1 = 2,\ \mu_2 = 1,\ \Delta t = 0.1,\ \gamma = 0.99$). 126

4-11 Queue length for tandem queue without propagation delays.$x_{2,d} = 8{+}2\sin(0.1t)$. Simulation parameters: $\mu_1 = 1.25,\ \mu_2 = 1,\ \lambda = 0.3,\ \eta = 0.5,\ \Delta t = 0.1$ 127

4-12 Queue length for tandem queue obtained by using a controller without considering the first queue length information in the feedback. 127

4-13 Tandem queues with delays ($\mu_1 = 2,\ \mu_2 = 1, \tau_1 = \tau_2 = 2.5, \eta = 0.1, \lambda = 0.04, x_{2,d} = 8 + 2\sin(0.1t)$). 129

4-14 Queue length for tandem queue with propagation delays. 129

4-15 Airport layout and queuing representation. 130

4-16 Variation of the mean service and demand for a typical good weather day (07/12/2015). 131

4-17 Average queue length, taxi-out time and wheels-off delay from the simulations of the departure metering strategy for a typical day (May 7, 2015). 137

4-18 Correcting for the prediction errors using recursive least squares ($\mu = 1,\ \mu_a = 1.5,\ C = C_a = 1$). 140

5-1 Illustration of reachable sets 145

5-2 Illustration for computing the interval reach set for monotonic systems . . . 146

5-3 Reachable set for the expected queue length for M/M/1 queue with uncertain mean arrival rate (λ). 149

5-4 The reachable set for the mean queue length in the network. 151

5-5 Illustration of wrapping effect: the resulting interval enclosure of the transformed interval sets can be significantly conservative (source: [7]) 152

5-6 Comparing interval enclosure with and without partitioning the initial set. . 153

5-7 Reachable set for tandem queue system at $t = 5$. 153

6-1 A sample profile of aircraft speed and total fuel flow rate during the taxi-out phase of flight for A330-343. 159

6-2 The plot on the top shows a comparison of the speed profile obtained from the smoothed estimate with the actual FDR data. The middle plot shows the normalized likelihood (mode fraction) for the differet modes. The bottom plot shows the indicator function for the acceleration event obtained from the mode fraction along with the total fuel flow rate. 161

6-3 Comparison between measurements obtained from FDR and ASDE-X data for a sample flight. 165

6-4 Statistics of word counts and WERs computed for the test set. 169

6-5 Uncertainty score of the transcription 175

7-1 Heat maps obtained by averaging values over 69 A320 taxi-out flight trajectories in LSZH. 185

7-2 (a) Reservoir storage levels as a percentage of capacity (%) over the days leading up to the floods [10], (b) Queuing network representation of the reservoir network. 189

A-1 CDG in West-flow runway configuration. 192

A-2 Number of pushbacks (per 15 min), declared departure capacity and queue length for a typical good weather day at CDG (July 10th, 2017). 192

A-3 Comparison between model predictions and data for CDG. 194

B-1 Analysis with different acceleration thresholds 205
B-2 Comparison between FDR and ASDE-X estimates for a sample trajectory. . 207

C-1 Model architecture for the ASR system. 210

List of Tables

1.1 Trade-off between airport surface models of various fidelity. 24

2.1 Taxi-out times, including the unimpeded and delay components, at EWR, DFW, and CLT, separated by ramp area (gate-to-spot) and active movement area (spot-to-runway). The statistics are computed using multiple data sources [138, 67, 68] for a three month period (May-July 2016) for DFW and EWR, and a five month period for CLT (May-July 2015; May-Jun 2016). . . 46

2.2 Error statistics of the taxi-out time predictions from the analytical queuing model on a test set. 66

2.3 Error statistics for the taxi-out time predictions for individual flights from the queuing model based on the test set for different quartiles of the actual data. 66

2.4 Error statistics for the taxi-out and taxi-in time predictions for CLT-NF, based on 7,484 departures and 8,477 arrivals in the test dataset. 67

2.5 Error statistics of the taxi-out time predictions on a test set using the queuing model. 67

2.6 Surface-airspace integrated queue model departure transit time prediction statistics. 70

2.7 Error statistics for a typical day at CLT-NF (07/12/2015) containing 672 departures and 664 arrivals. 75

3.1 Error statistics for the baseline taxi-out time predictions from the queuing simulation over a 15 day period. 83

3.2 Effects of departure metering with 20-min planning horizon without EOBT uncertainty. 86

3.3 EOBT update statistics computed for lookahead horizons $\leq$ 40 min (Airline A). 88

3.4 EOBT error statistics for different lookahead times (Airline A). 90

3.5 EOBT error statistics for Airline B and Airline C at EWR computed for different lookahead horizons. 91

3.6 Average taxi-out time reduction in minutes for different EOBT error distributions with a planning horizon of 20-min. The optimal buffer for the ATD-2 logic is indicated in brackets. 94

3.7 Annualized fuel burn savings ($\times 10^6$ kg) using the ATD-2 departure metering logic. 95

3.8 Comparison of the departure metering benefits with and without EOBT uncertainty (assuming $\sigma = 6$ min). Opt. refers to results from the optimal controller. 98

3.9 Computational times for the ATD-2 logic and the optimal control approach. 99

3.10 Mean absolute displacement of the takeoff order with and without EOBT uncertainty (assuming $\sigma = 6$ min). Opt. refers to results from the optimal control approach. 100

4.1 Comparison of simulations of departure metering approaches for CLT-NF (perfect EOBT case). 138

4.2 Comparison of simulations of departure metering approaches for CLT-NF (EOBT uncertainty, $\sigma = 6$ min). 138

6.1 Model equation for the fuel burn and the error statistics for different aircraft types. The number of data points in the test set is shown below the aircraft type in brackets. The numbers below the model equation indicate the p-value of the model coefficients, and the row below shows (R^2, MPE, MAPE). . . . 163

6.2 Error statistics computed from ASDE-X trajectories using fuel burn models developed from FDR data for A330 data set 165
6.3 Extracting call-signs from reference transcripts. 173

A.1 Error statistics using analytical queue models. 194
A.2 Error statistics for stochastic simulations of baseline. 195
A.3 Comparison of simulations of departure metering approaches for CDG. . . . 195

B.1 Event detection statistics for different acceleration thresholds (a_t) using FDR data 205
B.2 Event detection statistics for different acceleration thresholds (a_t) using ASDE-X data 207

Chapter 1

Introduction

1.1 Motivation

Airports are the critical nodes of the air transportation network. With the globalization of industries and growth in leisure travel, airports play an important role in connecting people and businesses, thereby transforming economies. The deregulation of the airline industry, new business models such as low cost carriers, and more recently, increasing incomes in emerging economies such as China and India, have all led to an increase in air traffic over the years. In 2019, there were about 38 million commercial flights that carried 4.5 billion passengers and 58 million tons of freight (35% of the total freight by value), illustrating the massive scale of operations [89]. Despite a large downturn in traffic during 2020-21 due to the COVID-19 pandemic, it is expected that the traffic will return to pre-COVID levels in a few years and then continue to grow significantly into the future [78]. Furthermore, forecasts indicate that the conventional air traffic will more than double over the next 20 years, and expect the emergence of new concepts such as advanced air mobility [4, 114]. These increases in air traffic demand have not been matched by increases in the capacities of major airports, which are largely determined by their physical infrastructures. The imbalance between capacity and demand, particularly during periods of peak traffic, has led to congestion at major airports worldwide. Moreover, delays arising at an airport can cascade to other airports because of the networked nature of air transportation system [147]. The resulting delays from airport congestion have an economic impact on airlines and passengers, as well as an environmental impact due to excessive fuel consumption and emissions. In 2018, the estimated annual cost of flight delays was $28 billion in the US alone, and aviation emissions increased by 3% compared to the previous year [6].

A straightforward approach to reduce congestion is to improve airport capacity by adding new infrastructure. However, such airport expansion can be very expensive, has long lead times, and may not be feasible in major metropolitan areas [36, 12]. Additionally, the regulatory clearance for airport expansion is often delayed because of political and environmental reasons, a recent example being the third runway at London Heathrow [15]. The long lead times of airport expansion projects can also result in the realized demand exceeding the planned capacity. On the other hand, the development of new greenfield airports can face challenges due to a lack of connectivity and high investment costs [11]. Strategic and tactical traffic management approaches are therefore needed to mitigate congestion under currently available airport capacity.

1.2 Background and Related Literature

Airport surface traffic management is important because aircraft spend around 10–30% of their flight time just taxiing at the airport, and this could contribute to 5-10% of the total fuel burn for short-to-medium range flights [54]. Different approaches to airport traffic management have been previously proposed. These approaches can be broadly classified as regulatory improvements, administrative changes, and operational improvements. Regulatory improvements correspond to changing the current air traffic rules to improve operations. One such example is wake turbulence recategorization, which involves revising the runway separation requirements between aircraft to improve airport capacity [57]. Administrative changes include efficient strategies for demand management, such as slot allocation and congestion pricing to regulate traffic [81].

Operational changes to mitigate congestion include departure metering [158], taxiway routing [61, 189], runway scheduling [25, 84], optimal runway configuration selection [29] and reducing runway occupancy time by efficient braking [161]. Taxiway routing refers to efficiently routing the aircraft on the airport surface to reduce taxi times while considering other constraints such as meeting a desired target take-off time. Complimentary to taxiway routing is runway scheduling, that refers to optimizing the sequence of aircraft using the runway to improve airport throughput. Additionally, optimizing braking profile during landing roll can significantly reduce the runway occupancy time (by $\approx 50\%$), resulting in improved airport capacity [161]. Another promising operational approach to reduce airport surface congestion is departure metering that refers to tactically holding departures at the gate during periods of congestion [158, 69, 59].

All these approaches for reducing airport congestion often require a model for the traffic movements [35, 160, 101, 52]. In addition to their use in the development of congestion control algorithms, models of the airport surface can be used to characterize airport operations, analyze new and existing procedures, and evaluate the environmental impacts of aviation. Next, we present a discussion on various airport surface traffic models.

1.2.1 Modeling airport surface traffic

Models of the airport surface can been broadly described as microscopic, mesoscopic, or macroscopic, based on their level of fidelity [146, 157]. *Microscopic* models (for example, NASA's Surface Operations Simulator and Scheduler (SOSS), or commercial tools such as SIMMOD), specify the dynamics for each aircraft as well as the detailed airport layout [185]. Model predictions require the simulation of individual aircraft trajectories, yielding high-fidelity results. The main disadvantage of microscopic models is that the predictions can be computationally intensive (especially in the presence of uncertainty), simulations can 'gridlock'[1] requiring manual intervention and resolution, and the adaptation of models to new airports and operating conditions can be challenging. In *mesoscopic* models, the taxiways and runways are represented by links with associated travel-time and service time distributions, leading to lower fidelity compared to the *microscopic* model [101]. Another

---

[1] 'gridlock' refers to the situation in which aircraft comes to completle stand-still during taxi operation because of traffic congestion in all directions

<table>
<tr><th rowspan="2">Parameter</th><th colspan="3">Airport surface models</th></tr>
<tr><th>Macroscopic</th><th>Mesoscopic</th><th>Microscopic</th></tr>
<tr><td>Level of detail (fidelity)</td><td colspan="3">⟶</td></tr>
<tr><td>Computational tractability</td><td colspan="3">⟵</td></tr>
<tr><td>Ease of adaptation across airports</td><td colspan="3">⟵</td></tr>
<tr><td>Amount of data required</td><td colspan="3">⟶</td></tr>
<tr><td>Ability to build robust decision-support tools</td><td colspan="3">⟵</td></tr>
</table>

Table 1.1: Trade-off between airport surface models of various fidelity.

variant of *mesoscopic* model that has been proposed is to split the taxiway segments into smaller cells and model the movement of traffic using cell transmission model or other similar approaches [35, 187]. The identification of these models requires the availability of large amounts of surface trajectory data in order to estimate transit time distributions of individual links under different conditions, which could be a limitation. Finally, we refer to queuing models, and other aggregate statistical models, as being *macroscopic* in nature [159, 157, 110]. These models are based on aggregate variables such as queue lengths and travel times over relatively large parts of the airport (as opposed to individual aircraft or taxiway links). Despite their lower fidelity, their simplicity makes these models amenable to fast-time simulations and easy to integrate into decision support tools. Table 1.1 gives a summary of the trade-offs between models of different fidelity. Macroscopic models (such as queuing models) are promising given that they can be easily adapted to different airports, require relatively smaller amounts of data for training, and one can use such models for robust decision-support tools.

A variety of macroscopic queuing models, ranging from deterministic to stochastic models, have been proposed for representing airport traffic movements [81, 157]. In these models, congestion hot-spots on the airport surface are represented using queues, and the taxi times are estimated to be the unimpeded times[2] plus the wait times in the queues. For the queuing process, the arrivals are typically considered to be Poisson, and the service time distributions are assumed to belong to an exponential family (such as exponential or Erlang). Such assumptions are often made to obtain analytical expressions for the evolution of the queuing

---

[2] travel times during periods of low traffic

process, although these assumptions might be invalid for certain applications. We discuss abstract queuing models in more detail in Section 1.2.3.

Queuing models for the airport surface have been used to: analyze airport operations, investigate the impact of infrastructure changes (such as additional runways), and develop decision support tools for traffic management [93, 160, 92]. They have also been used in environmental assessment tools such as FAA's Aviation Environmental Design Tool (AEDT) [66]. However, airport surface queuing models have traditionally assumed that congestion occurs exclusively near the departure runway [160, 157, 159, 17]. While this is true at some airports, at others with complex layouts such as Charlotte Douglas International airport (CLT), congestion can occur at multiple locations. Figure 1-1 shows a snapshot of traffic movements at CLT on a typical day, where we can notice queues being formed in the ramp area, near the departure runway, and the runway crossing area. This requires one to model the traffic congestion on the airport surface as a network of queues. Although queuing networks have been proposed to represent the airport surface traffic [90], a tractable queuing network model that explicitly considers the non-stationary behavior of the airport operations has not been proposed in the prior literature. We address this technical gap in this thesis.

1.2.2 Departure metering

In this section, we discuss departure metering, a promising approach to reduce airport surface congestion. In the absence of departure metering, pilots push back from the gate and start taxiing whenever ready, which often results in long queues of aircraft during periods of high departure demand. A departure metering procedure tactically holds aircraft at their gates during periods of congestion, and releases them in an appropriate manner such that the aircraft pass through smaller queues, while still maintaining runway throughput. The result is a reduction in the taxi-out time of flights (travel time from the departure gate to take-off), and fuel savings. Departure metering has the estimated potential to provide \$5.5-9.5 billion in monetary benefits at the top 35 airports in the US over a 20-year period [134], and is an integral part of airport surface management programs internationally [69, 79, 39]. In the

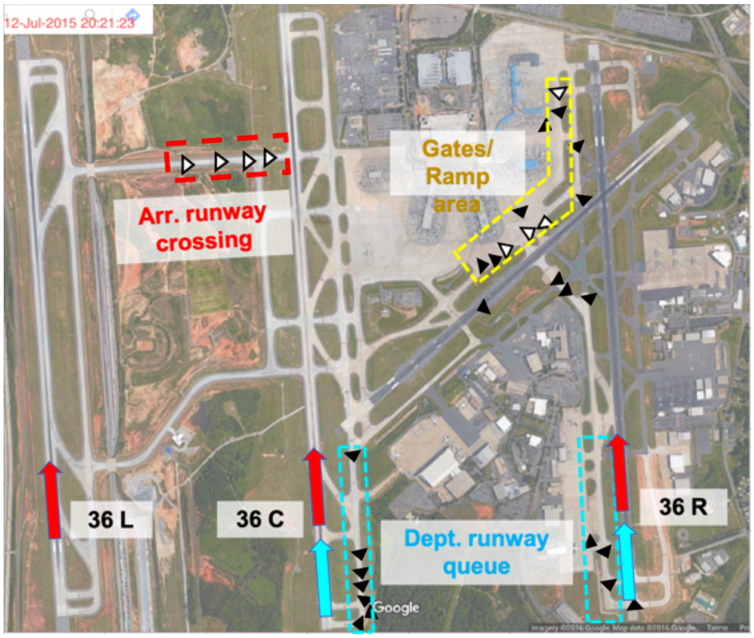


Figure 1-1: Snapshot of traffic movements at CLT indicating queuing at different regions. Departures and arrivals are represented using black and white triangles, respectively.

US, departure metering is a part of the FAA's Terminal Flight Data Manager (TFDM), a NextGen initiative [69, 62]. Although the departure metering concept originated in the late 1990s [71], there has been increased recent interest, fueled by access to more granular data (such as flight tracks), theoretical developments, technological improvements (e.g. electronic flight strips, push towards higher automation), and plans for operational deployment of departure metering at major airports.

The main goal of departure metering is to assign appropriate hold-times to maximize benefits in terms of taxi-out time reduction, while maintaining runway utilization. Several approaches to departure metering, based both on heuristics and optimization algorithms, have been previously investigated [146, 179, 160, 101, 187]. One of the earliest strategies for departure metering is a simple threshold policy, know as the N-control, that was proposed under the Departure Planner concept [71]. In the most common variant of the N-control policy, departures are held at their gates if the total number of departures taxiing to the runway is greater than a threshold value. To improve the benefits of departure metering, researchers have proposed a variety of control and optimization techniques to make the gate hold decisions. The techniques include proportional-integral feedback control [187], dynamic programming [160, 101], partially observable Markov decision processes (POMDPs) [35], robust optimization [133], and stochastic optimization [31]. The effectiveness of these approaches vary depending on the accuracy of the underlying model, robustness of the algorithm to deal with various uncertainties [121], and more importantly, whether the resulting recommended decisions are practical for real-world implementation. The taxi-out process is subject to significant uncertainties, including the time that an aircraft is ready to pushback (its pushback time), taxi routes, taxi speeds, and the separation between successive take-offs [148, 109]. Prior work on departure metering under uncertainty has generally focused on the uncertainty in taxi times and runway capacity [35, 160, 31, 133]. By contrast, demand-side uncertainty, namely, the uncertainty associated with the time at which an aircraft will be ready to pushback, has received little attention. In this thesis, we develop departure metering algorithms that are robust to these uncertainties, and evaluate their impacts.

Field demonstrations have been valuable in understanding the real-world benefits and scope for improvement of different departure metering approaches. In the US, there have been several field demonstration campaigns at major airports, such as in Boston-BOS [160], New York-JFK [169], Dallas-DFW [127], and more recently, in Charlotte-CLT [149]. Additionally, several studies have estimated the theoretical benefits of departure metering at different airports [155, 168]. As one would expect, the results indicate varying benefits in terms of taxi-out time reduction, depending on the level of congestion at the airport.

In addition to developing techniques to optimally hold departures at the gate, researchers have also investigated other closely-related techniques to improve the overall efficiency of departure metering. For example, researchers have proposed algorithms for robust gate assignment of flights to reduce gate-conflicts[3] [103]. Additionally, there have been studies to tactically control departure pushbacks considering surface as well as terminal airspace congestion [49]. However, the benefits from considering airspace congestion are likely to be minimal, given the uncertainties associated with airport surface operations. There have also

---

[3] A gate conflict occurs when an arriving flight needs to wait for the gate occupied by another aircraft.

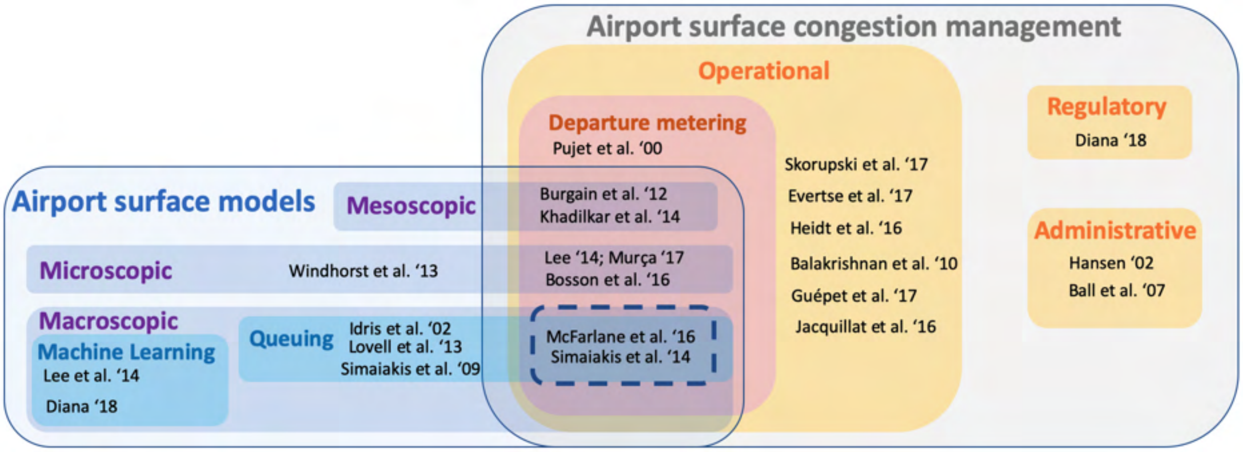


Figure 1-2: Airport surface traffic models and congestion management techniques.

been variants of departure metering in which departures are held near the spot[4] instead of the departure gates [124]. Spot metering has been considered keeping in mind that the FAA only has jurisdictional control over the active movement area in the US, and the ramp areas (including the gate pushbacks) are controlled by the ramp controllers (usually, the airline). However, gate metering is more practical in terms of the overall benefits, and there is push to accommodate gate metering by coordinating with the ramp controllers.

Figure 1-2 illustrates a broad classification of the different airport traffic models and congestion management approaches that we have discussed so far. Queuing models have been widely-used to represent the traffic movements at airports and develop departure metering algorithms. However, as we had mentioned earlier, the key limitations are that prior airport queuing models do not account for multi-region congestion that might be important at certain complex airports, and that prior departure metering algorithms have not addressed demand-side uncertainty. These limitations are addressed in this thesis. Next, we discuss queuing network abstractions in a more general context.

1.2.3 Queuing network models

Queuing networks have been used to model congestion in a wide range of infrastructures, including communication systems, production systems, industrial supply chains, and trans-

---

[4] Spot is the exit of the ramp area into the active movement area

portation systems [178, 181, 147, 131, 130]. A queuing network is a collection of interconnected *servers* that represent the system's capacitated resources, and *customers* who wish to utilize these resources. For example, in an urban traffic network, road intersections can be viewed as the servers, and vehicles as the customers. The demand for a resource can be close to – or even exceed – its capacity, leading to congestion and the formation of large queues; this impact could cascade further into other resources. Congestion results in higher operating costs and increased wait times.

Different mathematical models for the queuing process have been proposed for well over a century, following the seminal paper by Erlang in 1909 [58]. However, most of these are discrete probabilistic models that make restrictive assumptions (such as exponential service time distributions, stationarity, etc.), and may not be valid in practice [167]. Moreover, such discrete probabilistic models result in a large state space for queuing networks, that can impede the analysis and development of control strategies using standard techniques such as dynamic programming. This sentiment is echoed by some leading researchers, for example, John Kingmann, one of the pioneers of queuing theory, has expressed very strong views about re-thinking the probabilistic modeling framework given the problems associated with analyzing queuing network models with probabilistic primitives [105]. To overcome the limitations of discrete probabilistic models, various queuing models based on a continuous approximation of the queue length have been proposed [175]. However, these models may not be amenable to standard control approaches (e.g., diffusion approximation models [104]), or may not generalize beyond a particular application [96]. Recently, an alternative approach has been proposed to model queuing systems using robust optimization techniques, that replaces the probability distributions associated with arrival and service times with uncertainty sets as primitives [28]. Some of the challenges with the robust optimization approach include computational tractability for large queuing networks and incorporating non-stationary distributions. Another recent technique involves using machine learning models such as recurrent neural networks to represent the queuing behavior [74].

In addition to modeling the queuing process, a key challenge in these systems is the development of control strategies that can reduce congestion, while still satisfying operational constraints. The control inputs can vary depending on the specific application: examples include the rate at which customers are sent into the system, or the capacity of servers. In the case of departure metering, one is interested in controlling the input rate (departure pushback rate) into the queuing network (representing airport traffic).

1.2.4 Control algorithms for queuing networks

A variety of control frameworks, based on models of varying complexity, have been proposed to reduce congestion in queuing networks. A significant amount of early research focused on the optimal control of arrival and service rates using Markov Decision Processes (MDPs) [166, 167]. However, the Markov chain representation for queuing dynamics that was used in this MDP framework relied on restrictive assumptions (such as Poisson arrivals, exponential service time distributions, stationarity, and no propagation delays), which may not be valid in practice. In order to better understand the problem of congestion control in queuing networks, we describe the two main contexts in which it has typically been studied: Internet congestion control and urban traffic networks.

Internet congestion control

With the growth of the Internet, there was much interest in the analysis of congestion control protocols for communication networks [145]. The objective of these protocols is to maintain a desired quality of service, as measured in terms of delay, throughput, packet loss, or jitter. The primary congestion control mechanism in these networks was the regulation of the sending rate at the source (e.g., the transmission control protocol), and the queue length at the routers by dropping packets (e.g., active queue management techniques) [72, 70]. Control-theoretic techniques have been used to analyze the stability of such protocols and tune parameters [118]. Internet congestion control also presents some domain-specific challenges; for example, feedback to a source can only be based on delayed packet-loss information rather than actual queue length information.

Control-theory has been used to a limited extent in the design of congestion control protocols. Prior work has considered fluid-flow models, typically non-linear delay differential equations with time-delays to account for the round-trip travel time from the source to the receiver [131]. The resulting models have allowed researchers to use standard control techniques such as PID, $\mathcal{H}_\infty$, and variable structure controllers [86, 96]. To apply these techniques, the models were often linearized around an equilibrium point, and in a few cases, time-delays were even ignored [86, 118, 41]. Most models assumed that the queue length of the bottleneck server is always positive, and that the queue lengths elsewhere were zero. Consequently, the resulting approaches tend to perform poorly in practice, due to a lack of robustness to time-varying traffic, delays, and capacity [96]. Recent developments have included the application of reinforcement learning algorithms for Markov Decision Processes to queuing networks [55]. However, one of the drawbacks of the reinforcement learning approach is that it does not provide any theoretical guarantees; doing so remains an active area of research.

Urban traffic networks

Traffic congestion on urban roads is often represented as a queuing network [142]. The problem of congestion control through regulating traffic signals corresponds to controlling the service rates for each of the flows in the queuing network. Timed traffic lights have been shown to be inefficient under time-varying arrival rates [142]. Furthermore, the control policy needs to account for the downstream impacts of throttling upstream flows. To overcome these challenges, adaptive traffic signaling approaches, such as optimization-based techniques, have been proposed to reduce congestion [76, 45, 151]. However, online optimization techniques for large queuing networks are computationally intensive, while decentralized approaches and heuristics are often sub-optimal. Another set of techniques used to control queuing networks are variants of backpressure algorithms [172]. Although backpressure control originated in communication networks for packet routing, there have been modifications to the algorithm to cater to urban traffic flows [111, 75].

An overview of the broad classification of non-stationary queuing models and control techniques is shown in Fig. 1-3, along with representative research papers associated with each category.

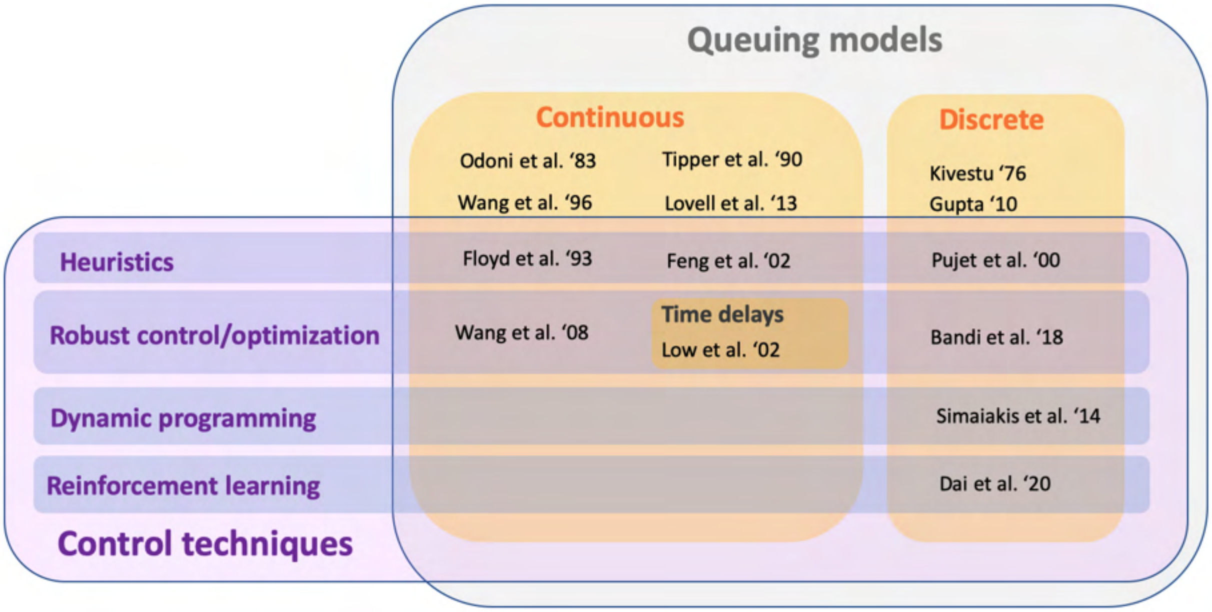


Figure 1-3: Queuing models and control algorithms

1.3 Contributions of this thesis

This thesis focuses on the design and evaluation of control algorithms for efficiently managing airport surface traffic through departure metering. Although this thesis focuses on airport operations, the models and control algorithms introduced here can be applied to other physical systems (e.g. communication networks, urban traffic networks). The primary research contributions of this thesis are as follows:

1. We propose new mathematical models of airport surface traffic movements that are capable of capturing congestion at multiple locations, and that can also account for the time-varying nature of demand and capacity.

2. We design and evaluate airport congestion control algorithms that are robust to uncertainties.

3. We develop techniques to control queuing networks with propagation delays between the servers, and with general service time distributions.

4. We introduce a new method to analyze the performance of queuing networks using reachability analysis.

5. We present opportunities for leveraging other emerging sources of aviation data for analyzing airport operations.

1.3.1 Mathematical models of airport surface traffic movements

Control algorithms to reduce airport congestion require models of aircraft movements on the airport surface. Queuing models are a natural choice since queues can reflect the congestion that occurs at an airport. Most often, only a single departure runway queue is used to represent airport congestion. However, a single queue might be insufficient at some airports where arrivals and departures interact, and even compete for airport resources such as taxiways (for example, traffic movements at CLT as seen in Fig. 1-2). Therefore, more complex queuing network models are needed for airports that have several points of congestion or multiple departure runways. Further, the airport queuing models need to accommodate time-varying traffic demand and airport capacity. An additional layer of complexity is that the queuing models also need to account for propagation time between servers to represent unimpeded travel time for aircraft to move between any two locations. The exact analysis of such non-stationary queuing networks with general service time distribution and time-delays (to represent propagation time between servers) is analytically challenging, motivating the need to develop efficient approximations.

In Chapter 2, we present the development of a new fluid model for time-varying queuing networks [18]. The model is a continuous approximation to the discrete queuing problem, derived by combining results from steady-state queuing theory with the flow conservation principle, to obtain a point-wise stationary approximation. The fluid flow model results in a computationally tractable nonlinear ordinary differential equation (ODE) representation of the queuing dynamics. The model can account for general service time distributions, multiclass queues and propagation time between servers in the queuing network. The proposed queuing model is one of the important methodological contributions of our work.

The queuing network model is adapted to represent airport traffic by leveraging multiple data sources that include historical flight schedules and flight tracks. The model enables the prediction of queue lengths at various congested regions on the airport surface such as those near the departure runways, runway crossing area and the ramp area close to the terminals. Further, using the estimated queue length information, the model can predict the taxi-out times for departures and taxi-in times for arrivals. To demonstrate the general applicability of our methodology, we validate our model for several major airports (in the US, Europe, and Asia), and the models were found to be reasonably accurate. For example, the mean absolute error in the taxi-out time prediction using our proposed queuing model is around 4 to 5 min, depending on the airport and runway configuration. Note that the accuracy of our proposed model is similar to more complex discrete probabilistic queuing models for the airport surface [157]. However, in contrast to prior probabilistic queuing models, our proposed model has several benefits: (a) the model can predict queue length at multiple congested regions on the airport surface, (b) the model can predict taxi-out time as well as taxi-in time, (c) the ODE representation for the queuing dynamics is amenable to the design of efficient congestion control strategies even for complex airports that need to be represented as a large queuing network. In addition to supporting the development of airport congestion control strategies, the proposed model for traffic movements can potentially provide better situational awareness for air traffic controllers, and aid in environmental impact assessments.

We also develop and validate an in-house microscopic airport simulator for testing different surface traffic management strategies. Further, we present an extension of the airport surface queuing network model to include operations in the terminal airspace for departures. Such an integrated surface-airspace model can support tactical scheduling to efficiently insert departures into overhead stream.

1.3.2 Airport congestion control in the presence of uncertainty

An effective traffic management approach to reduce congestion is departure metering, in which departures are appropriately held at their gates in order to reduce taxi-out delays, while ensuring no adverse impact on airport throughput. An aircraft saves fuel by waiting at the gate with its engines off, rather than idling in a departure queue with its engines on. Departure metering is estimated to potentially yield \$5.5-9.5 billion in monetary benefits at the top 35 airports in the US over a 20-year period [134], and is an integral part of airport surface management programs internationally [69, 60]. Departure metering algorithms require models of airport surface traffic, and knowledge of when a flight would be to be ready for pushback, called the Earliest Off-Block Time (EOBT). The key challenge is to develop departure metering algorithms that can yield higher delay reduction, without impacting the airport throughput under various operational uncertainties. While EOBTs are known to be inaccurate due to several reasons, there has been little prior research on characterizing EOBT uncertainty and its impact on departure metering.

In Chapters 3-4, we propose new departure metering algorithms using optimal control and robust control techniques [22, 21]. The control algorithms are based on our airport queuing network models, that account for congestion in multiple areas. We compare our departure metering algorithms to other approaches, including NASA's Airspace Technology Demonstration-2 (ATD-2) logic, which recently completed field trials at CLT in 2019 [179]. Our robust control approach for departure metering, that explicitly accounts for model uncertainties, performs better than the other approaches in stochastic simulations. The departure metering benefits are evaluated for three major US airports: Newark Liberty International airport (EWR), Dallas Fort Worth International airport (DFW), and Charlotte Douglas International airport (CLT). The departure metering benefits depends on the current level of congestion at the airport as expected. For example, the robust control approach yields an average taxi-out time reduction of about 2.2 min per flight at CLT, without resulting in any significant loss in airport throughput.

Further, we analyze the current levels of uncertainty in the EOBT information published by the airlines. The EOBT uncertainty represents uncertainty associated with the time at which an aircraft will be ready to pushback, that has received little attention. Airlines update the EOBT information at irregular intervals till the flight pushes back from the gate. As one might expect, the magnitude of the EOBT uncertainty is larger for longer look-ahead horizons. However, even for look-ahead horizons of 10-20 min (when tactical departure metering decisions are made), the standard deviation of the EOBT errors is about 4 to 6 min, which is of the same magnitude as the errors in the taxi-out time prediction from the airport queuing models. This highlights the need to consider EOBT uncertainty to evaluate the benefits of departure metering algorithms. Moreover, we found that the current levels of EOBT uncertainty lead to a 50% reduction in benefits at some airports when compared to an ideal case with no EOBT uncertainty [20, 21]. Further, our analysis was able to quantify the marginal improvement in departure metering benefits due to reduction in EOBT uncertainty. The airlines can use such estimates to decide on investment decisions to develop better models to reduce the EOBT uncertainty.

1.3.3 Robust control of queuing networks

In addition to providing an efficient solution to departure metering, we also make some fundamental contribution to the control of queuing networks. Queuing networks have been widely-used to model congestion in various infrastructure systems. In these systems, one is interested in developing control strategies to reduce congestion or queue length. In many practical systems, the service-time distribution can take any form (not just exponential) and can be non-stationary (time-varying), and there can be propagation delays between the nodes in the network, making the traditional MDP approaches unfeasible. Other control-theoretic approaches often make restrictive assumptions on the queuing dynamics (such as linearization or ignoring the time delay in the system) to simplify the model for the analysis, which might not be applicable.

To solve this challenging control problem, in Chapter 4 we propose a new robust control approach for queuing networks. In particular, we consider the regulation of arrivals into a queuing network in order to maintain a desired level of occupancy (queue length) in the system. The dynamics of the queuing network is represented by a fluid-flow model, which is then used to develop a robust controller for tracking the desired queue length. We would like to highlight that designing a control policy for the resulting system is a challenging problem in itself because the system is nonlinear, with model uncertainties, time-delays and input saturation (to represent non-negativity of sending rates). To address these challenges, we develop a controller based on a sliding mode control approach, with predictor-based feedback to account for propagation delays. For a single queue, we determine sufficient conditions for tracking the queue length, and bounds on the tracking error using Lyapunov analysis. We also investigate the tracking performance for queues in tandem. The proposed framework can be extended to any queuing network topology. Finally, we demonstrate the practical applicability of our approach to tactically manage aircraft departures in order to reduce congestion on the airport surface.

1.3.4 Reachability analysis for analyzing the performance of queuing networks

In addition to developing control algorithms for queuing networks, a closely related problem that is of significant interest is to analyse the performance of queuing networks. One is often interested in computing performance metrics such as queue length and wait time. Traditionally, stochastic simulations are used to compute the performance metrics. The simulations involve tracking individual customers in the system by sampling the arrival and service time distributions, and such simulations need to be performed multiple times to compute statistical quantities of interest. However, simulations might not be computationally tractable for large queuing networks as it involves sampling from a large joint distribution, particularly, if one wants to investigate the impact of uncertainty in network parameters such as the service time of the servers.

In Chapter 5, we propose an alternative tool to analyze the performance of large-scale uncertain queuing networks. In particular, we present an approach to compute the range of expected queue length in a network given the range of uncertainty in the queue parameters using reachability analysis. Reachability analysis is a technique that has been used for controller verification and model checking. It involves computing the sets of future states achievable by a system given an uncertain set of initial states and input parameters. We utilize the queuing network model developed in Chapter 2, and some recent results in reachability analysis to compute the range of expected queue length given uncertainty in the server service rate, arrival rates and initial queue length.

The approach bears some philosophical similarity to the recent robust optimization approach to analyze the performance of queuing networks [28] in a sense that both the methods take a non-probabilistic approach by specifying the underlying randomness as uncertainty sets. However, in contrast to the the robust optimization approach, our approach relies on control-theoretic techniques for continuous dynamical systems, and the underlying queuing model is different. This gives the reachability analysis approach a significant computational advantage. To the best of our knowledge, this is the first attempt in employing reachability analysis for queuing networks.

1.3.5 Characterizing airport operations using emerging aviation data sources

To develop practical decision support tools for air traffic management (such as departure metering), one needs to have accurate models of airport operations that are developed using actual operational data. In recent years, a variety of data sources of different granularity are available to characterize airport operations. Our proposed queuing models to represent airport traffic movements were developed using flight trajectories, flight schedules and weather information. However, there are other emerging data sources of airport operations that are largely unexplored which can aid in the development of air traffic models and decision support tools. In Chapter 6, we present models to analyze airport operations using other sources of aviation data using two illustrative examples.

First, we present a statistical model to accurately estimate aircraft fuel consumption during taxi operations using flight trajectories. The fuel burn model is intended to aid environmental monitoring of current operations and for the development of new procedures. The recent availability of Flight Data Recorder (FDR) data provides direct measurements of engine fuel flow rates and flight trajectories. Our analysis shows that periods during which a taxiing aircraft accelerates are correlated with an increase in its fuel flow rate. We therefore develop a multi-model smoother to extract acceleration events from the trajectory data, and use these events to develop statistical models to estimate aircraft fuel burn. The proposed trajectory-based fuel burn models have a lower mean absolute percentage error of up to 37% relative to the traditional fuel burn models used in the industry tools that consider only taxi times as inputs. In addition to FDR data (which has restricted access due to privacy concerns), we show that our proposed models work well even with noisy trajectories from airport surveillance systems (such as ASDE-X) that are readily available to regulatory agencies such as the FAA. Our fuel burn models could potentially be used to obtain more accurate estimates of noise and emissions on the airport surface, quantities of significant interest to the regulatory agencies such as the FAA to monitor current operations and perform what-if analysis to design new procedures.

Next, we present an approach to extract operational information from air traffic controller voice communications that has largely been overlooked. Air traffic controller (ATC) voice communications contain enormous amounts of operational data such as runway assignment, taxi routes, aircraft intent, heading, altitude, etc. Furthermore, ATC voice data contains certain information that are not available from other data sources. A good example in the context of modeling departure operations is the aircraft push-ready time. To make use of such operational information, one needs to have the ability to extract the information from the voice data in an efficient way. A natural approach is to use Automatic Speech Recognition (ASR) systems. However, ASR systems proposed to date for the ATC applications have not yet demonstrated the levels of accuracy needed for practical deployment [137]. Factors such as noisy radio channels, high speech rates, and diverse accents pose challenges to the development of ASR systems for the ATC domain. We present an automatic speech recognition model tailored to the ATC domain that can transcribe ATC voice to text. The transcribed text is used to extract operational information such as call-sign and runway number. The models are based on recent improvements in machine learning techniques for speech recognition and natural language processing. In addition to aiding model development, automatic transcription of ATC communications has the potential to improve system safety, operational performance, conformance monitoring, and to enhance air traffic controller training.

1.4 Organization of this thesis

The reminder of this thesis is organized as follows:

- In Chapter 2, we present a new queuing network model for the aircraft movements on the airport surface. The model is developed and validated for major airports. Additionally, we discuss extending the surface queuing model into the terminal airspace, to predict the transit time from the departure gate to the final departure fix. We also present a comparison of the queuing model performance with an in-house microscopic simulator.

- Chapter 3 focuses on the departure metering algorithms. We first present the potential benefits of departure metering with the ATD-2 logic, along with a methodology to pick the optimal buffer required for the logic. We also investigate the impact of uncertainty in the airline-supplied demand information on the benefits of departure metering. Finally, we present a new optimal control approach to departure metering.

- In Chapter 4, we present a robust control framework for controlling arrivals into a queuing network with propagation delays between servers. We determine sufficient conditions for tracking the queue length at a desired value, and bounds on the tracking error for certain special cases. Furthermore, we apply the robust control framework for departure metering and evaluate its benefits using simulations.

- In Chapter 5, we introduce reachability analysis as a means to analyse the performance of queuing networks.

- In Chapter 6, we discuss opportunities for leveraging other emerging sources of aviation data in airport operations analysis, through two illustrative examples: (a) estimating aircraft taxi fuel consumption from flight trajectories, and (b) extracting operational information such as runway number and flight ID from air traffic controller voice communications.

- In Chapter 7, we present a summary of the thesis, opportunities for practical implementation, and discuss some promising directions for future research.

Chapter 2

Queuing network model of the airport surface

2.1 Introduction

In this chapter, we present a new queuing network model to represent the traffic movements on the airport surface. Airport surface queuing models have traditionally assumed that congestion occurs exclusively near the departure runway [160, 157, 159, 17]. While this is true at some airports, at others with complex layouts such as Charlotte Douglas International airport (CLT), congestion can occur at multiple locations. Moreover, arrivals and departures interact and even compete for airport resources such as taxiways. In this chapter, we illustrate that at such airports, multistage congestion can be best represented by a queuing network. We first present a queuing network representation for the traffic movements on the airport surface, followed by a fluid-flow model for the queuing network that can account for the time-varying nature of the queues. The models are validated using actual operational data from several major airports. Finally, we also show a comparison of the predictions obtained from the queuing model with those obtained from an in-house microscopic simulator of the airport surface.

2.2 Queuing models of airport surface operations

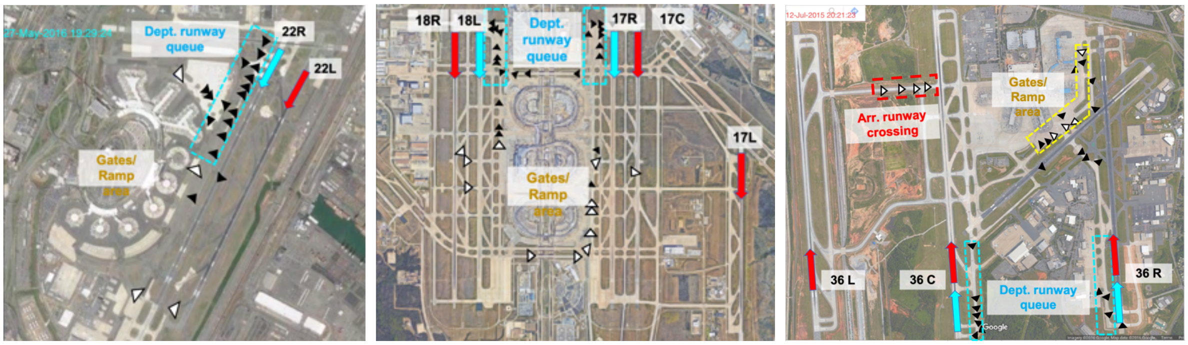
Queuing models can be used to determine the lengths of various queues on the airport surface in order to predict the taxi times of departures and arrivals. The taxi-out time of a flight is determined as the sum of the unimpeded travel time (the time it would take for the aircraft to move from the gate to the runway with no congestion) and the wait time in the queues. In general, queues are formed when the demand for resources exceeds the available capacity. In the case of the airport surface, runways and certain taxiways become bottlenecks during periods of heavy traffic, leading to the formation of queues. Since the level of congestion depends on demand, aircraft fleet mix, airport procedures, taxi routes, runway configuration (set of runways used for arrivals and/or departures) and weather, queuing models need to be adapted to different operating conditions.

In this section, we present the queuing network model for the following three airports to illustrate the general applicability: Dallas Fort-Worth International Airport (DFW), Charlotte Douglas International Airport (CLT) and Newark Liberty International Airport (EWR). These airports are chosen in part because of the differences in layout and traffic. DFW (621,684 aircraft movements in 2017, the 4th busiest airport in the world by number of movements) and CLT (546,845 aircraft movements, the 7th busiest airport in the world by number of movements) are major hubs for American Airlines [68]. There have been demonstrations of departure metering at CLT under Phase 2 of the ATD-2 program since November 2017, and DFW has been selected as the site for Phase 3 of the program. EWR handled 432,941 operations in 2017, is a hub for United airlines in the highly-constrained New York metroplex, and is prone to frequent congestion.

2.2.1 Queuing network models of EWR, DFW, and CLT

Figure 2-1 shows the airport layouts of EWR, DFW, and CLT, along with a snapshot of the traffic movements on the surface to illustrate surface queues. Departures and arrivals are represented as black and white triangles, respectively. Each of these airports operates primarily in one of two runway configurations: North Flow (NF) and South Flow (SF).

Runway configuration refers to the set of active runways that are used for an extended time period. Figure 2-1 illustrates the most prevalent runway configuration: SF for EWR (58% of the time in 2017), SF for DFW (70% of the time), and NF for CLT (53% of the time). The layout for DFW is cropped to show only the primary runways used in South Flow.



(a) EWR-SF (b) DFW-SF (c) CLT-NF

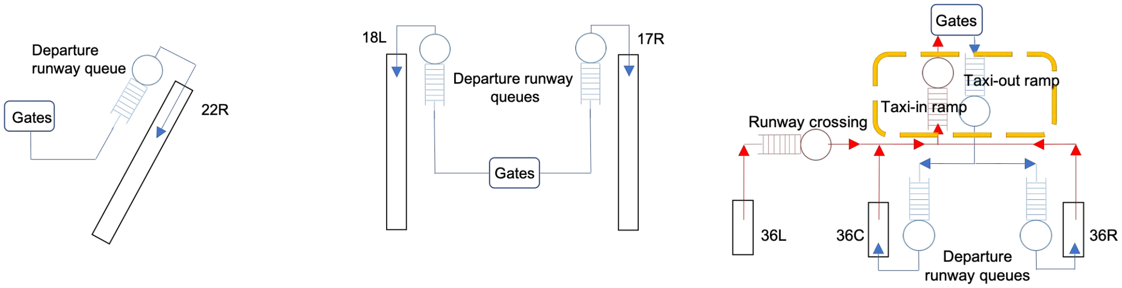


Figure 2-1: Snapshots of traffic movements at EWR, DFW and CLT, and the corresponding queuing representations under the most frequently-used configurations. The primary departure and arrival runways are indicated in the figure using blue and red arrows, respectively. Departures and arrivals are represented as black and white triangles, respectively.

All three airports experience large queues of aircraft waiting for take-off near the departure runways. The number of departure runway queues varies by airport and runway configuration: the scenarios considered in this thesis result in one departure runway queue at EWR (22R), and two each at DFW (18L, 17R) and CLT (36C, 36R). Runway separation requirements are the primary driver of their capacity, and depend on weather (instrument vs. visual meteorological conditions, IMC or VMC), the fleet mix of aircraft using the runway, and the relative proportion of arrivals and departures [156].

Some airports (for example, CLT) experience queuing in the ramp area (region close to the airport terminal building) in addition to near the departure runways. Consequently, flights at CLT spend nearly half of their taxi-out time in the ramp area (Table 2.1). Queuing network models can be used to represent multiple points of congestion on the airport surface. Figure 2-1 shows examples of queuing network models for EWR, DFW, and CLT. Since EWR and DFW do not exhibit significant ramp congestion, queuing is assumed to occur primarily at the departure runway(s). By contrast, the queuing network model for CLT includes a ramp queue in addition to two departure runway queues. Arrivals and departures interact in the ramp area, motivating the need to model the taxi-in process. The taxi-in process in CLT-NF is represented as follows: flights landing on the leftmost runway (36L) pass through a runway crossing queue and a taxi-in ramp queue, whereas flights landing on one of the other runways just pass through the taxi-in ramp queue. The service rate of the taxi-out ramp server is modeled as a function of the taxi-in ramp queue length, and vice versa. The three queuing network models (EWR-SF, DFW-SF, and CLT-NF) correspond to a progression of complexity, going from a single runway queue to two runway queues, and finally multiple runway and ramp queues.

Airport	**Config.**	**Avg. gate-to-spot (min)**			**Avg. spot-to-runway (min)**			**Avg. taxi-out (min)**
		Unimpeded	Delay	Total	Unimpeded	Delay	Total	
EWR	NF	5.7	3.1	8.8	6.0	6.4	12.4	21.2
	SF	5.9	3.2	9.1	5.4	6.1	11.4	20.6
DFW	NF	5.1	2.4	7.4	6.5	4.4	10.9	18.4
	SF	5.1	2.4	7.5	5.4	3.9	9.4	16.8
CLT	NF	6.3	3.5	9.8	5.1	5.3	10.4	20.2
	SF	7.4	4.2	11.6	3.2	4.4	7.6	19.2

Table 2.1: Taxi-out times, including the unimpeded and delay components, at EWR, DFW, and CLT, separated by ramp area (gate-to-spot) and active movement area (spot-to-runway). The statistics are computed using multiple data sources [138, 67, 68] for a three month period (May-July 2016) for DFW and EWR, and a five month period for CLT (May-July 2015; May-Jun 2016).

The taxi-out processes under each of the secondary runway configurations (EWR-NF, DFW-NF, CLT-SF) is represented using a single queue for each active departure runway. Although the average gate-to-spot delays are not negligible for EWR (NF and SF) and CLT (SF) (as seen Table 2.1), a separate ramp queue for these runway configurations is not considered because the gate-to-spot delays are caused by the build up of the departure runway queue and its subsequent overflow into the ramp area. This is unlike in the case of CLT (NF), where the gate-to-spot delays are caused because of the bottlenecks in the ramp area.

2.2.2 Models of queue dynamics

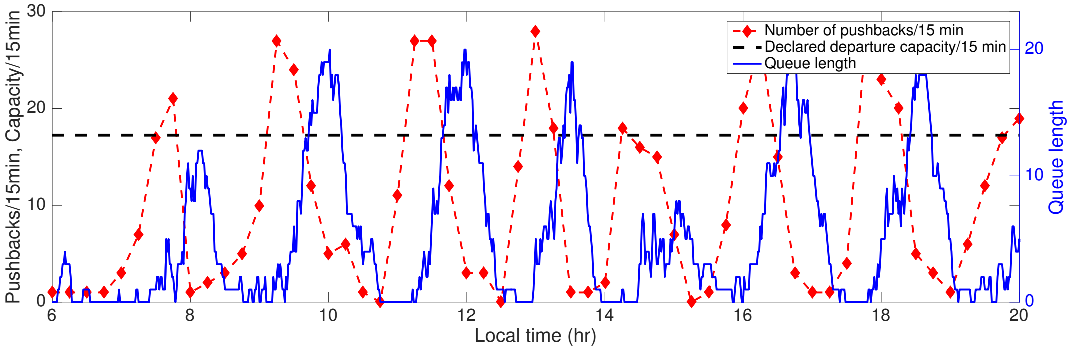


Figure 2-2: Number of pushbacks (per 15 min), declared departure capacity (per 15 min) and sum of the queue lengths near the two departure runways for a typical day at CLT (05/22/2016).

A distinguishing feature of queues on the airport surface is their non-stationarity, which is driven by the time-varying nature of demand and capacity. Figure 2-2 shows the number of pushbacks, declared departure capacity [68], and the sum of the queue lengths near the two departure runways at CLT on a representative good weather day (5/22/2016). The variations in departure demand (pushbacks) are as expected at a hub airport with banked operations. Demand exceeds departure capacity during each departure bank, leading to the formation of large queues that fluctuate over the course of the day. Capacity also varies because of changes in weather and arrival rate, resulting in time-varying queues.

Fluid model of non-stationary queues.

The exact analysis of non-stationary queuing networks is analytically challenging, motivating the use of numerical simulations or approximations [106, 82, 117]. Probabilistic models using Markov chains can be restrictive in terms of the service time distributions they can accommodate, and lead to high-dimensional state spaces for large queuing networks. The size of the state space also poses a challenge to the development of control strategies. A key innovation of this work is the development of a fluid model of the time-varying taxi-out queuing network. The model is a continuous approximation to the discrete queuing problem, derived by combining results from steady-state queuing theory with the flow conservation principle, to obtain a point-wise stationary approximation [175, 183, 18].

Single queue and server.

We first consider the simple case of a single queue and a single server, with time-varying arrival and service rates. Let $x(t)$ represent the average number of customers in the queue at time t. Let $\lambda(t)$ and $\mu(t)$ denote the average arrival rate and service rate at time t, respectively. The word "average" here denotes the ensemble average at a particular time-instant. Let $f_{\text{in}}(t)$ and $f_{\text{out}}(t)$ represent the in-flow and out-flow rate from the queue at time t. From the flow conservation principle, we have:

$$\dot{x}(t) = -f_{\text{out}}(t) + f_{\text{in}}(t). \tag{2.1}$$

Assuming that there are no constraints on the queue length, $f_{\text{in}}(t) = \lambda(t)$. For the out-flow, $f_{\text{out}}(t) = \mu(t)\rho(t)$, where $\rho(t)$ is the average utilization of the server. The queue dynamics is given by:

$$\dot{x}(t) = -\mu(t)\rho(t) + \lambda(t). \tag{2.2}$$

The average utilization, $\rho(t)$, is approximated by a function, $G(x(t))$, which satisfies the following properties: (a) $G(0) = 0$ and $G(\infty) = 1$; and (b) $G(x)$ is strictly concave and nonnegative $\forall x \in [0, \infty)$, in order to represent congestion. The dynamics for $x(t)$ can then be rewritten in terms of $G(x)$ as:

$$\dot{x}(t) = -\mu(t)G(x(t)) + \lambda(t), \quad x(0) = x_0. \tag{2.3}$$

The expression for $G(x)$ is obtained by matching the steady-state number of customers in the system. Assuming a Poisson arrival process, the Pollaczek-Khinchine formula (2.4) provides an expression for the mean number of customers (x_s) at steady state [183], namely,

$$x_s = \rho + \frac{\rho^2(1 + C_v^2)}{2(1 - \rho)}. \tag{2.4}$$

Here, C_v is the coefficient of variation of the service time distribution. Expressing ρ in terms of x_s, we get:

$$\rho = \frac{x_s + 1 - \sqrt{x_s^2 + 2C_v^2 x_s + 1}}{1 - C_v^2}. \tag{2.5}$$

Rewriting x_s in terms of ρ and using the fact that $G(x)$ is an approximation of ρ:

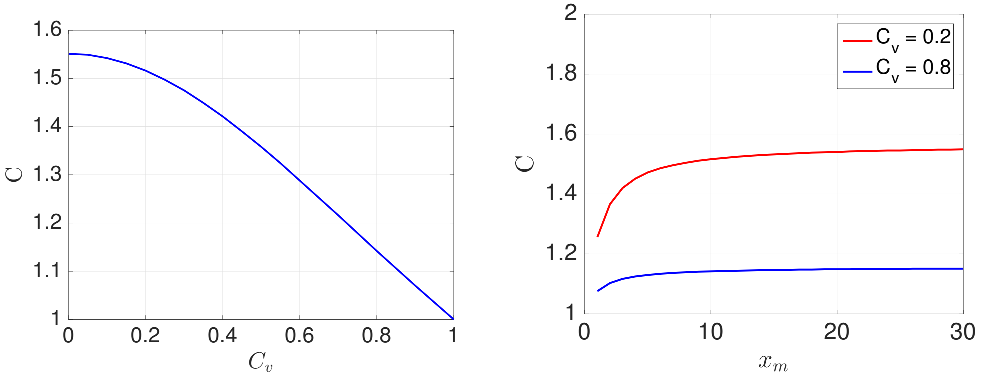
$$\rho(t) \approx G(x) = \frac{x + 1 - \sqrt{x^2 + 2C_v^2 x + 1}}{1 - C_v^2}. \tag{2.6}$$

For a server with exponential service time distribution, $C_v = 1$, and the above expression for $G(x)$ simplifies to $x/(1 + x)$. Motivated by this expression, the function $G(x)$ is approximated by $Cx/(1 + Cx)$ for a generic service time distribution, where the parameter C is given by:

$$\arg\min_{C} \int_0^{x_m} \left(G(x) - \frac{Cx}{(1 + Cx)}\right)^2 dx, \tag{2.7}$$

where x_m denotes the maximum queue size expected in the system. The parameter C is essentially an empirical value that depends on C_v and x_m. The dependence of the C parameter on C_v for $x_m = 10$ is shown in Figure 2-3(a). The maximum queue size expected in a system can be obtained empirically from operational data. Additionally, the value of C does not change significantly with x_m if x_m is considered to be sufficiently large (see Figure 2-3(b)). This is primarily because $G(x)$ saturates close to one for large values of x. Moreover, C is less sensitive to x_m for larger values of C_v and it is completely independent of x_m for $C_v = 1$ (case with exponential service time distribution).

Figure 2-4 compares the approximation for $G(x)$ and the actual value for the case of an Erlang distribution with shape parameter of 10 and rate of 5, resulting in a coefficient of variation (C_v) of 0.2. In this example, if $x_m = 10$, then $C = 1.5$. There is a good match between the approximation and the actual value. Finally, the above approximation for $G(x)$



(a) Dependence on C_v for $x_m = 10$ (b) Dependence on x_m for $C_v = 0.2, 0.8$

Figure 2-3: The dependence of the C parameter on C_v and x_m.

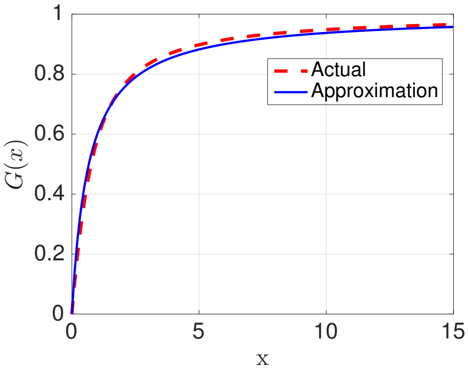
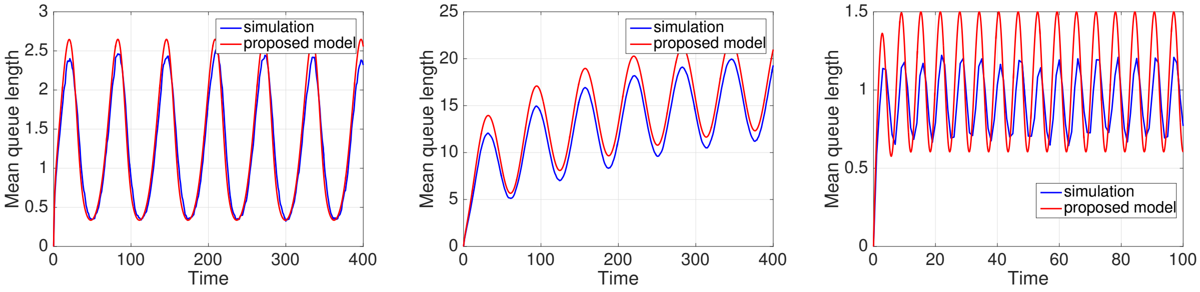


Figure 2-4: Comparison between the approximation for $G(x)$ and the actual value ($C_v = 0.2, x_m = 10, C = 1.5$).

results in the following ODE for the evolution of the mean queue length:

$$\dot{x}(t) = -\mu(t)\frac{C(t)x(t)}{1 + C(t)x(t)} + \lambda(t). \tag{2.8}$$

We note that the above expression for the evolution of the ensemble mean queue length is one of the important methodological contributions of this thesis. We illustrate the performance of the analytical queuing model for some standard queues. Figure 2-5 compares the mean queue length for a $M_t/M/1$ queuing system obtained from the analytical queuing model, against the corresponding output of a discrete stochastic simulation (obtained from 3,000 independent samples). The results from the analytical queuing model are found to closely match the simulation. As the load on the server ($\frac{\lambda(t)}{\mu}$) increases, the model begins to deviate from the discrete simulation (Figure 2-5(b)). The model also shows deviations if the queue length fluctuates rapidly, as seen in Figure 2-5(c), which was obtained by increasing the sinusoidal frequency of the arrival rate into the queue. Such a deviation is expected since the assumption of a point-wise stationary approximation breaks down. Finally, although prior work has considered finite queue buffers for point-wise stationary fluid flow models [183], the results are too complex for practical implementation. However, the limitation of unconstrained queue sizes does not preclude these analytical queuing models from being used at most major airports (that will be shown by validating the model predictions with actual operational data later in Section 2.2.5).



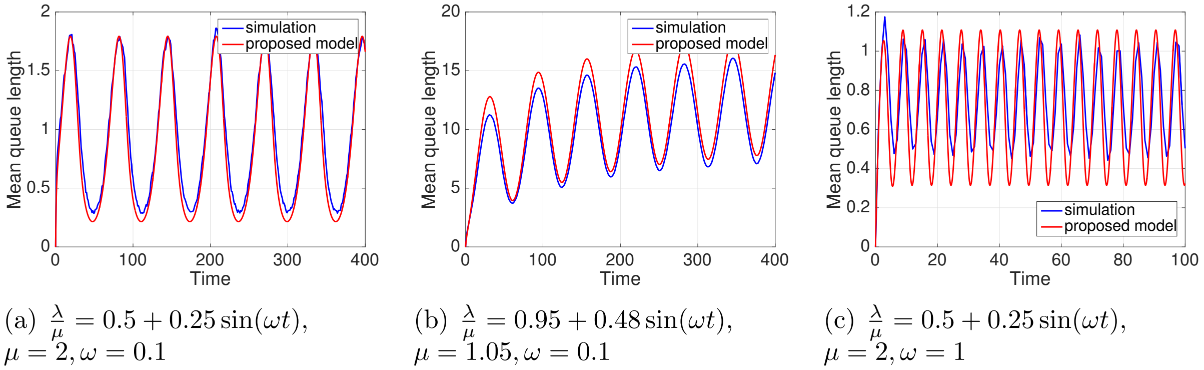
(a) $\frac{\lambda}{\mu} = 0.5 + 0.25\sin(\omega t)$, $\mu = 2, \omega = 0.1$

(b) $\frac{\lambda}{\mu} = 0.95 + 0.48\sin(\omega t)$, $\mu = 1.05, \omega = 0.1$

(c) $\frac{\lambda}{\mu} = 0.5 + 0.25\sin(\omega t)$, $\mu = 2, \omega = 1$

Figure 2-5: Comparison of the mean queue lengths obtained from simulations and the proposed analytical queuing model for an $M_t/M/1$ queue. Note the different x-axis limits.

We consider the case of an Erlang service time distribution (Figure 2-6), which is encountered frequently in communication networks and transportation systems. Once again, the analytical model is a close match to the exact simulations, while offering a considerable computational advantage. For example, the computational time for the queuing simulations (with 3000 independent samples) for this particular case was 65 s, whereas, the analytical model took 2 ms, several orders of magnitude smaller.



(a) $\frac{\lambda}{\mu} = 0.5 + 0.25\sin(\omega t)$, $\mu = 2, \omega = 0.1$

(b) $\frac{\lambda}{\mu} = 0.95 + 0.48\sin(\omega t)$, $\mu = 1.05, \omega = 0.1$

(c) $\frac{\lambda}{\mu} = 0.5 + 0.25\sin(\omega t)$, $\mu = 2, \omega = 1$

Figure 2-6: Comparison of the mean queue lengths obtained from simulations and the proposed analytical queuing model for an $M_t/E_{10}/1$ queue. Note the different x-axis limits.

Multiclass queues.

The model for a single queue can be extended to handle multiple classes of customers under the assumption that all classes have equal priority and same service time distribution. For example, in the context of airport surface operations, one can consider aircraft with different runway assignments as multiple classes of customers (which will be presented in detail later in Section 2.2.3). Let $i = 1, 2, \cdots, l$ denote different classes of customers in the system, and x_i be the number of customers of class i in the queue buffer. The evolution of total number of customers in the queue ($x_T = \sum_{i=1}^{l} x_i$) can be obtained using Eq. (2.8):

$$\dot{x}_T = -\mu \frac{Cx_T}{1 + Cx_T} + \lambda. \tag{2.9}$$

The effective mean service rate for each class in the queuing dynamics is assumed to be proportional to the fraction of customers of that particular class in the queue buffer, considering service time distributions are same for all the customers . Using this assumption, the evolution of mean queue length of a particular class i is given by:

$$\dot{x}_i \;=\; -\mu \frac{Cx_T}{1 + Cx_T} \frac{x_i}{x_T} + \lambda_i \;=\; -\mu \frac{Cx_i}{1 + Cx_T} + \lambda_i. \tag{2.10}$$

Queuing networks.

The single queue model can be extended to a network of multiple queues using the flow conservation principle: The output of one queue becomes the input to the second queue if they are connected. Let R be the routing matrix, with elements r_{ij} representing the fraction of customers joining queue j after being served by server i. Let λ_i be the exogenous input into queue i with mean service rate μ_i. The dynamics of the mean queue length for each queue in the network is given by:

$$\dot{x}_i = -\mu_i \frac{C_i x_i}{1 + C_i x_i} + \lambda_i(t) + \sum_j \mu_j \frac{C_j x_j}{1 + C_j x_j} r_{ji}. \tag{2.11}$$

While deriving the analytical model for a single queue, we assumed that the arrival process into the queue is Poisson. However, the output of one queue that becomes the input to the second queue need not follow a Poisson process for a general service time distribution. We make an approximation that the arrivals into the queue follow a Poisson process even in the case of a queuing network to obtain Eq. (2.11). However, one needs to note that this particular assumption might not work well for small queue buffers or when there is low variability in the service time distribution (a detailed discussion on the consequences of Poisson assumption is presented in Section 2.4.3).

Queuing networks with time-delays.

Another aspect that arises in many realistic queuing networks are time-delays due to propagation. Note that the propagation delay does not include the wait time in the queue. Let τ_{ij} be the propagation time (travel time) from server i to j. Then, the mean queue length is given by the following delay differential equation:

$$\dot{x}_i = -\mu_i \frac{C_i x_i}{1 + C_i x_i} + \lambda_i + \sum_j \mu_j(t - \tau_{ji}) \frac{C_j(t - \tau_{ji}) x_j(t - \tau_{ji})}{1 + C_j(t - \tau_{ji}) x_j(t - \tau_{ji})} r_{ji}(t - \tau_{ji}). \tag{2.12}$$

Since the arrival times and service times are stochastic in nature, the queue length is a random variable. The proposed analytical queuing model governs the evolution of the ensemble mean queue length. By contrast, probabilistic queuing models such as Markov chains provide the probabilities of queue length at any time instant. However, such probabilistic models are often complex, making it difficult to model large queuing networks and to develop feedback controllers. As an alternative, one can develop congestion control strategies using just the ensemble mean queue length (that is provided by our proposed model).

2.2.3 Application of analytical queuing models to the airport surface

Suppose the airport surface is represented using a queuing network with p queues. Let $\mathbf{x}(t) \in \mathbb{R}^p$ be a vector of queue lengths on the airport surface at any time instant t. Let $\mathbf{u}_d(t) \in \mathbb{R}^q$ be a vector of the pushback rates at time t, where each component, $\mathbf{u}_{d_i}(t)$ represents the pushback rate to the i^{th} runway. The pushback rate refers to the number of departures pushing back from the gate per unit time (say, 5-min). Similarly, let $\mathbf{u}_a(t) \in \mathbb{R}^q$ be a vector of landing rates of arrivals on the runway. Let $\mu(t) \in \mathbb{R}^p$ be a vector of the time-varying mean service rates for each of the servers, and let $\mathbf{C}(t) \in \mathbb{R}^p$ be the vector of parameters associated with the approximation of the utilization factor for each of the servers (from Eq. (2.7)). The input to each queue in the network is assumed to be Poisson. The evolution of the queues can then be expressed as:

$$\begin{aligned}\dot{\mathbf{x}} = \mathbf{f}\Big(&\mathbf{x}(t), \mathbf{x}(t-\tau_1), \cdots, \mathbf{x}(t-\tau_m), \mu(t), \mu(t-\tau_1), \cdots, \mu(t-\tau_m), \mathbf{C}(t), \mathbf{C}(t-\tau_1), \cdots, \mathbf{C}(t-\tau_m), \\ &\mathbf{u}_d(t-\tau_{m+1}), \cdots \mathbf{u}_d(t-\tau_w), \mathbf{u}_a(t-\tau_{m+1}), \cdots \mathbf{u}_a(t-\tau_w)\Big), \end{aligned} \tag{2.13}$$

where τ_k is the time to move unimpeded between two points in the queuing network, for $k = 1, 2, \cdots, w$. The function $f(.)$ is obtained from the fluid model and depends on the connectivity of the queuing network. We consider the taxi-out process for the following cases:

- **[Case 1]** An airport represented using q departure runway queues:

$$\dot{x}_{r_i} = -\mu_{r_i}(t)\frac{C_{r_i}(t)x_{r_i}(t)}{C_{r_i}(t)x_{r_i}(t)+1} + u_{d_i}(t-\tau_i),\ i = 1, 2, \cdots, q \tag{2.14}$$

  where x_{r_i} represents the queue length of the i^{th} departure runway, and τ_i is the average unimpeded travel time from a gate to the i^{th} departure runway. The unimpeded time is computed as the 10th percentile of the taxi-time distribution. This model with only runway departure queues is used for EWR ($q = 1$), DFW ($q = 2$), and CLT-SF ($q = 2$).

- **[Case 2]** An airport modeled by a single ramp queue and q departure runway queues:

  We model the ramp queue as a multi-class queue, the class of customers representing the runway assignment of the aircraft in the queue. The service rate for a particular class is proportional to the number of customers of that class in the queue. The queuing dynamics are then given by

$$x_s = \sum_{i=1}^{q} x_{s_i} \tag{2.15}$$

$$\dot{x}_{s_i} = -\mu_s(t)\frac{C_s(t)x_{s_i}(t)}{C_s(t)x_s(t)+1} + u_{d_i}(t-\tau_{gs}) \tag{2.16}$$

$$\dot{x}_{r_i} = -\mu_{r_i}(t)\frac{C_{r_i}(t)x_{r_i}(t)}{C_{r_i}(t)x_{r_i}(t)+1} + \mu_s(t-\tau_{si})\frac{C_s(t-\tau_{si})x_{s_i}(t-\tau_{si})}{C_s(t-\tau_{si})x_s(t-\tau_{si})+1}, \tag{2.17}$$

  where x_{s_i} represents the number of aircraft in the ramp queue that are bound for the i^{th} departure runway, τ_{gs} is the average unimpeded travel time from the gate to the spot, and τ_{si} represents the unimpeded travel time from the spot to the i^{th} runway. All departures are assumed to traverse through a single congested spot to reach the runway. While this assumption is found to be reasonable for CLT-NF, the approach can be easily extended to scenarios with multiple ramp queues.

For CLT, in addition to the taxi-out queues, we model the taxi-in queues to capture the ramp congestion. The taxi-in queuing dynamics consists of a runway crossing queue and ramp queue. The taxi-in queuing model for CLT is as follows: Flights landing on 36L have to pass through a runway crossing queue before entering the ramp area. Let $u_{a,ri}$ be the landing rate (of arrivals) on the i^{th} runway. The input to the runway crossing queue is the landing rate on runway 36L, with an associated delay (τ_{a1}) accounting for the travel time to reach the runway crossing queue from the point of touchdown. The dynamics of the runway crossing queue length ($x_{a,r1}$) are given by

$$\dot{x}_{a,r1}(t) = -\mu_{a,r1}(t)\frac{C_{a,r1}x_{a,r1}(t)}{1 + C_{a,r1}x_{a,r1}(t)} + u_{a,1}(t - \tau_{a1}). \tag{2.18}$$

An aircraft is defined to be in the taxi-in ramp queue if it has entered the ramp area but is yet to reach the gate, and has exceeded the unimpeded spot to gate time. The inflow to the taxi-in ramp queue is the sum of landing rates on two runways (36C and 36R) and the output from the runway crossing queue, delayed by the average unimpeded spot to gate time (τ_{as}). We could also account for the travel time from the runway (36C and 36R) to the spot in the delay term. However, we ignore it here since it is a small value, but consider it later while computing the taxi time. The equation for the taxi-in or arrival ramp queue length ($x_{a,s}$) is given by

$$\dot{x}_{a,s}(t) = -\mu_{a,s}(t)\frac{C_{a,s}x_{a,s}(t)}{1 + C_{a,s}x_{a,s}(t)} + \mu_{a,r1}(t-\tau_{as})\frac{C_{a,r1}(t - \tau_{as})x_{a,r1}(t - \tau_{as})}{1 + C_{a,r1}(t - \tau_{as})x_{a,r1}(t - \tau_{as})} + u_{a,2}(t-\tau_{as}) + u_{a,3}(t-\tau_{as}) \tag{2.19}$$

2.2.4 Service rates of the servers

The model parameters such as the service rates of the servers were determined from airport operational data that include flight tracks [67], the actual pushback, in-air (wheels-off), landing (wheels-on) and in-gate times, and gate assignments [138], and meteorological conditions at the airport [68]. The data was obtained for a three month period (May-July 2016) for DFW and EWR, and a five month period for CLT (May-July 2015; May-Jun 2016). The dataset includes the periods of peak demand experienced in summer. Approximately 70% of the data was used to train the model parameters, and the rest was used for testing. The service time distribution of each server is obtained by computing the difference between successive out-times from the queue when there is pressure on the server[1]. A brief description

---

[1] In our analysis, we consider the server is under pressure if the queue length is greater than two

about determining the service rates for the different types of servers is discussed below:

Ramp queues

Congestion develops in the ramp area primarily because some taxi paths are shared by both arriving and departing flights, while they head in different directions. The service rate of the taxi-in ramp server is modeled as a function of the queue length of the taxi-out ramp queue, and vice versa. Fig. 2-7(a) shows that the mean service rate of the taxi-in ramp server decreases linearly as the number of aircraft taxiing out increases, illustrated using data from CLT-NF. A linear fit to the data yields the following relationship for the service rates as a function of the traffic on the ramp

$$\mu_{d,s}(t) = -0.033 x_{a,s}(t) + 1.7 \tag{2.20}$$

$$\mu_{a,s}(t) = -0.066 x_{d,s}(t) + 1.6, \tag{2.21}$$

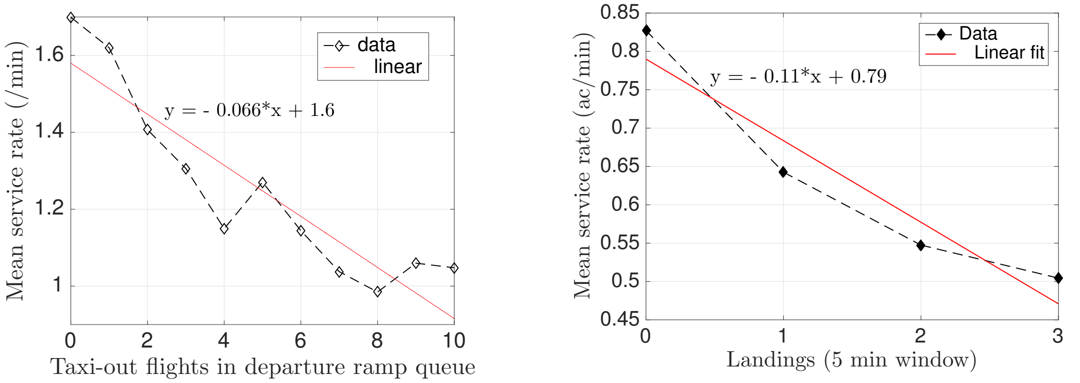
where the service rates are expressed in aircraft/min. Note that the slopes are quite different for taxi-in and taxi-out ramp servers, even though they both represent queuing at the ramp. The mean service rate drops faster for the taxi-in ramp server with increase in departures on the ramp, than for the taxi-out server with increase in arrivals on the ramp. A possible reason is that departures are prioritized relative to arrivals, since the parking gates would need to be released before the arrivals can use them.

Departure runway queues

The separation requirements between consecutive operations on a runway drive its capacity. The separation times depend on various factors such as the relative sizes and types of operation (landing or takeoff) of the aircraft using the runway, and weather conditions at the airport. The variability due to aircraft size is very small at these airports since most of the flights belong to the 'large' category of aircraft weight class (97% at CLT, 91% at DFW and 85% at EWR[2]). We assume that the departure throughput of the runway server within a

---

[2] Based on on ASPM from May-Aug 2016 for DFW, EWR and May-Jun 2016 for CLT.



(a) Mean service rate of taxi-in ramp server (b) Mean service rate of runway 36R (VMC)

Figure 2-7: Mean service rate of ramp and runway servers for CLT-NF.

small time window (5 min) depends only on: (a) the number of landings on that runway in that time window; and (b) the prevailing weather conditions (VMC/IMC) at the airport.

Fig. 2-7(b) shows the dependence of the mean service rate of the departure runway (36R) server for CLT-NF as a function of the number of landings on it, under VMC. The mean service rate is seen to decrease as the number of landings increases. An empirical model of the mean service rate for the two runway servers for CLT-NF in different weather conditions is as follows:

$$\mu_{d,r2}(t)|_{VMC} = -0.14n_{a,r2}(t) + 0.82; \quad \mu_{d,r3}(t)|_{VMC} = -0.11n_{a,r3}(t) + 0.79 \quad (2.22)$$

$$\mu_{d,r2}(t)|_{IMC} = -0.14n_{a,r2}(t) + 0.74; \quad \mu_{d,r3}(t)|_{IMC} = -0.11n_{a,r3}(t) + 0.77 \quad (2.23)$$

Here, $n_{a,ri}(t)$ denotes the number of landings on runway i in a 5-min time window containing t. The intercept of the mean service rate is lower during IMC when compared to VMC, as is to be expected. One can obtain similar models for other airports and runways configurations. One needs to note that the runway service time distributions can also vary across airports because of different procedures and fleet mix as seen in Fig. 2-8.

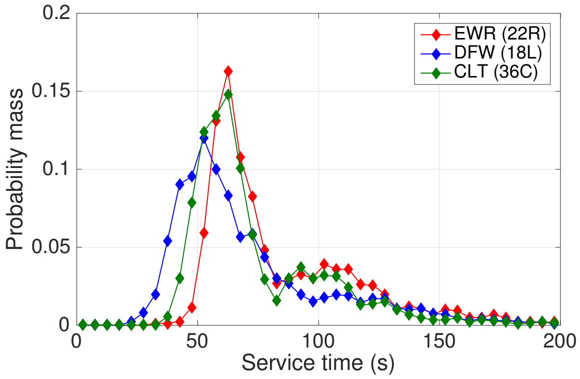


Figure 2-8: Service time distribution for departure runway server conditioned on no landings in the 5 min window for runways at three different airports.

Runway crossing queue

For CLT, flights that land on the leftmost runway (18R/36L) need to cross the active runway (18C/36C) before they can reach their gates. Air traffic controllers need to accommodate the crossing aircraft in between takeoff and landing operations, leading to the formation of a runway crossing queue (Fig. 2-1). The runway crossing queue is modeled for CLT since one is interested in the arrival ramp queue. A similar analysis can be performed for the other airports.

The distribution of the time between two successive crossings, obtained over intervals with a non-empty runway crossing queue is shown in Fig. 2-9. It indicates a bimodal service time distribution. The initial part of the distribution (< 50s) corresponds to successive crossing in one platoon, i.e. multiple crossings without a take-off/landing between them. The latter part of the distribution (> 90 s) corresponds to successive crossings that had at least one take-off/landing in-between them. As the number of runway crossings in an interval was not well-correlated with either the departure runway or the runway crossing queue lengths, we assume a constant mean service rate for the runway crossing server. The mean service rate is estimated to be 0.48 aircraft/min. Since runway crossings can occur at two points at CLT, the effective service rate is approximated to be $\mu_{a,r_2} = 2 \times 0.48 = 0.96$ ac/min. Although runway crossing can sometimes occur at just one point, we make an assumption that the runway crossing always utilize two points because it is challenging to predict the runway crossing procedure. Moreover, the initial part of the service time distribution (< 50s) can be used under low demand for takeoffs/landings, and the complete distribution can be used during periods of high demand. However, this distinction does not yield significant performance gains in practice, since runway 36C is almost always busy at CLT.

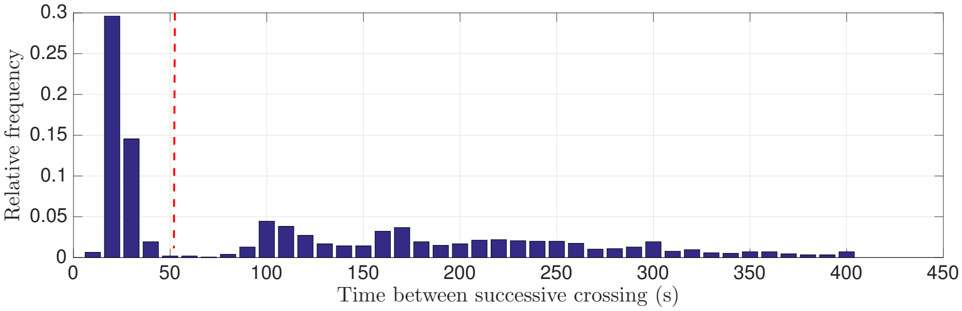
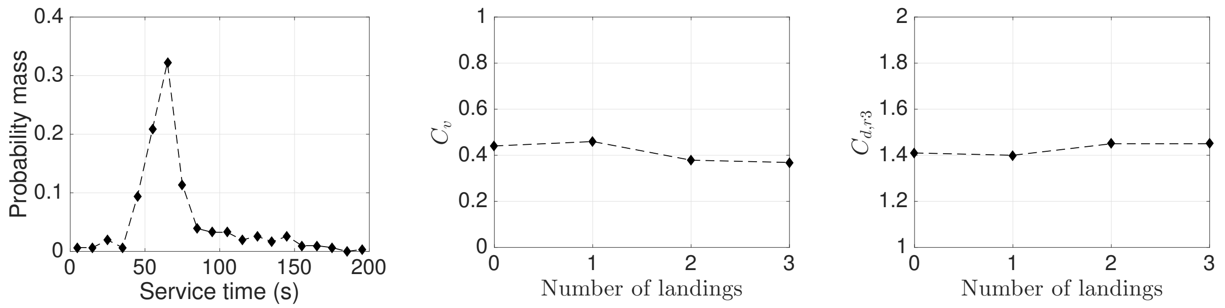


Figure 2-9: Distribution of the time between successive crossings, when there is pressure on the runway crossing server for CLT-NF.

Variability of service times

Even if the mean service rates of servers are conditioned on different parameters to capture dependencies, the service process is inherently stochastic and shows some variability. This variability affects the utilization function of the server, and is captured by the parameter C in the queue model (Eq. (2.8)). Fig. 2-10(a) shows the empirical service time distribution obtained from data for an illustrative departure runway . The mean service time is around 73 s, and the standard deviation is around 32 s, resulting in a coefficient of variation (C_v) of 0.44. Fig. 2-10(b) shows that C_v as a function of number of landings on the runway in a 5-min window is almost a constant. The value of C that is used to approximate the utilization function $G(x)$ therefore also does not vary significantly with the number of landings (Fig. 2-10(c)), and can be approximated by a constant C. Similarly, it was found that treating $C_i(t)$ as a constant for all the other queues was a reasonable approximation.



(a) Service time distribution (landings =0)

(b) Coefficient of variation of the service time distribution

(c) C value

Figure 2-10: Service time of the departure runway server 36R at CLT in VMC conditions.

Computing the taxi-time

The wait times of aircraft entering the queue are determined using the predictions of queue length and time-varying mean service rates. Let $x(t_{in})$ be the predicted queue length at the time when the aircraft enters the queue ($t = t_{in}$), $\mu(t)$ the mean service rate of the server, and Δt an appropriately small time-step. Then, the wait time (W) for an aircraft that enters the queue at t_{in} is estimated as follows:

```
$Q = x(t_{in})$;
$q = Q$; $t = t_{in}$; $W = 0$;
while $q > 0$ do
    $q = q - \mu(t)\Delta t$;
    $W = W + \Delta t$;
    $t = t + \Delta t$;
end
$W = W + q/\mu(t)$;
```

Algorithm 1: Calculation of wait time in a queue.

The predicted taxi time is the sum of unimpeded travel time and wait times in the different queues. To summarize, the proposed queuing model can predict surface queue lengths and taxi-out times, given the pushback times, landing times and weather conditions, as well as the appropriate initial conditions on the queue lengths.

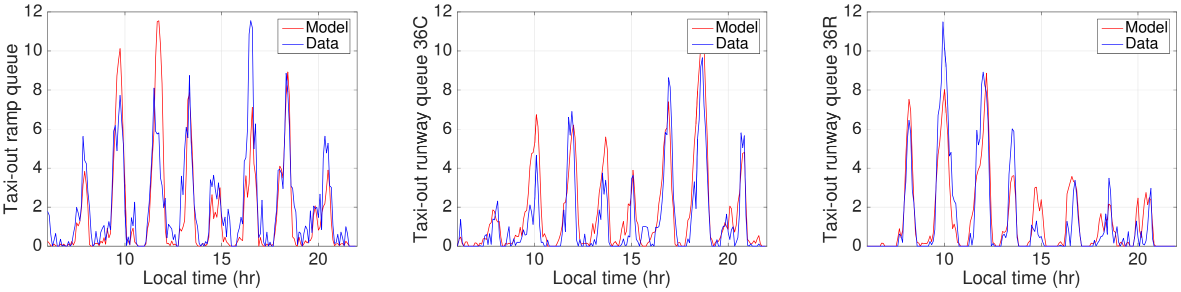
2.2.5 Predictive performance of the queuing model

The actual pushback times, landing times, and weather conditions that are required as model inputs were obtained from historical data. The pushback times and landing times were converted into the pushback rate ($u_{d_{r_i}}(t)$) and the landing rate ($u_{a_{r_i}}(t)$), respectively. The pushback times conditioned on runway assignments were binned into 5-min windows for the entire day, and an average pushback rate was determined for each time-window. The landing rate is computed in a similar way. The unimpeded times were obtained by considering the 10^{th} percentile of the taxi-time distributions. The queuing dynamics are numerically integrated forward in time from the beginning of the day with a time discretization of 1-min. Finally, the taxi times are determined after computing the queue lengths.

Prediction performance on an illustrative individual day

For illustrative purposes, we consider queue length and taxi time predictions for a typical day in the test dataset at CLT (Jun 25, 2016; VMC conditions). Figure 2-11 shows the queue length predictions for the taxi-out ramp queue and departure runway queues. The queue lengths shown are averaged over a 5-min time-window. The data corresponds to a time-based definition of queue length, in which an aircraft is said to be in the runway queue if it has spent unimpeded spot-to-runway time after exiting the spot but is yet to take-off (a similar definition for the ramp queue). Our analysis has found that this time-based definition is consistent with the observed physical queue. We see a good qualitative match between the model predictions and observed data. The prediction errors are comparatively higher for the ramp queue when compared to the runway queues, which is expected since there is more uncertainty in the ramp area (because of the variability in the taxi-routes and complex traffic movement). The model assumed a single server for the departure ramp queue (since that is generally observed); the prediction errors could therefore be higher during the occasional periods when more than one spot is simultaneously used by departing flights.

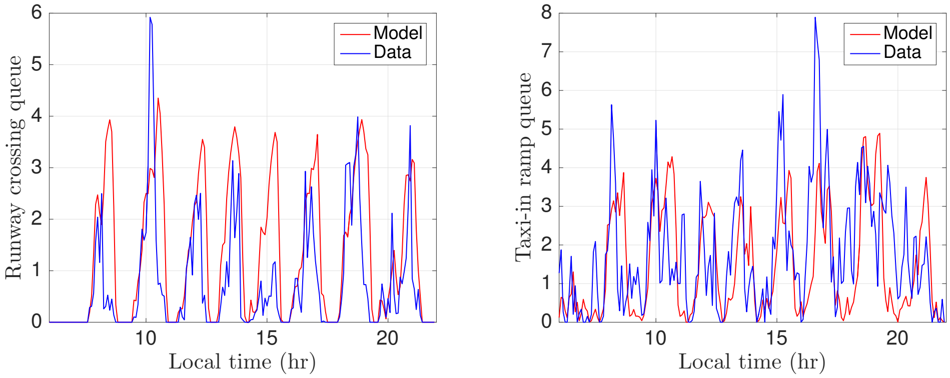
A comparison of the predicted (model) and actual queue lengths for flights taxiing in is shown in Fig. 2-12. Once again, we see a reasonable agreement between the model and data,



(a) Taxi-out ramp queue (b) Taxi-out runway queue for 36C (c) Taxi-out runway queue for 36R

Figure 2-11: Queue length predictions for taxi-out ramp and runway queue for a typical day (Jun 25, 2016) at CLT.

especially considering the fact that there is considerable operational variability in runway crossing procedures. The reasons for the prediction errors for taxi-in ramp queue are similar to those mentioned for the taxi-out ramp queue.

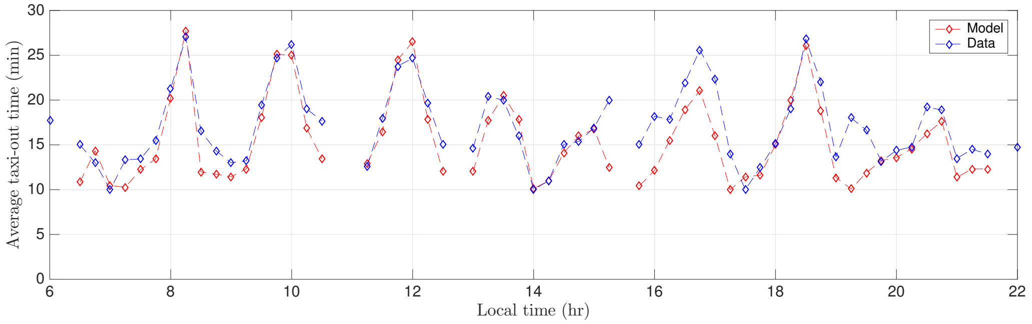


(a) Runway crossing queue (b) Ramp queue

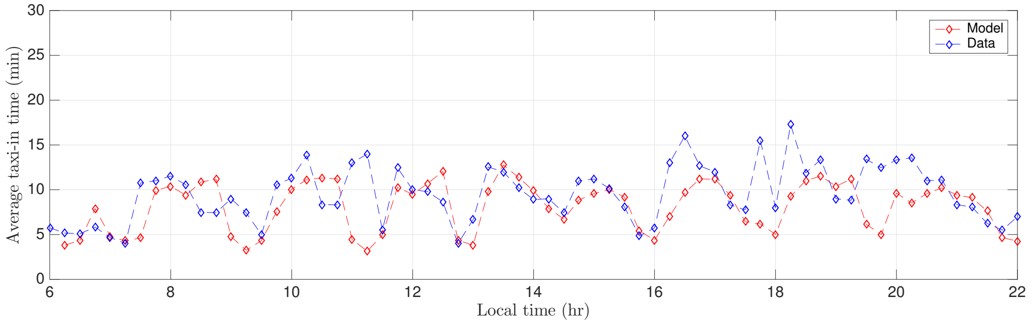
Figure 2-12: Queue length predictions for taxi-in ramp and runway crossing queue for a typical day (Jun 25, 2016) at CLT.

Fig. 2-13(a) shows the predicted and actual average taxi-out times. Each data point corresponds to an average over a 15-min window and the flights are binned based on the actual out-gate time. While the predictions generally match the observed values, we note that large errors in the taxi-out time prediction correspond to errors in the queue length prediction. A similar trend is seen in the taxi-in time predictions shown in Fig. 2-13(b). The taxi-in flights are binned into 15-min windows based on the actual landing times.

A comparison of the predicted and observed departure runway queue lengths at DFW



(a) Taxi-out time



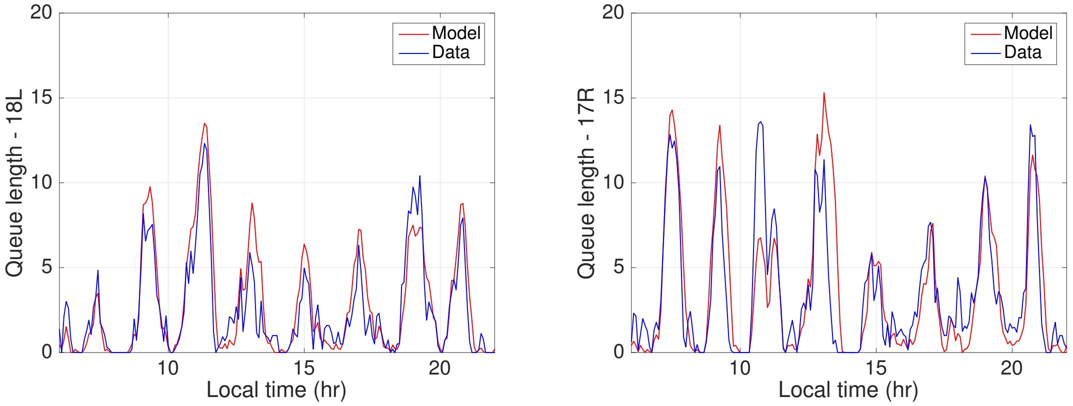
(b) Taxi-in time

Figure 2-13: Taxi times averaged over 15 min interval for a typical day at CLT in North-Flow configuration (Jun 25, 2016).

for a typical day is shown in Fig. 2-14. The taxi-out times for this particular day, averaged over 15-min windows, are shown in Fig. 2-15. These figures also show a good match between the predictions and observed values.

Aggregate Prediction Performance over Test Data Set

Here we present the aggregate error statistics for all the three airports (EWR, DFW and CLT), computed for an independent test set. Table 2.2 shows the aggregate error statistics of taxi-out time prediction for individual flights. Flights with taxi-out times greater than 50 min were not included while computing the statistics. The mean errors (ME) are found to be small relative to the mean taxi-out times and negative at all three airports. The mean absolute error (MAE) of the taxi-out time predictions is similar to the results obtained from other probabilistic queuing models [157]. For example, the MAE of the taxi-out time



(a) 18L (b) 17R

Figure 2-14: Comparison of model predictions and observed queue lengths on a typical good weather day at DFW (07/20/2016).

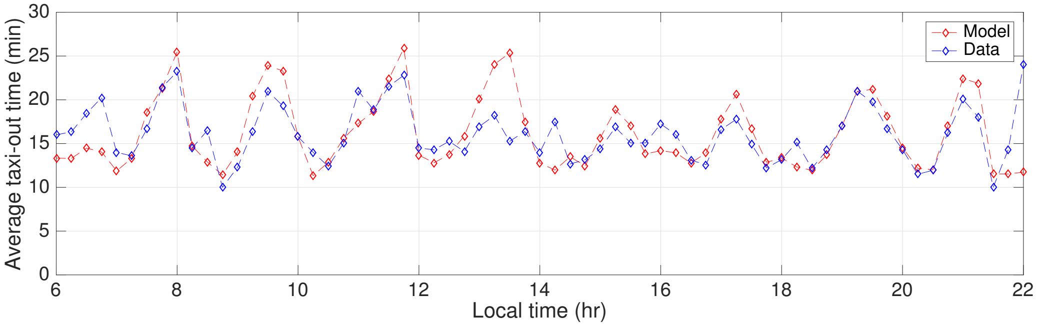


Figure 2-15: Taxi-out times averaged over 15-min intervals on 07/20/2016 at DFW.

at EWR using a probabilistic queuing model was reported to be around 5.5 min in [157], which is similar to the result using our proposed model. A high value of MAE indicates that the takeoff sequence does not necessarily follow the first-come-first-served assumed by the queuing model. Figure 2-16 compares the predicted and actual taxi-out time distributions. We note that the right tail of the taxi-out time distribution is not captured by the queuing model, resulting in a negative value of the mean error. This right-tail is more pronounced for EWR and CLT, where in addition to queuing delays, surface gridlock can cause further delays that are not accounted for by the queuing model. Table 2.3 shows the prediction error statistics for individual flights grouped by different quartiles of the actual taxi-out time. For flights in the first quartile (0-25th percentile), the MAE is small and ME is positive. The model slightly over-predicts taxi-out time in this range because it assumes that the unimpeded time is the 10th percentile of the taxi-out time distribution. For the flights belonging to the upper quartile (75th-100th percentile), the magnitude of the ME is negative and large. From the point of view of departure metering, positive errors correspond to over-prediction of the taxi-out time, leading to overly aggressive hold decisions, loss in runway utilization, and delays in wheels-off time. By contrast, negative taxi-out time errors will yield lower metering benefits, but there would be no loss in throughput and no additional delays.

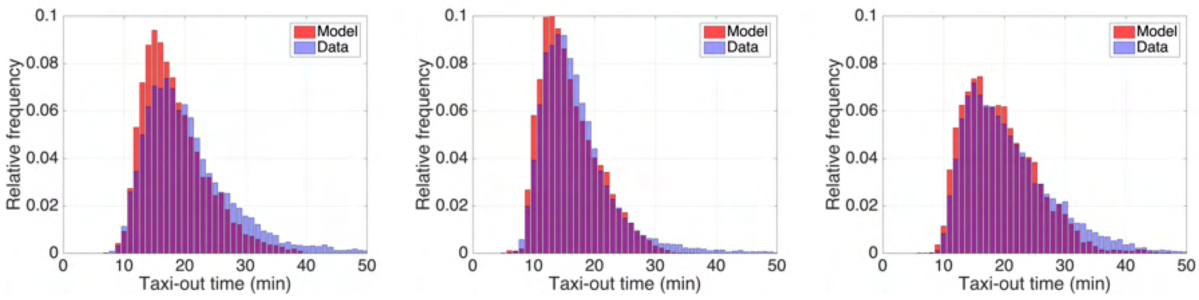
Airport	**Configuration**	**Number of departures**	**Taxi-out time (min)**			**Percentage of flights with $\lvert$error$\rvert < 5$ min**
			Mean	**ME**	**MAE**	
EWR	NF	2,596	21.3	-1.4	4.9	61.9
	SF	6,877	21.5	-1.9	5.3	61.1
DFW	NF	2,769	18.5	-1.2	3.9	72.3
	SF	19,207	17.4	-1.1	4.2	71.5
CLT	NF	7,464	20.1	-1.4	4.4	69.0
	SF	6,120	20.1	-1.9	5.2	61.0

Table 2.2: Error statistics of the taxi-out time predictions from the analytical queuing model on a test set.

Airport	**Configuration**	0-25th **%ile**		25th-50th **%ile**		50th**-**75th **%ile**		75th**-**100th **%ile**	
		ME	**MAE**	**ME**	**MAE**	**ME**	**MAE**	**ME**	**MAE**
EWR	NF	2.9	3.6	0.8	3.5	-1.7	4.6	-7.2	8
	SF	2.5	3.2	0.7	3.4	-1.5	4.4	-9.2	10.1
DFW	NF	1.4	2.5	0.3	2.9	-0.7	3.5	-5.1	6.4
	SF	1.7	2.5	0.8	2.9	-0.4	3.5	-5.8	7.4
CLT	NF	1.8	2.9	0.3	3.2	-1.3	3.9	-6.3	7.5
	SF	2.5	3.3	0.5	3.5	-2.3	4.6	-7.7	9.1

Table 2.3: Error statistics for the taxi-out time predictions for individual flights from the queuing model based on the test set for different quartiles of the actual data.

Table 2.4 shows the breakup of the taxi-out time prediction errors as well as taxi-in time errors for CLT-NF, computed using the test set. We can see that the prediction errors are higher in the ramp area compared to the active movement area for both taxi-out as well as taxi-in. The higher errors in the ramp area is expected because of more uncertainty in operations in that region.



(a) EWR-SF (b) DFW-SF (c) CLT-NF

Figure 2-16: Comparison of the predicted and actual taxi-out time distributions.

Taxi-time	**Segment**	**Mean (min)**	**ME (min)**	**MAE (min)**	**Percentage of flights with |error| < 5 min**
Taxi-out	Gate → spot	9.68	-0.87	3.08	82
	Spot → rwy	10.40	-0.58	2.70	86
Taxi-in	Rwy → spot	4.49	-0.09	1.38	96
	Spot → gate	6.06	-0.99	3.34	83

Table 2.4: Error statistics for the taxi-out and taxi-in time predictions for CLT-NF, based on 7,484 departures and 8,477 arrivals in the test dataset.

2.2.6 Applicability to other airports

The queuing models are applicable to other airports as well. For each airport, one needs to determine the congestion hotspots and represent it using an appropriate queuing network. To demonstrate the applicability, we have developed and validated similar queuing network models for Charles de Gaulle Airport (CDG) in Paris and Changi (SIN) in Singapore [22, 135]. Table 2.5 shows a summary of the error statistics for taxi-out time predictions of individual flights using the queuing model for CDG and SIN. The results shown in the table correspond to the more frequently used runway configuration at these airports (West-flow at CDG and North-flow at SIN). A more detailed description of the queuing representation and validation results for CDG is presented in Appendix A and details for SIN can be found in [135].

Table 2.5: Error statistics of the taxi-out time predictions on a test set using the queuing model.

Airport	**Number of departures**	**Taxi-out time (min)**			**% of flights**
		Mean	**ME**	**MAE**	**|error| < 5 min**
CDG	14,100	13.3	-0.3	3.0	82.4
SIN	20,171	17.1	-0.1	3.3	75.3

2.3 Integration of surface and airspace models for CLT

We can extend the airport surface queuing network model to include operations in the terminal airspace. Immediately after takeoff, a flight is vectored to begin following a particular Standard Instrument Departure (SID) route, a standardized set of navigational aids (waypoints and fixes) designed to ease the transition into the en route stream. Due to the lack of predictability in departure airspace, controllers are forced to maintain conservative spacing between aircraft, reducing the airspace efficiency. We develop a model for the departure airspace, and combine it with the surface model in order to obtain an integrated surface-airspace model. We would like to highlight that many aircraft do not exactly follow the SID profiles but the likely operationally used paths can be determined from empirical flight track analysis. The goal of the integrated model is to predict the transit time from pushback at the gate to the final departure fix (FDF). The resulting models can potentially improve predictability, and support the objective of inserting departures more efficiently into the overhead stream [13]. There has also been recent research efforts into integrating surface and terminal-area operations to reduce congestion [122, 107, 95]. A detailed analysis on this topic has been published in our paper [24], here we present an overview of the key results.

Fig. 2-17 shows a schematic of the six major Standard Instrument Departure routes (SIDs) that handle around 62% of the departures from CLT. Flights generally choose one of the SIDs depending on their destination airport when filing their flight plan. For example, flights going to airports in the North-East US (such as BOS, JFK, PHL) would file the BARMY1 SID. We develop a transit time prediction model for each one of the six SIDs, and integrate it with the queuing network model of the surface, to obtain a surface-airspace transit time prediction model. The transit time to the final departure fix can potentially depend on various factors, such as the en route traffic, distance to the fix, weather, and runway configuration.

Our analysis indicated that traffic density is presently not a major factor impacting departure airspace transit times at CLT [24]. The integrated surface-airspace model takes the prevailing runway configuration and CLT airport weather into consideration, and predicts

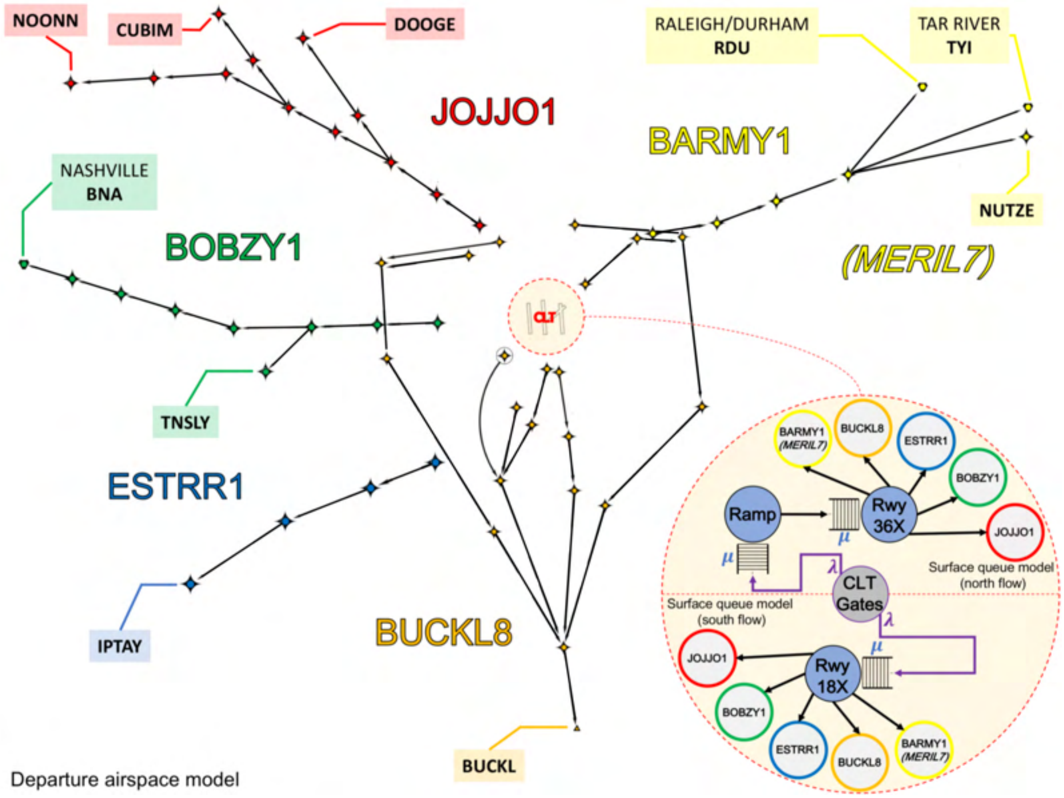


Figure 2-17: High-level system diagram of the integrated surface-airspace model for predicting gate to final departure fix transit times (Image courtesy of Max Z. Li [24]).

the pushback to Final Departure Fix (FDF) transit time for a CLT departure, if the flight uses one of the SIDs currently considered in the model. We would like to highlight that weather plays an important role in the accuracy of these predictions. The pushback to FDF transit time is computed as the sum of the predicted pushback to takeoff time (via the surface queue model), and the mean transit time from takeoff to the FDF (via the airspace model). A high-level schematic of the integrated surface-airspace transit time prediction model is shown in Fig. 2-17.

Table 2.6 contains the error statistics and other metrics related to the integrated surface-airspace transit time model. The normalized root-mean-square error (NRMSE) is the RMSE normalized with respect to the corresponding mean transit time from pushback to the final departure fix. The intuition behind the normalization is that the farther away a final departure fix is, the more variation that naturally occurs in the transit time. We also provide the percentages of predicted transit times that fall within ±5 min and ±10 min of the actual transit times. We see that for all scenarios except for the JOJJO1.NOONN transition, typically 50%–60% of predicted transit times are within five minutes of the actual transit times. We note that JOJJO1.NOONN is an anomaly due to the very small number of departures that flew the transition.

SID Procedure	**Distance (nm)**	**MTT (min)**	**RMSE (min)**	**NRMSE**	**|error|** <**5 min, %**	**|error|** <**10 min, %**	**Mean error**	**Mean |error|**	**Counts**
							(min)		
MERIL7.MERIL	78.1	34.4	7.0	0.2	56.0%	86.3%	-0.1	5.4	2,008
BUCKL8.BUCKL	41.4	29.6	6.7	0.2	56.8%	86.3%	1.7	5.2	2,913
ESTRR1.IPTAY	106.8	40.3	6.9	0.2	57.1%	87.1%	0.1	5.3	2,013
BARMY1 (combined)	N/A	42.5	7.5	0.2	56.9%	83.9%	-0.8	5.6	1,326
BARMY1.RDU	113.1	41.5	7.8	0.2	54.2%	82.0%	-1.3	5.9	818
BARMY1.TYI	165.2	45.9	6.8	0.2	59.6%	87.1%	-0.9	5.1	255
BARMY1.NUTZE	164.4	42.6	7.1	0.2	62.9%	87.0%	1.1	5.1	253
BOBZY1 (combined)	N/A	60.2	7.7	0.1	51.2%	82.1%	-0.2	5.9	3,503
BOBZY1.BNA	285.7	66.2	7.6	0.1	51.6%	82.9%	0.4	5.9	2,860
BOBZY1.TNSLY	52.2	34.1	8.1	0.2	49.5%	78.5%	-2.9	6.3	643
JOJJO1 (combined)	N/A	43.5	6.9	0.2	59.9%	87.2%	-0.2	5.2	2,084
JOJJO1.DOOGE	124.6	41.7	6.7	0.2	60.2%	87.6%	-0.1	5.0	1,305
JOJJO1.CUBIM	141.2	46.0	6.8	0.2	60.3%	87.7%	-0.5	5.2	766
JOJJO1.NOONN	208.4	46.5	22.0	0.5	15.4%	15.4%	-1.2	19.00	13
CLT Dep. Airspace	**N/A**	**43.0**	**7.1**	**0.2**	**55.8%**	**85.3%**	**0.20**	**5.46**	**13,847**

Table 2.6: Surface-airspace integrated queue model departure transit time prediction statistics.

It is worth noting that the mean absolute error for the integrated surface-airspace model is only slightly higher when compared to the surface model for predicting the taxi-out times, even though the mean transit time from pushback to the final departure fix is at least twice as long as the mean taxi-out time. This indicates that airport surface operations, at least in the case of CLT, are more unpredictable compared to terminal airspace operations, and highlights the need to accurately model the airport surface. It would be interesting to see if the same behavior holds true for airports with more constrained and competitive departure airspace capacity, such as airports in the New York metroplex.

One could extend this model to accommodate additional departure fixes that are generally used in non-normal operations in adverse weather conditions. Such models would be helpful in determining the optimal departure fix for every flight during non-normal operations. For example, if there are high delays because of bad weather over a particular departure fix, the question is whether to wait on the ground and use the standard departure fix or use an alternate departure fix that might take a longer airborne time but might avoid long ground holds. We could answer such questions using extensions of this model as a basis.

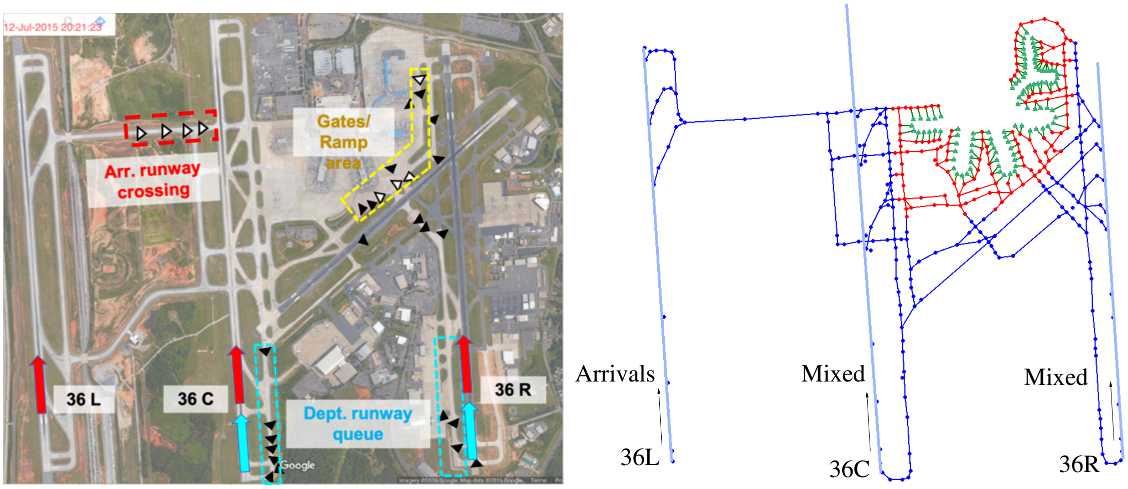
2.4 Comparison to microscopic simulation

We present a comparison of the estimates obtained from the "macroscopic" queuing network model with "microscopic" simulations. We first briefly describe our in-house microscopic simulator that was developed in MATLAB.

2.4.1 Microscopic simulator

Microscopic simulation refers to modeling individual flight trajectories on the airport surface. The runways and taxi-paths on the airport surface are represented using a node-link network. Figure 2-18 shows a node-link network for CLT in North-Flow runway configuration. The network consists of 581 nodes and 506 links, with 102 gates. The links in the active-movement-area (AMA), ramp area and pushback area, are indicated by blue, red and green, respectively. The departures enter the simulation environment at the time of pushback and a unique taxi-path is determined based on their spot and runway assignment, that is obtained from the historical flight data. The gate-to-spot and spot-to-runway taxi path is based on the shortest distance path in the node-link network. We found that the shortest taxi-path matches well with the historical track data for the active movement area using ASDE-X data. For the arrivals, we consider multiple taxi-paths for each runway-spot pair to account for multiple runway exits and runway crossing points. The taxi-path for the arrivals was randomly picked from a possible set of paths, with weights assigned to each path based on the historical frequency of occurrence. The runway separation times are sampled from the empirical distribution obtained from data. The runway separation times are conditioned on the aircraft weight class and aircraft type (sequence of arrivals or departures).

The following model is considered for the aircraft dynamics: aircraft accelerates at a constant value until it reaches its nominal taxi-speed and it can come to a sudden stop if needed for safe separation. Aircraft taxi-speed and pushback time were determined from historical flight data. Figure 2-19(a) shows a plot of the gate-to-spot distance against the

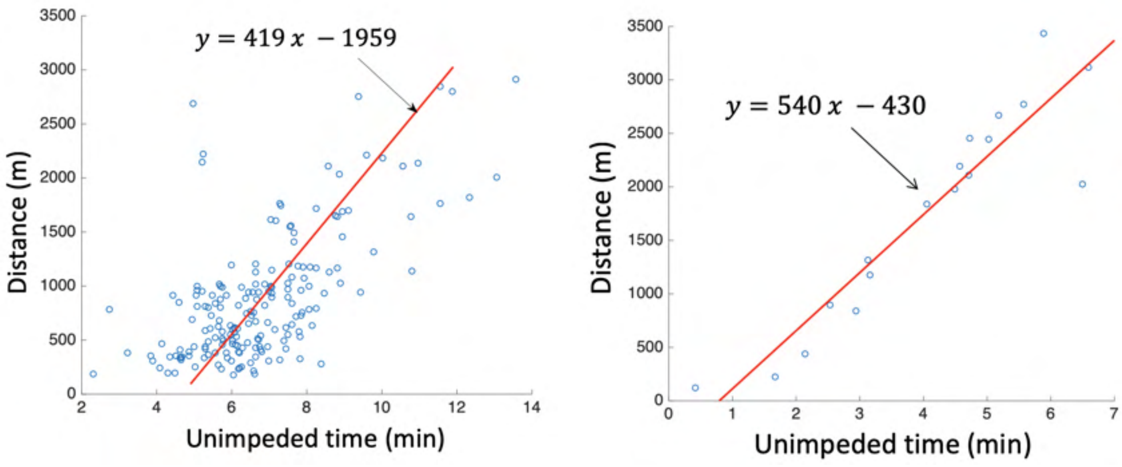


(a) Airport layout (b) Node-link network

Figure 2-18: Airport layout and corresponding node-link representation for CLT-NF.

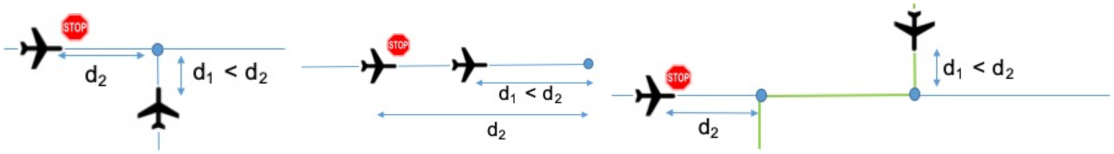
unimpeded gate-to-spot time for different gate-spot combinations. The unimpeded time is obtained as the 10th percentile of the empirical gate-to-spot time distribution for each gate-spot pair. The slope of the linear fit determines the nominal ramp taxi-speed and the x-intercept represents the average pushback time. Similarly, Fig. 2-19(b) shows a plot of spot-to-runway distance against unimpeded spot-to-runway time for different spot-runway combinations. The slope of this linear fit yields the AMA taxi speed. The nominal taxi-speed in the ramp area and AMA was determined to be 7m/s and 9 m/s, respectively. The average pushback time was determined to be 4.7 min. The pushback speed is then estimated from the pushback time using the average pushback distance at the gate and is found to be 0.13 m/s. Acceleration value for the aircraft during the taxi phase is considered to be 0.2 m/s^2. The acceleration during takeoff is assumed to be 1.6m/s and deceleration after landing is assumed to be 1.5 m/s. The landing speed just after touchdown is assumed to be 66 m/s.

The other key aspect of the simulator is to ensure safe separation during taxiing. Three basic rules are used to avoid loss in separation during taxiing as shown in Fig. 2-20. If two aircraft, that are approaching the same node, have their separation distance smaller than a threshold value, then the aircraft that is farther away from the node is stopped and the other aircraft is allowed to proceed (Figs. 2-20(a)-2-20(b)). If a set of common links is shared by



(a) Ramp area: gate-to-spot (b) AMA: spot-to-runway

Figure 2-19: Plot of distance vs unimpeded time for departures.



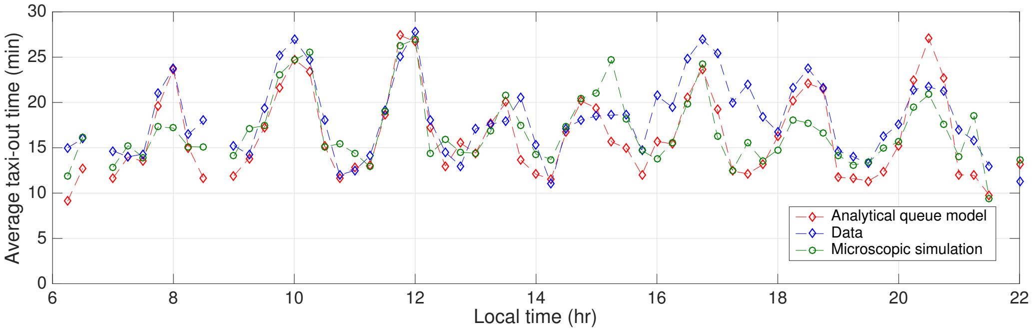
(a) Node conflict-Type 1 (b) Node conflict-Type 2 (c) Link conflict

Figure 2-20: Safe separation rules applied in the simulator.

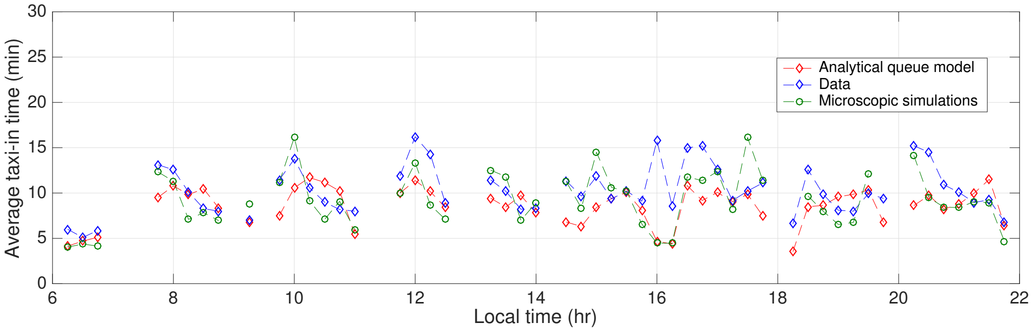
two aircraft and if they are supposed to move in the opposite direction on those common links, then the aircraft that is farther away from the closest node in the common link is stopped from entering the common link till the other aircraft passes through the common links (Fig. 2-20(c)). The threshold distance for safe separation is considered to be 30m for the ramp area and 80 m for the AMA. The simulator steps forward in discrete time steps (1 s) by updating positions of aircraft and their taxi-speeds, and ensuring safe separation rules are followed. Departing (Arriving) flights are introduced into the simulator at the pushback (landing) time and they are removed from the simulator after take-off (reaching gate).

2.4.2 Performace of the microscopic simulations

We compare the performance of the microscopic simulations with the analytical queuing model. Figure 2-21 shows the comparison of the average taxi-out and taxi-in time for a



(a) Taxi-out time (min)



(b) Taxi-in time (min)

Figure 2-21: Taxi-times averaged over 15-min windows for a typical day at CLT-NF (07/12/2015)

typical day of operation at CLT-NF. We see that the simulations are close to the actual data, however, the analytical model performs slightly better. The error statistics for the taxi time predictions considering individual flights (Table. 2.7) also indicate that the analytical model performs better than the microscopic simulations. Moreover, the computational time for the microscopic simulations is orders of magnitude more than the queuing model. The queuing model takes about 1 s for predicting the taxi-time of all flights for one day of operations (5:00-23:00 hrs), whereas, the microscopic simulation takes about 1260 s. A detailed comparison of the performance considering multiple days of operation will be done in the future. However, one needs to note that the microscopic simulation has numerous internal parameters (such as speed, acceleration, stopping criteria etc.), making it challenging to tune the parameters for improved performance.

Table 2.7: Error statistics for a typical day at CLT-NF (07/12/2015) containing 672 departures and 664 arrivals.

Parameter	Microscopic simulation			Analytical queue model		
	ME	MAE	Percentage of flights with $\lvert\text{error}\rvert < 5$ min	ME	MAE	Percentage of flights with $\lvert\text{error}\rvert < 5$ min
Taxi-out time (min)	-2.0	5.0	61.4	-1.1	4.0	68.2
Taxi-in time (min)	-1.1	3.6	78.6	-1.4	3.13	83.7

2.4.3 Discussion on the modeling philosophy

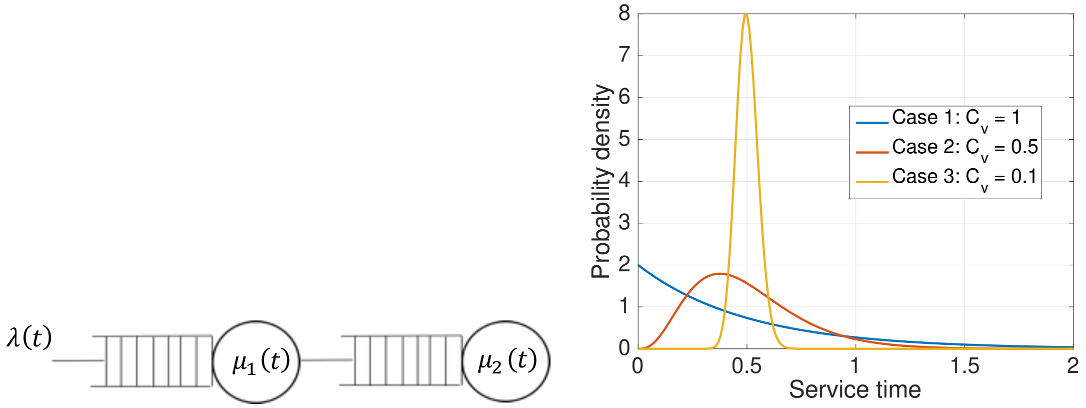
The queuing model can estimate macroscopic parameters such as queue length and taxi-time, whereas, the microscopic simulator can estimate individual flight trajectories, that are of higher fidelity. The microscopic simulator also gives the probability distribution of the quantities of interest, unlike the queuing model. However, the microscopic simulator has several drawbacks such as: (a) they are challenging to adapt to a new airport or runway configuration, (b) simulations are computationally intensive, (c) 'grid lock' that requires manual intervention and resolution[3], (d) non-analytical nature of the model makes it difficult to develop congestion control strategies using traditional approaches. The queuing model has advantages on all these fronts and is best suited to analyse current operations or interventions in terms of demand/capacity management. However, one needs to note that interventions should not largely affect the nature of flows or create new hot-spots that are not assumed in the queuing model. Some of the operational procedures that could impact the nature of flows are different sequencing for runway crossings, use of different taxi-paths and transitions during runway configuration changes.

2.4.4 Discussion on the Poisson assumption for queuing networks

In our proposed queuing network model (Eq. 2.11), we made a strong assumption that the customer arrivals into the queues follows a Poisson process. In this discussion, we present a few numerical examples to illustrate the validity of the Poisson arrivals assumption for general queuing networks. For the analysis, we consider the simplest queuing network with

---

[3] 'grid lock' refers to the situation in which aircraft comes to completle stand-still during taxi operation because of traffic congestion in all directions



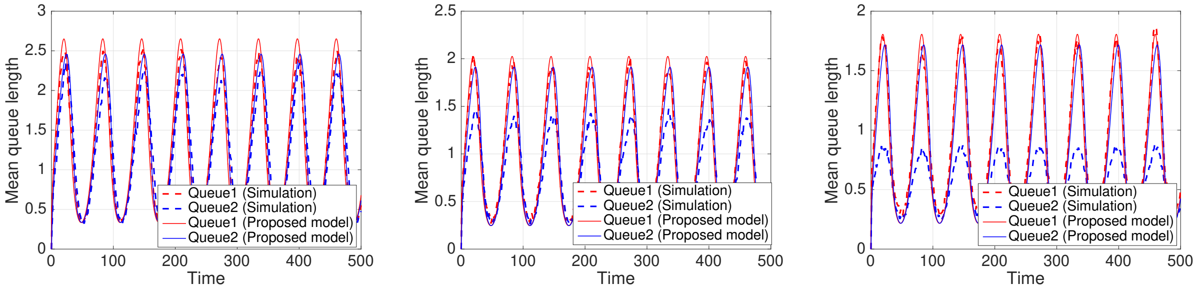
(a) Illustration of queues in series.

(b) Service time distribution for the three cases.

Figure 2-22: Illustration of queues in series and service time distribution.

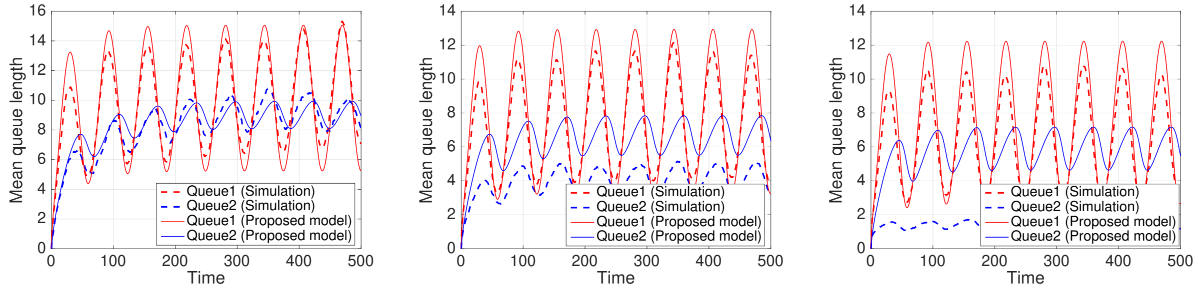
two queues in series (as shown in Fig. 2-22(a)). We consider the arrivals into the first queue to be Poisson at the rate $\lambda(t)$. For the sake of illustration, we consider Erlang service time distribution for the two servers with a constant mean service rate ($\mu_1 = \mu_2 = 2$). To study the influence of variability in service time and its impact on the Poisson assumption, we consider three cases by changing the coefficient of variation ($C_v = \frac{\sigma}{\mu}$) of the service time distribution: (a) $C_v = 1$; (b) $C_v = 0.5$; (c) $C_v = 0.1$. The three cases span highly variable service time ($C_v = 1$) to nearly constant service time ($C_v = 0.1$) as shown in Fig. 2-22(b).

Figure 2-23 shows a comparison of the mean queue lengths obtained from discrete stochastic simulations and the proposed analytical queuing network model for the three different cases considered in our analysis. We notice that for the case with a high variability in service time ($C_v = 1$), the model-predicted queue lengths match closely with the simulation results. We note that $C_v = 1$ corresponds to an exponential service, and the output of a queue with exponential service time distribution is known to be Poison under steady state condition. However, as the variability in service time is reduced ($C_v = 0.5, 0.1$), we notice that the simulation results deviate significantly from the proposed model. The absolute value of the deviations increases with reduction in C_v and increase in instantaneous utilization factor $(\frac{(t)}{\mu})$ as seen in Fig. 2-24. These observations illustrate that one needs to exercise caution, and check the validity of the Poisson arrivals assumption when applying the queuing network model to a physical system. In the case of airport operations, the queuing network model was able to yield good accuracy in predicting the queue lengths and taxi-times as shown earlier in this chapter.



(a) $C_v = 1$ (b) $C_v = 0.5$ (c) $C_v = 0.1$

Figure 2-23: Comparison of the mean queue lengths obtained from simulations and the proposed analytical queuing network model for different service time distributions. $\lambda(t) = 1 + 0.5\sin(0.1t), \mu_1 = \mu_2 = 2$.



(a) $C_v = 1$ (b) $C_v = 0.5$ (c) $C_v = 0.1$

Figure 2-24: Comparison of the mean queue lengths obtained from simulations and the proposed analytical queuing network model for different service time distributions. $\lambda(t) = 1.9 + 0.5\sin(0.1t), \mu_1 = \mu_2 = 2$.

Chapter 3

Congestion control for airport surface operations

3.1 Introduction

Departure metering, in which departures are appropriately held at the gate in order to reduce taxi-out time, while ensuring no adverse impact on the airport throughput, has been shown to be an effective congestion management technique [179, 158, 154]. An aircraft saves fuel while waiting at the gate with its engines off compared to idling in a taxi queue with its engines on. In this chapter, we present two approaches to departure metering based on the queuing model for the airport traffic that was presented in the previous chapter: NASA's Airspace Technology Demonstration-2 (ATD-2) logic [179], and a new approach based on optimal control

Additionally, we also investigate the impact of demand uncertainty on the benefits of departure metering using empirical data from three major US airports (EWR, DFW, CLT). The predicted departure demand a few hours prior to operations is based on the Scheduled Off-Block Times (SOBTs) published by the airlines. Closer to the time of operation, departure demand can be based on the Earliest Off-Block Times (EOBTs)[1], which represent

---

[1] Note: We adopt the nomenclature used by the US Surface Collaborative Decision Making (S-CDM) community for different data elements. The nomenclature used by EUROCONTROL's Airport-CDM is

the times at which flights are expected to be ready for pushback. In the US, airlines are now encouraged to publish EOBT information for their flights, and they are expected to dynamically update each of them until the flight pushes back from the gate [63]. The dynamic updates are intended to make the EOBT more representative of the actual demand compared to the static SOBT. Most departure metering concepts assume the availability of EOBT information while making tactical gate-hold decisions [63, 60]. While prior efforts have focused on near real-time departure metering (for example, over the next 15 minutes), departure metering concepts call for assigning the gate-hold times over longer time-horizons, in order to improve the predictability of operations [115]. The EOBT is also a key input to the calculation of downstream events within the S-CDM concept [139]. The actual pushback time (also referred to as the *call-ready* or *push-ready* time) of a flight often differs from the published EOBT due to several reasons. The accuracy of the published EOBT depends on an airline's operating policies with respect to connecting passengers, and aircraft and crew schedules, as well as CDM procedures [63, 26]. Despite the significant influence of EOBT accuracy on surface operations, there has, to the best of our knowledge, been limited research on characterizing its magnitude and impact of the resulting uncertainty on the efficacy of departure metering [27, 126].

The analyses presented in this chapter are of potential benefit to multiple stakeholders such as airlines, airport operators, and air navigation service providers (such as the FAA or EUROCONTROL). For example, emerging terminal automation systems such as the FAA Terminal Flight Data Manager (TFDM) rely on accurate EOBTs to be published for input to their departure metering functions. Improvements in EOBT accuracy will require investments on the part of the airlines in terms of new technologies and procedures. Our analyses help estimate the returns of these investments in terms of taxi-out time and fuel burn savings.

---

significantly different; for example, EOBT is referred to as Estimated Off-Block Time [63, 60].

3.2 Control algorithms for departure metering

During periods in which demand exceeds capacity, queues build up on the airport surface, leading to an increase in taxi-out times. The idea behind departure metering is to tactically assign hold-time for departures at the gate during periods of peak traffic, so that smaller queues are formed on the airport surface, leading to a reduction in taxi-out time. The challenge is to assign appropriate hold-times to maximize benefits in terms of taxi-out time reduction, while maintaining runway utilization. Departure metering is an integral part of the FAA's Terminal Flight Data Manager (TFDM), a NextGen initiative [69]. Several algorithms have been previously proposed to determine the appropriate hold times at the gate [158, 146, 18, 102]. In this chapter, we consider two approaches to departure metering: The first one based on NASA's ATD-2 logic, and the second based on an optimal control formulation.

3.2.1 Airspace Technology Demonstration-2 (ATD-2) logic

The ATD-2 logic for departure metering computes a gate-hold time for each flight based on its predicted taxi-out time as follows [179]:

$$TOBT = \max(EOBT, TTOT - UTT - Y), \tag{3.1}$$

where $TOBT$ is the Target-Off-Block-Time or the new gate release time decided by the controllers, $TTOT$ is the Target Take-Off-Time, UTT is the unimpeded time to take-off that depends on the gate-runway pair, and Y is the excess queue time buffer [179]. The target take-off time is computed by adding the predicted taxi-out time to the flight's EOBT. In our work, we use the queuing model presented earlier to obtain the taxi-out time prediction for each flight. In other words, the hold time assigned to each flight is the predicted wait time in queue for that flight minus the excess queue time buffer. The idea is to transfer the predicted wait time in the queues to a gate-hold time, saving fuel. The excess queue time buffer (Y) is subtracted to account for errors in the taxi-out time prediction. The buffer value (Y) is a constant design parameter that is chosen for each runway configuration. It is important to choose the optimal buffer parameter. If the buffer parameter is too high, it will lead to decreased benefits; if the buffer parameter is too low, it could lead to reduced runway utilization. An illustration of the ATD-2 logic comparing it with the default scenario with no metering is shown in Fig. 3-1.

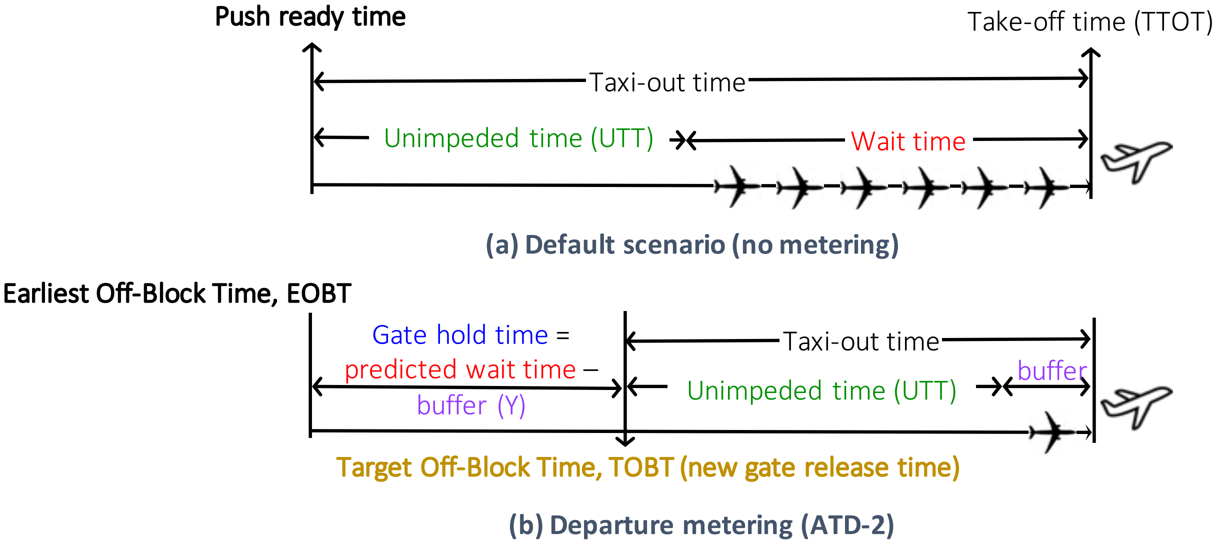


Figure 3-1: Illustration of the ATD-2 logic for departure metering.

Given the weather condition (IMC/VMC), landing times for arrivals, runway assignments and EOBT time for departures, the queuing model can predict the taxi-out time for each flight based on the current queue length at an airport. The hold time is then assigned to each flight based on its predicted taxi-out time. To have higher predictability in the system, the hold decisions are made T_p minutes prior to a flight's EOBT (where T_p is the planning horizon).

3.2.2 Simulation environment for evaluation

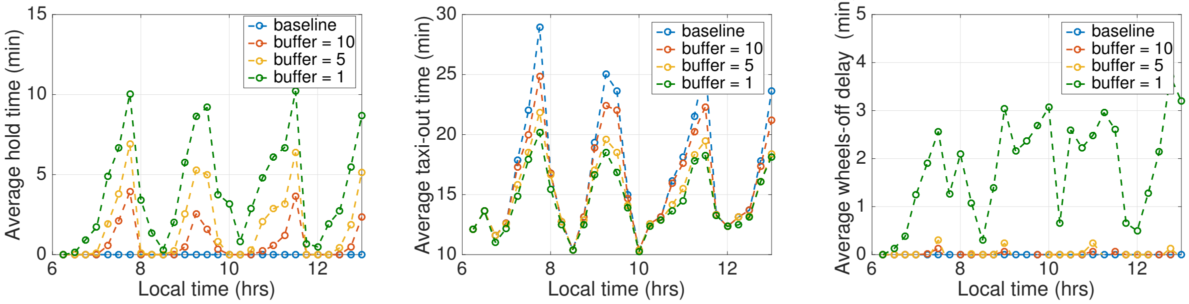
The departure metering approaches are evaluated using a simulation of airport surface operations. The simulators are based on discrete queuing network models (as described in Section 2.2), with the service time for each server being sampled from an empirical distribution. The empirical service time distributions are a function of the airport weather and traffic (such as number of landings on the runway and traffic on the ramp) as we had discussed earlier. The simulations are repeated multiple times to obtain consistent statistics (a Monte Carlo simulation with 20 runs). We consider a 15 day period for each airport-runway configuration pair to evaluate the benefits of departure metering. Table 3.1 shows the error statistics for the taxi-out time predictions from the simulator in the baseline case without any metering by comparing it with the actual data. Although we consider a 15-day period for each airport-runway-configuration pair, the total number of departures vary between NF and SF, because the total time duration operating in a particular configuration is different. The results indicate that the errors are small relative to the mean taxi-out times, but their magnitudes are slightly larger than the analytical queuing model.

Airport	**Configuration**	**Number of departures**	**Taxi-out time (min)**			**Percentage of flights with |error| $<$ 5 min**
			Mean	**ME**	**MAE**	
EWR	NF	4,117	21.6	0.4	5.4	60.4
	SF	4,493	21.0	0.5	5.4	59.0
DFW	NF	6,377	18.7	-0.4	4.2	70.6
	SF	9,212	17.3	0.6	4.9	64.4
CLT	NF	6,447	20.1	1.1	4.6	64.2
	SF	4,493	19.7	-1.0	5.0	63.8

Table 3.1: Error statistics for the baseline taxi-out time predictions from the queuing simulation over a 15 day period.

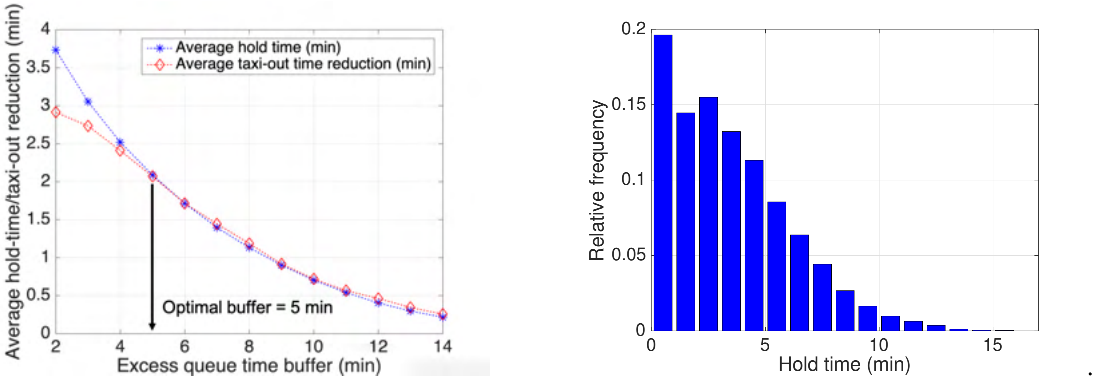
3.2.3 Departure metering benefits with accurate EOBTs

In this section, we assume that we have accurate EOBT information, that is, the estimate of the push-ready time published by the airlines is accurate. This is a reasonable assumption for scenarios in which the hold time is assigned to flights when the pilot calls ready [158, 179], including the current phase of the ATD-2 demonstration. However, in the long-term, the plan is to assign gate holds several minutes (say 10-30 min) prior to the flight's EOBT. At time T_p prior to a flight's EOBT, the taxi-out time for a flight is predicted from the analytical queuing model using the EOBT information for all the departures, landing times for future arrivals, weather condition and current traffic at the airport. For the purpose of evaluating the departure metering benefits with accurate EOBTs, we consider the flight's EOBT to be equal to its Actual Off-Block Time (AOBT) obtained from historical data. The gate release time (TOBT) for the flight is then determined using Eq. (3.1). The flights are released at TOBT in the discrete queuing simulation of the airport. In the field demonstrations at CLT, NASA uses a different model for taxi-out time prediction, a microscopic node-link model of the airport, instead of a queuing model [179].



(a) Average hold time (b) Average taxi-out time (c) Average wheels-off delay

Figure 3-2: Metering simulation results for different excess queue time buffer values. The plots show averaged values over a 15 min time window for a typical day at DFW in North-Flow configuration (07/20/2016).



(a) Average taxi-out reduction and hold time for different buffers (min).

(b) Hold time distribution for the flights that were held.

Figure 3-3: Departure metering statistics for DFW-SF. The statistics were calculated over 9,212 flights in a 15 day period.

Figure 3-2 shows the simulation results of departure metering with a planning horizon of 20 min for different values of the excess queue time buffer (Y) and the baseline case that corresponds to no metering, on a typical day at DFW. The reduction in taxi-out time is correlated with the hold-time, which in turn is larger during periods when the baseline taxi-out time is high. As the excess queue time buffer increases, the average hold-time decreases, leading to lower taxi-out time reduction benefits. On the other hand, a lower excess queue time buffer might lead to aircraft being held longer than necessary, leading to a delayed wheels-off time (Fig. 3-2(c)). The excess queue time buffer is therefore an important parameter that influences the taxi-out time reduction and airport throughput. In the field demonstrations at CLT conducted by NASA, ramp managers are allowed to pick the excess queue time buffer based on their experience [179]. We present a methodology for systematically selecting the excess queue time buffer parameter.

The wheels-off delay for a flight because of departure metering is given by,

$$\begin{aligned}
\text{wheels-off delay} &= (\text{wheels-off time})_{\text{metering}} - (\text{wheels-off time})_{\text{baseline}} && (3.2)\\
&= \Big((\text{taxi-out time})_{\text{metering}} + \text{hold-time} + \text{AOBT}_{\text{baseline}}\Big) \\
&\qquad\qquad - \Big((\text{taxi-out time})_{\text{baseline}} + \text{AOBT}_{\text{baseline}}\Big) && (3.3)\\
&= \text{hold-time} - \Big((\text{taxi-out time})_{\text{baseline}} - (\text{taxi-out time})_{\text{metering}}\Big) && (3.4)\\
&= \text{hold-time} - (\text{taxi-out time reduction}) && (3.5)
\end{aligned}$$

The difference between the hold-time and taxi-out time reduction is equal to the wheels-off delay. A positive value of this change indicates that the runway is underutilized. Figure 3-3(a) shows the average hold time and taxi-out time reduction for different excess queue time buffers in DFW-SF. The figure indicates the sensitivity of the average hold-time and taxi-out time to the buffer parameter. As the buffer value increases, the hold-time as well as reduction in the taxi-out time decreases. We can see that at small values of the excess queue time buffer, the wheels-off delay is positive, indicating that the runway is underutilized. The optimal excess queue time buffer is chosen to achieve as large a taxi-out time reduction as possible while not losing runway utilization. For the 20 min horizon considered here, the optimal excess queue time buffer is 5 min for DFW-SF, with an average taxi-out time reduction of 2.1 min.

Parameter	**EWR**		**DFW**		**CLT**	
	NF	**SF**	**NF**	**SF**	**NF**	**SF**
Optimal buffer (min)	7	8	5	5	7	5
Mean taxi-out time reduction(min)	2.0	1.7	2.0	2.0	2.7	1.9
Mean hold-time (min)	2.1	1.7	2.0	2.1	2.8	1.9
Mean wheels-off delay (min)	0.07	0.02	0.05	0.02	0.10	0.06
Percentage of flights held	61%	56%	58%	59%	64%	53%
Mean hold time of the flights held (min)	3.4	3.1	3.5	3.5	4.3	3.6

Table 3.2: Effects of departure metering with 20-min planning horizon without EOBT uncertainty.

The optimal buffer size and taxi-out time reduction vary by airport and runway configuration. Table 3.2 shows the optimal buffer size and taxi-out time reduction from the simulations of departure metering, assuming perfect EOBT information. As one would expect, the optimal buffer size is higher for airports with higher taxi-out time prediction errors. Although the taxi-out delays at EWR and CLT are larger than those at DFW, the reduction in taxi-out time is about the same due to differences in prediction errors. At all the airports, about 60% of flights are held at their gates, and the mean hold time of these flights is about 3 min. The distribution of the hold time for the flights that were held is shown in Fig. 3-3(b) for DFW-SF.

Gate conflicts

A gate conflict occurs when an arriving flight needs to wait for the gate occupied by another aircraft that is supposed to depart but which is currently being held. Gate conflicts could potentially decrease the benefits of departure metering, since an arrival may need to use a gate at which a departure is being held. Fig. 3-4 shows the cumulative distribution of the time between pushback and the next arrival time at the gate for operations at DFW. It shows that only 20% of the departing flights have an arriving aircraft that uses the same gate within 20 min of the departing flight's pushback time. Since the simulations indicate that the average hold time for the flights held at the gate is 3.5 min and maximum hold time is 14 min (corresponding to optimal buffer size), we do not expect a significant reduction in the taxi-out time savings due to gate conflicts. However, one needs to note that the impacts of gate conflicts could be different at other airports depending on various factors such as the level of congestion (that decides the hold time for departures) and the current level of gate utilization.

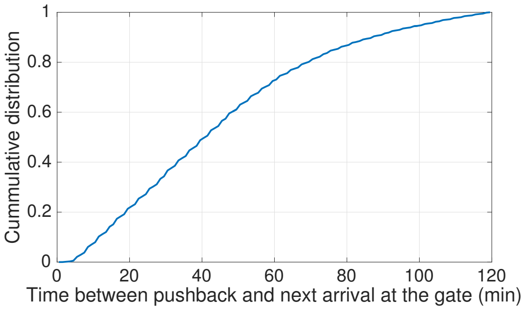


Figure 3-4: Cumulative distribution of the time between the actual pushback time and the next arrival at a gate.

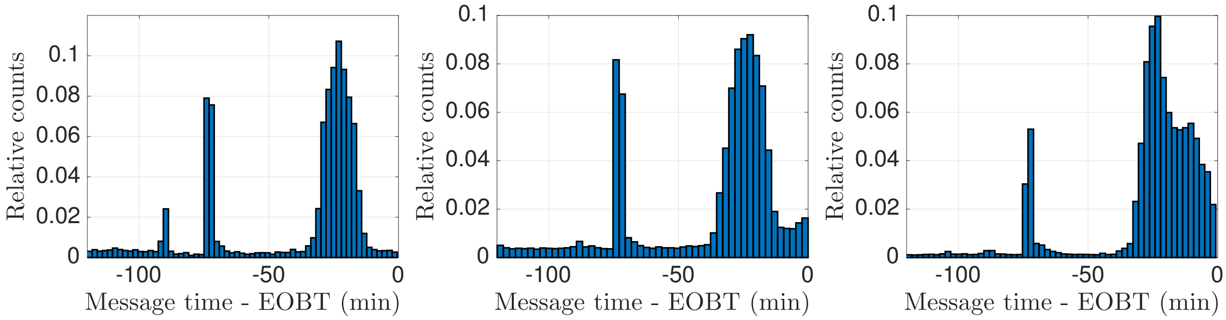
3.3 Impact of EOBT uncertainty

The results for the taxi-out time prediction and departure metering presented so far assumed that the EOBT information provided by the airlines was accurate. However, the EOBTs published by airlines often deviate from the actual push-ready times, motivating the study of the impact of EOBT errors on departure metering.

3.3.1 Evaluating EOBT uncertainty

EOBT information was obtained from the FAA's Traffic Flow Management System (TFMS) for EWR, DFW, and CLT. Based on the data availability, we considered the following periods for our analysis: Jan 01 - Nov 11, 2018 (EWR), May 01 - July 31, 2018 (DFW) and Dec 13, 2017 - Feb 12, 2018 (CLT). Flights departing between 9AM and 12PM (local time) were removed from CLT's dataset to eliminate any effects of the ATD-2 departure metering demonstration.

Figure 3-5 shows the distribution of the time when the EOBT information was published for a flight relative to its EOBT for a major airline (*Airline A*) at the three airports. Most



(a) EWR (b) DFW (c) CLT

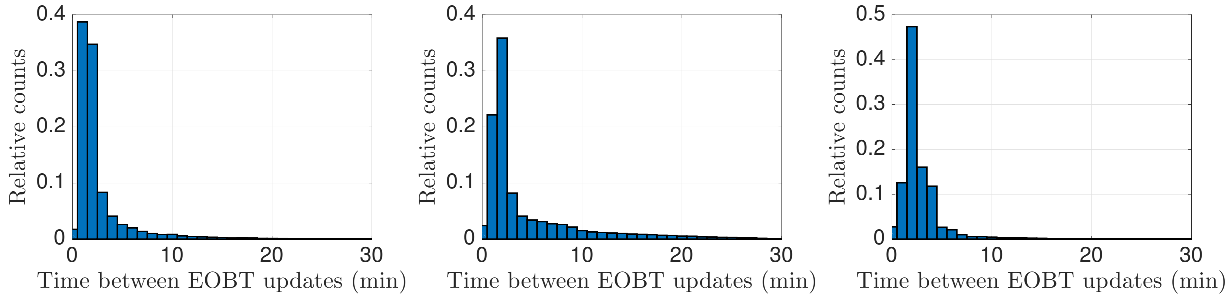
Figure 3-5: Message time of the EOBT information relative to the EOBT time (Airline A).

of the EOBT messages appear to be published within 40 min prior to the flight's EOBT (median: 20-23 min), with an additional spike at around 73 min. Further analysis indicates that airlines publish the Initial Gate Time of Departure (IGTD) or the original scheduled gate pushback time prior to the 40 min mark (including the spike at 73 min mark seen in Fig. 3-5) [20]. Table 3.3 shows the EOBT update statistics for the three airports.

Parameter	**Mean**			**Median**			σ		
	EWR	**DFW**	**CLT**	**EWR**	**DFW**	**CLT**	**EWR**	**DFW**	**CLT**
Message time − EOBT (min)	-22.7	-22.2	-18.6	-23.0	-23	-20.1	5.5	5.5	8.3
Time between updates (min)	2.7	4.4	2.8	1.6	1.8	2.0	3.3	5.3	2.6
No. of EOBT messages per flight	10.3	9.8	13.3	9	8	10	5.3	5.8	10.2

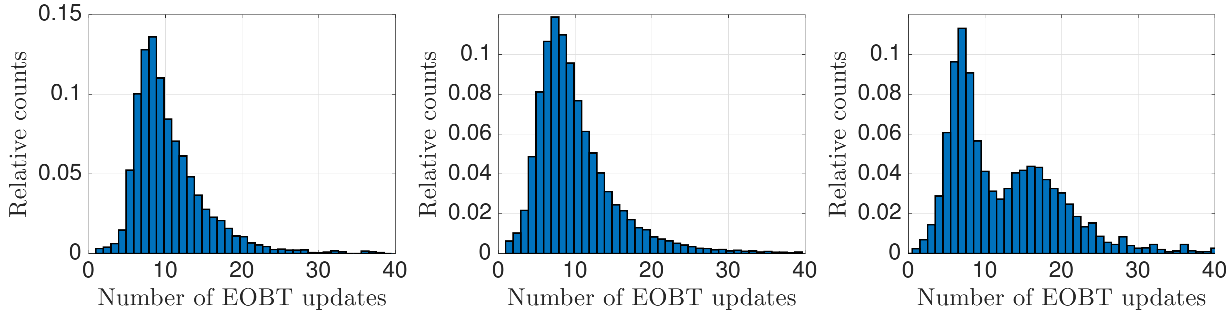
Table 3.3: EOBT update statistics computed for lookahead horizons $\leq$ 40 min (Airline A).

The EOBT information is also dynamically updated by airlines at irregular intervals until the flight pushes back from the gate [63]. The EOBT updates are a concern for the decision makers who need a 'stable' information for making decisions. Figure 3-6 shows the distribution of the time between EOBT updates for a flight, considering messages within 40 min of the flight's EOBT. Only 28% of the flights operating out of EWR had their EOBT published (during the period considered in our analysis, Jan 01 - Nov 11, 2018). However, all flights with a planned departure are expected to publish EOBT information in the near future [63]. Figure 3-7 shows the distribution of the number of EOBT updates for a flight,



(a) EWR (b) DFW (c) CLT

Figure 3-6: Time between EOBT updates for lookahead time less than 40 min (Airline A).

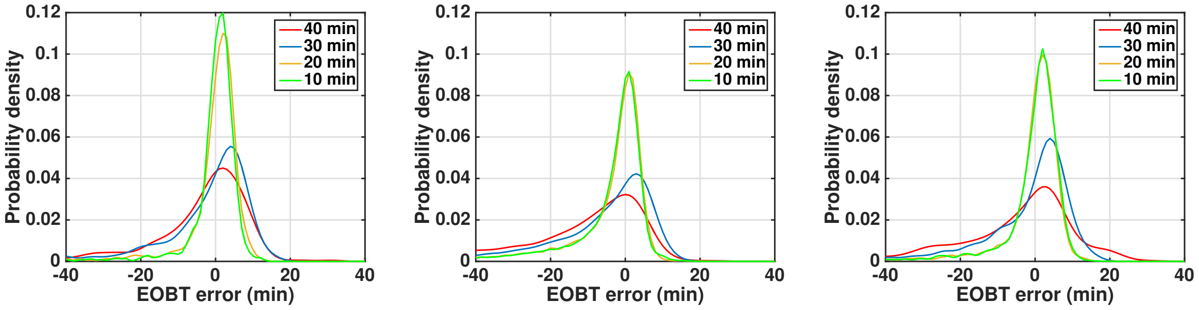


(a) EWR (b) DFW (c) CLT

Figure 3-7: Distribution of number of EOBT updates recieved by a flight which had its EOBT published (Airline A).

given that EOBT information for it was published at least once. The median number of updates is between 8 and 10 depending on the airport.

The EOBT error was calculated by subtracting the ASPM-derived Actual Off Block Time (AOBT) for that flight from the published EOBT [68]. Note that AOBT is only available post-event, after the flight has pushed back from the gate. The EOBT errors errors were compared over different lookahead time horizons (the difference between the EOBT message transmission time and the EOBT value published at that time). Figure 3-8 shows the results of fitting a Probability Density Function (PDF) to the EOBT error distributions for different lookahead times, obtained for Airline A at EWR, DFW and CLT. The key error statistics are presented in Table 3.4. As expected, the standard deviations of the distributions increase with increasing lookahead time. The EOBT error distributions also vary by airport. It is worth noting that the standard deviation of the EOBT error, even for lookahead times of



(a) EWR (b) DFW (c) CLT

Figure 3-8: EOBT error distribution for different lookahead horizons (Airline A).

10-20 min, is of the same order of magnitude as the taxi-out time prediction errors from the queuing model. It is therefore important to consider the uncertainty in EOBT information while evaluating the benefits of departure metering. The EOBT error distributions for all lookahead horizons are negatively skewed. The distributions for a 20-min horizon show that flights generally do not pushback more than 10 min prior to their EOBT; however, there are flights that get delayed beyond 20 mins of the published EOBT.

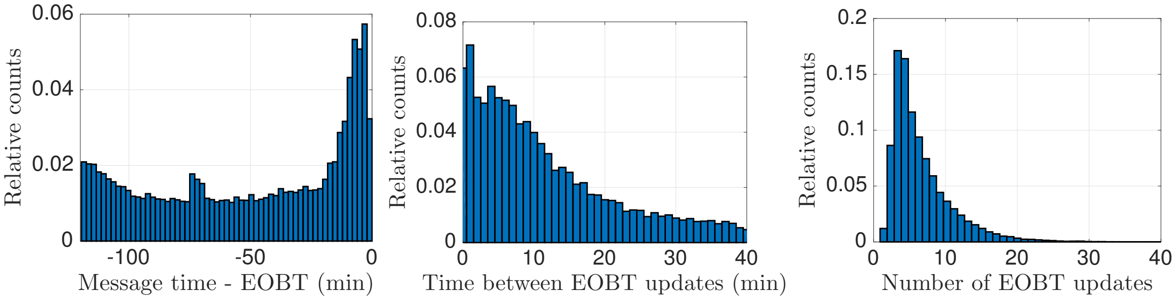
The EOBT error distributions vary by airline, since each airline does its EOBT computation internally using different data sources and algorithms. Table 3.5 shows the EOBT error statistics for two other airlines at EWR (*Airline B* and *Airline C*). The errors are significantly higher for these two airlines compared to Airline A. Fig. 3-9 presents the EOBT update distributions for Airline B at EWR, and shows that Airline B tends to update its EOBT information less frequently than Airline A.

Lookahead time (min)	**Mean error (min)**			**Median error (min)**			**σ error (min)**		
	EWR	**DFW**	**CLT**	**EWR**	**DFW**	**CLT**	**EWR**	**DFW**	**CLT**
10	1.0	-1.8	1.5	1.1	-0.4	1.8	3.1	5.9	3.9
20	1.5	-1.8	1.4	1.8	-0.2	1.7	3.5	6.1	3.9
30	0.0	-6.2	-0.1	2.0	-2.0	2.2	8.2	13.7	8.9
40	-1.5	-12.9	-4.5	0.0	-7.0	0.0	8.9	18.4	14.7

Table 3.4: EOBT error statistics for different lookahead times (Airline A).

Lookahead time (min)	Mean error (min)		Median error (min)		σ error (min)	
	Airline B	**Airline C**	**Airline B**	**Airline C**	**Airline B**	**Airline C**
10	-3.9	-3.9	-1.0	0.2	11.4	9.8
20	-7.7	-5.1	-5.0	0.3	14.3	10.9
30	-7.9	-7.2	-5.0	0.5	14.8	9.8
40	-9.1	-7.5	-4.0	0.7	17.0	13.2

Table 3.5: EOBT error statistics for Airline B and Airline C at EWR computed for different lookahead horizons.



(a) Message time (b) Time between EOBT updates (c) Number of EOBT updates

Figure 3-9: EOBT information update statistics for Airline B at EWR.

3.3.2 Parametric analysis of the impacts of EOBT uncertainty

The uncertainty in EOBT information impacts departure metering for two reasons:

- *Reduced accuracy of taxi-out time predictions*: Departure metering algorithms require an estimate of taxi-out times to determine gate-hold times. The taxi-out time predictions need the EOBTs as inputs (for example, to a queuing model). Errors in EOBT result in taxi-out time prediction errors, leading to inefficient gate-hold times.

- *Noncompliance with assigned gate release times*: The pushback time recommendation (TOBT) is given by the EOBT time plus the gate-hold time. However, a flight may not be ready to push at its TOBT due to errors in its EOBT, resulting in a loss in runway throughput.

We conduct a parametric study of the impacts of inaccurate EOBTs on departure metering using the ATD-2 logic. We assume that the EOBT error distributions are Gaussian with a zero mean, and we vary the standard deviations. This methodology allows us to investigate the impact of different error distributions in a systematic way. The range of the parametric space for the standard deviation of the error distribution follows from the operational data. The EOBT time is synthetically computed for every flight in the data-set by adding a random variable sampled from the EOBT error distribution to the AOBT (in other words, EOBT = error + AOBT).

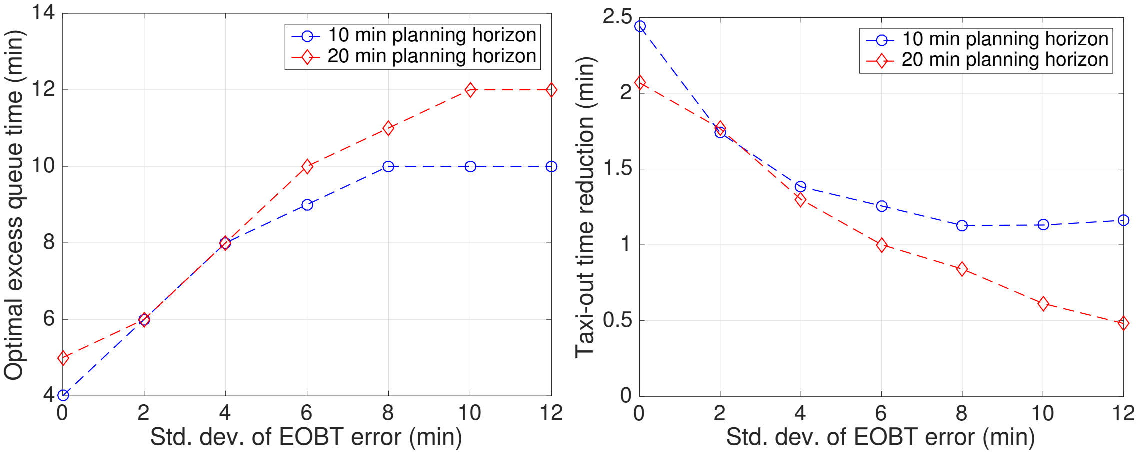
For a flight with an uncertain EOBT, we can have one of the following three cases:

1. *Push-ready time* $<$ *EOBT*: In this case, the metering logic would recommend a gate-hold until the EOBT time (i.e. TOBT = EOBT) even if the predicted excess queue time (predicted wait time) is equal to zero. While this strategy may improve predictability, flights would have to hold unnecessarily at the gate, resulting in a loss in runway throughput. To prevent unnecessary holds, the metering logic is modified to let flights pushback at the push-ready time if the predicted excess queue time (along with the buffer) is equal to zero.

2. *EOBT* $\leq$ *push-ready time* $\leq$ *TOBT*: In this case, the flight waits and then pushback occurs at its TOBT. The gate-hold time would be expected to translate into a reduction in taxi-out time.

3. *TOBT* $<$ *push-ready time*: In this case, the flight is not ready to pushback at its TOBT and the pushback occurs later. Once again, there is a loss of predictability and a potential loss in runway throughput because the flight did not pushback when expected.

The optimal buffer size in the ATD-2 logic can be adjusted to account for the EOBT uncertainties, in addition to the taxi-out time prediction errors. The optimal excess queue time buffer is the smallest value of the queue buffer for which there is no increase in the wheels-off time. Figure 3-10(a) shows the optimal excess queue time buffer for different EOBT distributions and two different planning horizons (10-min and 20-min) for DFW-SF. For a particular planning horizon, the optimal excess queue time buffer increases with an increase in EOBT error and then saturates beyond a particular point. This behavior is



(a) Optimal excess queue time buffer (b) Taxi-out time reduction

Figure 3-10: Optimal excess queue time buffer and taxi-out time reduction for DFW-SF.

as expected, since the taxi-out time prediction error increases with EOBT uncertainty and needs to be compensated with a larger excess queue time buffer. For higher EOBT errors, a larger buffer size is required at the 20-min planning horizon than the 10-min horizon, due to the expected decrease in prediction performance.

There is an inherent trade-off in choosing the planning horizon: A longer planning horizon is preferable since it leads to more predictability, but it yields lower departure metering benefits. One would ideally identify the optimal value of the planning horizon by evaluating the tradeoff between predictability and fuel savings; however, it is difficult to do so since the value of predictability is not known.

We can also evaluate the departure metering benefits for different EOBT error distributions using the optimal excess queue time buffer. The taxi-out time savings decreases as the standard deviation of the EOBT error increases (Figure 3-10(b)). At a 20-min planning horizon, with the current level of EOBT uncertainty at DFW (standard deviation of 6 min), there is a 50% decrease in taxi-out time reduction benefits compared to the case with perfect EOBT information.

Table 3.6 shows the average taxi-out time reduction and optimal buffer for different levels of EOBT uncertainty at different airports and a 20-min planning horizon. The trends in the reduction of metering benefits with EOBT uncertainty are consistent across all the airports.

EOBT error σ (min)	**EWR**		**DFW**		**CLT**	
	NF	**SF**	**NF**	**SF**	**NF**	**SF**
0	2.0 (5)	1.7 (8)	2.0 (5)	2.1 (5)	2.7 (7)	1.9 (5)
2	1.6 (6)	1.3 (8)	1.7 (6)	1.8 (6)	2.3 (8)	1.6 (5)
4	1.2 (8)	1.2 (9)	1.2 (8)	1.3 (8)	1.8 (10)	1.4 (6)
6	0.9 (10)	1.2 (10)	0.9 (10)	1.0 (10)	1.3 (12)	1.3 (7)
8	0.7 (11)	0.9 (12)	0.7 (11)	0.8 (11)	1.1 (13)	1.1 (8)
10	0.5 (12)	0.9 (13)	0.5 (12)	0.6 (12)	1.0 (13)	1.1 (9)
12	0.5 (12)	0.9 (13)	0.5 (12)	0.5 (12)	0.9 (13)	0.9 (10)

Table 3.6: Average taxi-out time reduction in minutes for different EOBT error distributions with a planning horizon of 20-min. The optimal buffer for the ATD-2 logic is indicated in brackets.

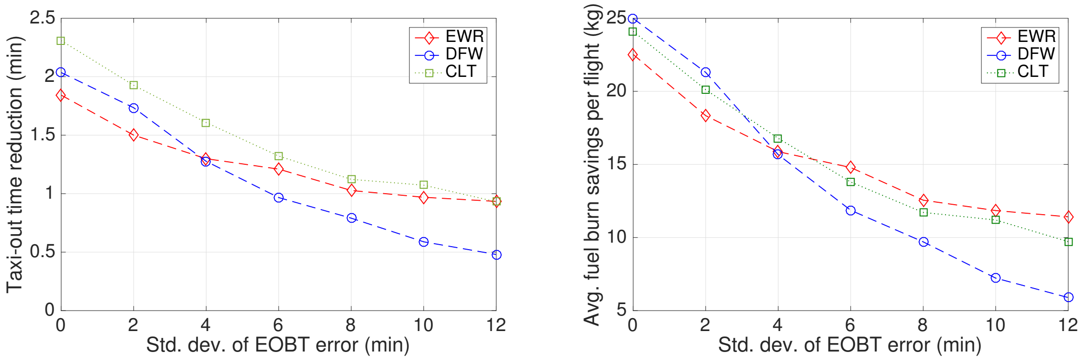


Figure 3-11: Annualized taxi-out time reduction and fuel burn reduction per flight using the ATD-2 logic for different levels of EOBT uncertainty.

Figure 3-11 shows the annualized average taxi-out time and fuel burn reductions as a function of the EOBT error. These calculations account for the relative frequency of different runway configurations and the average fleet mixes at the different airports. The values of the average taxi fuel flow rate per flight were found to be 0.20 kg/s for EWR and DFW, and 0.17 kg/s at CLT (due to the higher fraction of regional jets) [88]. Table 3.7 shows the estimated annualized total fuel burn saving at the three airports obtained by multiplying the average fuel burn savings per flight with the total number of operations in 2017.

Airport	**EOBT error σ (min)**						
	0	2	4	6	8	10	12
EWR	4.9	4	3.4	3.2	2.7	2.6	2.5
DFW	8.4	7.2	5.3	4	3.3	2.4	2
CLT	6.6	5.5	4.6	3.8	3.2	3.1	2.7

Table 3.7: Annualized fuel burn savings ($\times 10^6$ kg) using the ATD-2 departure metering logic.

3.3.3 An optimal control approach for departure metering

We propose an approach based on optimal control technique to departure metering in the presence of EOBT uncertainty, by leveraging the analytical queuing models developed in Section 2.2. The objective of the formulation is to regulate the rate of aircraft pushbacks so as to achieve smaller queue lengths on the airport surface, while maintaining runway throughput.

Let $\mathbf{x}(t) \in \mathbb{R}^p$ be a vector of taxi-out queue lengths on the airport surface at any time instant t. Let $\mathbf{d}(t) \in \mathbb{R}^q$ be the departure demand rate, with the elements representing the demand rate to each runway. The departure demand corresponds to the push-ready time for departures, and is obtained from the EOBT information provided by the airlines. These rates are obtained over 5-min windows. Let $\mathbf{u_d}(t)$ be the pushback rate at the gate that is assigned by the controllers, and $\mathbf{h}(t)$ be the number of aircraft held at the gate due to departure metering. The number of holds at any instant, $\mathbf{h}(t)$, is the difference between the total demand and total pushbacks until time t. The dynamics for the number of holds is then given by:

$$\mathbf{h}(t) = \int_0^t \mathbf{d}(x)dx - \int_0^t \mathbf{u_d}(x)dx \tag{3.6}$$

$$\dot{\mathbf{h}}(t) = \mathbf{d}(t) - \mathbf{u_d}(t) \tag{3.7}$$

The control objective is to minimize the length of taxi-out queues while maintaining throughput, the state variables are the taxi-out queue lengths on the airport surface and the number of holds, and the control variable is the departure pushback rate ($\mathbf{u_d}(t)$). The problem formulation for an airport queuing network is then as follows:

$$\min_{\mathbf{u_d}(t)} \int_0^T \Big(\mathbf{x}^{\mathrm{T}} Q \mathbf{x} + \mathbf{h}^{\mathrm{T}} R \mathbf{h}\Big) dt \tag{3.8}$$

Subject to: (3.9)

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{x}(t-\tau_1), ...\mathbf{x}(t-\tau_m), \mathbf{u}_d(t-\tau_{m+1}), ...\mathbf{u}_d(t-\tau_w), t) \tag{3.10}$$

$$\dot{\mathbf{h}} = \mathbf{d}(t) - \mathbf{u_d}(t) \tag{3.11}$$

$$0 \le x_i, h_i; \;\; 0 \le u_{di} \le u_m; \; i = 1,2,3,...w \tag{3.12}$$

$$u_{di}(t) = g_i(t), \; t \in [-\tau_{di}, 0); \; i = 1,2,3,...w \tag{3.13}$$

$$x_i(t) = \phi_i(t), \;\; t \in [-\tau_{ki}, 0], \; \mathbf{h}(0) = \mathbf{h}_0; \; i = 1,2,3,...w \tag{3.14}$$

Here, T is the optimization window over which the cost needs to minimized, and $Q \in \mathbb{R}^{p\times p}$ and $R \in \mathbb{R}^{q\times q}$ are constant weighting matrices. Equations (3.10) and (3.11) specify the dynamics for the length of the queues and number of holds. The inequalities in (3.12) impose nonnegativity constraints on the number of holds, queue lengths and pushback rate. The delay differential equations also require an initial history, specified by Eqs. (3.13)-(3.14). The cost function (3.8) penalizes a weighted sum of the square of the queue lengths and number of holds. The queue length is penalized to reduce taxi-out times, while the holds are penalized to avoid having large holds and to maintain runway throughput.

In Section 2.2, we saw that the dynamics of surface queues depend on the service time of several servers. The service time distributions for the departure runway queues depend on the landing rates and meteorological conditions, which are assumed to be known. The service time distribution of the taxi-out ramp queue requires the number of taxi-in flights on the ramp (in the case of CLT-NF). The taxi-in ramp queue length is pre-computed using the queuing model with the EOBTs and arrival times, and is used to determine the service rate for the taxi-out ramp queue. This service rate is used for the taxi-out ramp queue dynamics in the optimal control problem. In this manner, the interdependencies between departures and arrivals in the ramp are simplified so that only the arrivals impact the departures. This methodology reduces the number of constraints in the optimal control problem. To speed up the computation further, the optimal pushback rate is computed only when the runway queue length at T_p min in the future is predicted to be greater than 2.

The optimal control problem is solved in a receding horizon framework, accounting for the current state of the airport surface. The entire day is divided into 5-min intervals. At the beginning of each interval, pushback rate decisions are made for $[T_p, T_p + 5]$ min, where T_p is the planning horizon. The optimal control problem is solved for $[T_p, T_p + T]$ where T is the optimization window over which the cost is minimized, and the pushback rate is implemented for the first 5 min. The initial conditions for the optimal control problem at T_p min into the future are obtained using the queuing network model with the current state as the input. The number of aircraft that can be released during each 5 min window (n) is determined from the pushback rate. The first n aircraft in the 5 min window are released as per the optimal control decision and rest of the aircraft are pushed to the beginning of the next time window, awaiting decision for release. In contrast to the ATD-2 logic which assigns and freezes the hold time for flights that have an EOBT T_p min ahead, the optimal control approach only specifies the flights that need to be released in $[T_p, T_p + 5]$ time window, and pushes the remaining flights to the next time window. Consequently, aircraft can be pushed multiple times to the next time window but the release decision is made T_p minutes ahead.

The optimal control problem is solved numerically rather than analytically due to the challenges posed by time delays and nonlinear dynamics. The optimal control problem is solved by discretizing the state and control variables. The equations governing the dynamics are discretized using a first-order Euler method. The discretized control problem is then transformed into a non-linear programming problem (NLP) that can be solved using a standard solver in MATLAB.

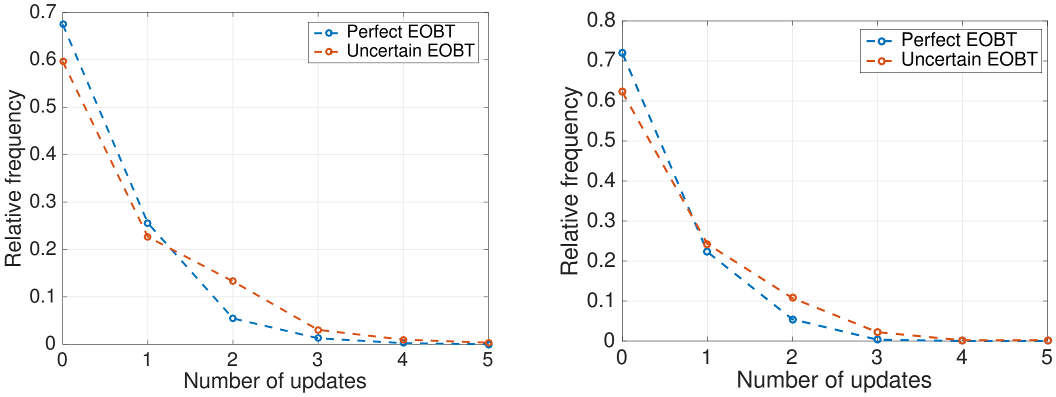
We demonstrate the benefits of the optimal control approach by simulating 3 days of operations in DFW-SF and CLT-NF. The queuing models of these two airport-configuration pairs differ significantly in complexity. The DFW dataset consisted of 2,708 departures and CLT dataset consisted of 1,817 departures. We penalize the departure runway queue in the cost function with a weight equal to one, and impose a weight of 0.4 for the holding term ($Q = \mathbb{I}_p, R = 0.4\mathbb{I}_q$). The weight of the holding term was determined in a similar way as we had determined the optimal queue buffer for ATD-2 logic, to ensure that the mean wheels off delay is close to zero. We consider T, the optimization window over which the cost is minimized to be 30-min for CLT and 60-min for DFW. The longer horizon yielded better departure metering benefits at DFW, because of the longer departure banks.

Table 3.8 compares the benefits of the optimal control approach and the ATD-2 logic. While the ATD-2 logic yields more benefit with perfect EOBT information, the optimal control approach performs better in the presence of uncertainty. The benefits in terms of taxi-out time reduction with the optimal control approach in the presence of EOBT uncertainty are about 66% higher for DFW-SF and 33% higher for CLT-NF. Figure 3-12 shows the relative counts of the number of times a flight is pushed to the next 5 min interval, given that it was asked to hold at least once. Most of the flights that are held at the gate receive at most two updates.

Parameter	**DFW-SF**				**CLT-NF**			
	Perfect EOBT		Uncert. EOBT		Perfect EOBT		Uncert. EOBT	
	Opt.	ATD-2	Opt.	ATD-2	Opt.	ATD-2	Opt.	ATD-2
Mean taxi-out time reduction(min)	1.4	1.9	1.0	0.6	1.3	2.6	1.2	0.9
Mean hold-time (min)	1.5	1.9	1.1	0.7	1.5	2.8	1.3	1.0
Mean wheels-off delay (min)	0.1	0.1	0.1	0.1	0.2	0.1	0.1	0.0
Fraction of flights held	0.3	0.6	0.2	0.1	0.3	0.6	0.2	0.1
Mean hold time of flights held (min)	4.9	3.5	5.3	6.3	4.5	4.3	6.2	6.8

Table 3.8: Comparison of the departure metering benefits with and without EOBT uncertainty (assuming $\sigma = 6$ min). Opt. refers to results from the optimal controller.

Table 3.9 compares the computational times of the optimal control approach and the ATD-2 logic, based on the simulation of a single day of airport operations. The computations were done on a computer with 2.8 GHz Intel Core i7 processor with 16 GB memory, and the scripts were run on MATLAB. The computational time of the ATD-2 logic corresponds to deciding the gate-holds for one or more flights that have the same EOBT. The computational time can vary slightly depending on the number of flights that share a particular EOBT. The computational time indicated for the optimal control approach is the time required to make a decision for every 5-min window in the receding horizon framework. The time duration to solve the optimal control problem can vary based on the instance of the optimization problem. In general, the ATD-2 logic is orders of magnitude faster than the optimal control



(a) DFW (SF)

(b) CLT (NF)

Figure 3-12: Relative frequency of the number of updates for the release time for the flights that were held at the gate using the optimal control policy.

approach in terms of computational time. However, the optimal control approach is still practical: The mean time for planning a 5-min window is around 2 sec, and maximum computation time is around 40 sec. Finally, the computational time for CLT is higher than that for DFW, because of the more complex queuing network.

Airport	**Mean comp. time (sec)**		**Max. comp. time (sec)**	
	ATD-2	**Opt. Ctrl.**	**ATD-2**	**Opt. Ctrl.**
DFW-SF	0.5×10^{-3}	1	17×10^{-3}	31
CLT-NF	0.9×10^{-3}	3	24×10^{-3}	39

Table 3.9: Computational times for the ATD-2 logic and the optimal control approach.

3.4 Discussions

3.4.1 Fairness of departure metering

The parameters of the departure metering algorithms were tuned to ensure that there was no loss in throughput, that is, no net delay for the departures from the airport. However, depending on the hold-time assigned to the flights, the take-off order in the presence of metering might be different compared to the baseline. This could result in one flight experiencing less delay at the expense of more delay for another flight.

Table 3.10 shows the mean absolute displacement in the takeoff order with metering, which is an indicator of unfairness. Here, "displacement" refers to the rank difference of the takeoff order at each runway with departure metering, compared to the baseline case without departure metering. The buffer parameter ensures that there is no loss in throughput, and that the average takeoff delay is zero. Ideally, in a fair allocation, the mean absolute displacement in the takeoff order should be equal to zero. However, a higher value of mean absolute displacement indicates that some flights take off earlier in the metering case compared to the baseline, while some others take off later. Ideally, fairness suggests that the takeoff order should be the same in the metering case as in the baseline. Table 3.10 shows that the mean absolute displacement is higher in the presence of EOBT uncertainty.

Category	**DFW-SF**				**CLT-NF**			
	Perf. EOBT		Uncert. EOBT		Perf. EOBT		Uncert. EOBT	
	Opt.	ATD-2	Opt.	ATD-2	Opt.	ATD-2	Opt.	ATD-2
Mean absolute displacement of flights that are not held	0.2	0.0	0.5	0.6	0.4	0.2	0.6	0.8
Mean absolute displacement of flights that are held	1.5	1.2	1.9	4.0	1.6	1.7	2.0	3.9
Mean absolute displacement of all flights	0.6	0.7	0.8	1.0	0.8	1.2	0.9	1.3

Table 3.10: Mean absolute displacement of the takeoff order with and without EOBT uncertainty (assuming $\sigma = 6$ min). Opt. refers to results from the optimal control approach.

3.4.2 Incentives to improve EOBT accuracy

The analysis presented in this chapter quantifies the benefits of more accurate EOBTs in terms of taxi-out time savings. The standard deviation of the EOBT error distribution is around 4 min at a 20-min time horizon for EWR and CLT, while it is 6 min at DFW (Table 3.4). If the airlines were to improve the EOBT accuracy by reducing the standard deviation of the EOBT error to about 2 min, the benefits in terms of taxi-out time savings increases by 15% at EWR, 75% at DFW, and 22% at CLT. This improvement in EOBT accuracy yields an additional fuel burn savings of 600K kg/yr at EWR, 3.4M kg/yr at DFW, and 1.1M kg/yr at CLT. Assuming the jet fuel price to be \$0.68/kg [87], the additional annual fuel savings will correspond to \$400K at EWR, \$2.3M at DFW, and \$740K at CLT. Further, one can also expect benefits in terms of cost of time to airlines and passengers. The airlines can use these estimates to decide on investment decisions that could improve their EOBT accuracy.

3.4.3 Other considerations for departure metering

Ramp operations (including clearance for gate pushbacks) in the US are primarily managed by the airlines and not by the FAA. While spot metering has been proposed as a solution to this 'jurisdictional' challenge, its practicality remains to be demonstrated. In the departure metering field tests at CLT under the ATD-2 program, gate-hold decisions were made by the ramp controllers (airline personnel) by coordinating with the single airline that handles more that 90% of the operations. The methodology presented in this thesis can be adapted to consider spot metering, assuming that the implementation challenges can be addressed.

Our work focused on uncertainties pertaining to EOBTs and taxi times. However, there are multiple sources of uncertainty on the airport surface; for example, departures being subject to Traffic Management Initiatives (TMIs) such as Approval Requests (APREQs) or Miles-In-Trail (MIT) restrictions. However, only a small fraction of flights (fewer than 9% of CLT departures) are affected by such TMIs. We therefore believe that we have considered the uncertainties that have the most significant impact on departure metering.

Finally, an important aspect that needs to be considered for real-world deployment of the departure metering algorithms is operational validation. The airport traffic models and the parameters associated with the control algorithms were obtained using historical operational data. However, the model parameters might need refinement during operational use because of changes in airport operations (such as changes in fleet mix, infrastructure, etc.). Therefore, one needs to actively monitor the performance of these models, and make refinements when needed.

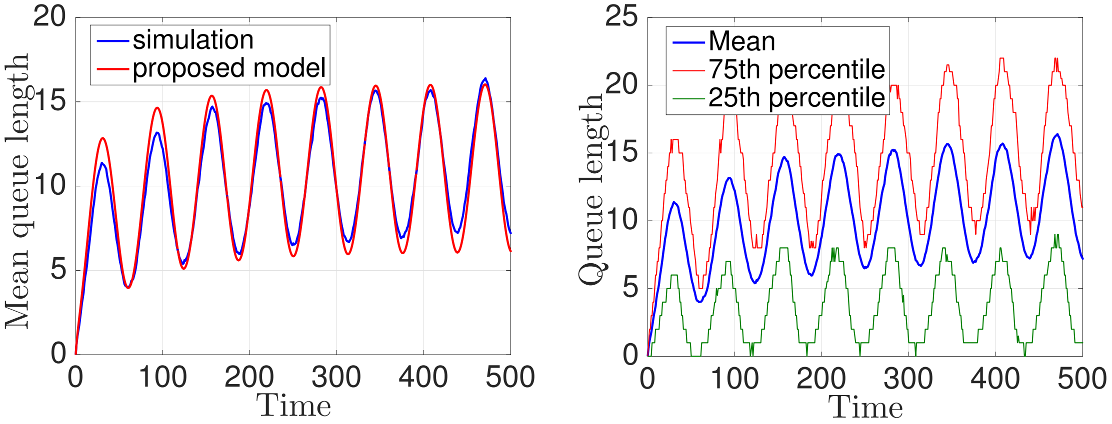
Chapter 4

Robust control for queuing networks: Applications to airport operations

4.1 Introduction

In the previous chapter, we presented a heuristic (ATD-2 logic) and an optimal control strategy to control the arrival rate (or flight pushbacks in the case discussed) into a queuing network to reduce congestion. The heuristic and the optimal control approach relied on the predictions of the queuing model to determine the pushback rate. However, the model predictions can be inaccurate, leading to inefficiencies. In this chapter, we present a robust control strategy that explicitly accounts for the model uncertainties to determine a policy to reduce congestion in a queuing network with time delays that account for the travel times between the servers in the network.

More generally, we propose a feedback controller to track the desired queue lengths by controlling the sending rate at the source. The proposed approach accommodates queuing networks with propagation time-delays between nodes, and arbitrary service time distributions. This is achieved through a sliding mode control approach, with a predictor-based feedback to compensate for the time-delays. Prior work on controlling queues using fluid models with techniques from nonlinear control theory have not accounted for a combination of time-delays, and multiple servers [32, 144]. Time-delays can be destabilizing; traditional techniques for analyzing the stability of time-delay systems (e.g., Lyapunov-Krasovskii or Lyapunov-Razumikhin methods) are challenging and often result in very conservative results for nonlinear systems [77, 73]. Using Lyapunov analysis, we determine sufficient conditions for tracking the queue length and bounds on the tracking error for the case of a single queue. We also illustrate the performance of the controller for tandem queues using simulations. In the context of departure metering, this corresponds to determining the release times of aircraft from their gates in order to maintain a limited queue at the runway, resulting in reduced taxi-out delays, without under-utilizing the runway.



(a) Comparison between simulation and proposed model.

(b) Simulation results.

Figure 4-1: Mean queue length for $M_t/M/1$ queue ($\lambda/\mu = 0.9 + 0.45\sin(0.1t);\ \mu = 2$).

It is worth noting that our queuing model governs the evolution of the ensemble mean queue length and not the actual queue length. Since the arrival times and service times are stochastic in nature, the actual queue length is a random variable. Probabilistic queuing models, such as a Markov chain representation of the queuing process, provide the queue length probabilities at any time using the Chapman-Kolmogrov equation. However, such models are often complex and restrictive, and are difficult to scale to controlling queue lengths in large queuing networks. Fig. 4-1(b) shows the ensemble mean queue length for a $M_t/M/1$ system obtained from the simulations, along with the 75th percentile and 25th percentile of the queue length at any time instant. The figure shows that the actual queue length at any time can be considered to be a small deviation from the mean queue length. Consequently, we can use a model of the mean queue length to control the actual queue length, by considering uncertainties in the model prediction. Our proposed model for the mean queue length, Eq. (2.8), allows us to develop robust controllers to account for model uncertainties. We first present the methodology for simple queuing networks and later apply the approach for departure metering at airports.

4.2 Control for a single queue

In this section, we consider the case of a single queue served by a single server, and then present the analysis for more complex queuing networks in Sections 4.3 and 4.4, which will be used to develop an algorithm for departure metering. The objective is to maintain a desired queue length by controlling the release times into the queue. We first consider the case in which there is no travel time from the source (where the customers are released) to the point of entry into the queue. We will later consider the scenario with propagation delays.

4.2.1 Case without propagation delays

The fluid-flow model for a single queue served by a single server is given by the following equation:

$$\dot{x} = \bar{\alpha}(x,t) + u(t), \tag{4.1}$$

where $\bar{\alpha}(x,t) = -\mu(t)C(t)x/(C(t)x+1)$, μ is the mean service rate of the server, $u(t)$ is the mean arrival rate into the queue and the parameter C primarily depends on the coefficient of variation of the service time distribution. We assume that the actual dynamics deviates from the model, but has a similar structure of the form:

$$\dot{x} = \alpha(x,t) + u(t), \tag{4.2}$$

where $\alpha(x,t)$ is an unknown function that is bounded as follows:

$$|\alpha(x,t) - \bar{\alpha}(x,t)| \leq F(x,t). \tag{4.3}$$

Motivated by the fact that the errors arise primarily due to uncertainties in the individual service times, we consider,

$$F(x,t) = a(C(t)x)/(C(t)x+1). \tag{4.4}$$

Here, a is a design parameter that needs to be chosen depending on the level of uncertainty. The objective of the control problem is to determine a sending rate $(u(t))$ in order to maintain the queue length at a desired value, $x_d(t) \geq 0$. Next, we present a few standard definitions and theorems on stability properties that are used to develop the feedback controller [162, 152].

Definition 1 (Equilibrium point) *The state x^* is said to be an equilibrium point of the system $\dot{x} = f(x,t)$ if: $x(t_0) = x^* \implies f(x^*,t) = 0$ for all $t \geq t_0$. Without loss of generality, one can always consider the origin to be the equilibrium point through a simple coordinate transformation.*

Definition 2 (Stability) *The equilibrium point $x = 0$ is said to be stable at initial time t_0 if, for any $R > 0$, there exists $r(R, t_0) > 0$, such that if $\|x(t_0)\| < r$, then $\|x(t)\| < R$ for all $t \geq t_0$. Otherwise, the equilibrium point is said to be unstable. Additionally, the equilibrium point is said to be uniformly stable if the value of r in the preceding definition can be chosen independent of t_0.*

Definition 3 (Asymptotic stability) *The equilibrium point $x = 0$ is asymptotically stable at initial time t_0, if it is stable and, in addition, there exists some $r(t_0) > 0$ such that if $\|x(t_0)\| < r$ then $\|x(t)\| \to 0$ as $t \to \infty$. Additionally, the system is globally uniformly asymptotically stable if these properties hold true for any choice of r.*

Definition 4 (Class K, KR functions) *A function $\phi(x)$ belongs to class K if it is continuous, strictly increasing, and $\phi(0) = 0$. Additionally, $\phi(x)$ belongs to class KR if $\phi(x)$ belongs to class K and $\phi(x) \to \infty$ as $x \to \infty$.*

Definition 5 (Positive definite functions) *A continuous function $V(x,t)$ is said to be a positive definite function if for some $\phi(.)$ of class KR, $V(0,t) = 0$ and $V(x,t) \geq \phi(|x|)$ for all $t \geq 0$. Additionally, $V(x,t)$ is called a negative definite function if $-V(x,t)$ is positive definite.*

Definition 6 (Decrescent functions) *A continuous function $V(x,t)$ is said to be decrescent if there exists a function $\gamma(.)$ of class K, such that, $V(x,t) \leq \gamma(|x|)$ for all $t \geq 0$.*

Theorem 1 (Lyapunov theorem for global asymptotic stability [152]) *Assume there exists a scalar function $V(x,t)$ with continuous partial derivatives such that: (a) $V(x,t)$ is positive definite and decrescent, and (b) $\dot{V}(x,t)$, which is given by $\Big(\frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(x,t)\Big)$, is negative definite. Then, the equilibrium point at the origin is globally uniformly asymptotically stable.*

Next, we develop a feedback controller for the sending rate such that the queue length asymptotically tracks a desired value. The control input must be non-negative since it represents the sending rate, thereby imposing an additional constraint on the model uncertainty $(F(x,t))$ to guarantee tracking. The following lemma presents the control law for tracking the queue length.

Lemma 2 *Consider the fluid-flow model for a single queue served by a single server without any propagation delays from the source (represented by the dynamics in Eq. (4.2), with the bounded model uncertainties described by Eq. (4.3)). A control input of the form $u(t) = \max\Big(-\bar{\alpha}(x,t) + \dot{x}_d - k\, sgn(x(t) - x_d(t)), 0\Big)$, with $k = F(x,t) + \eta$ for some $\eta > 0$, guarantees asymptotic tracking of the desired queue length $(x_d(t))$ if the bounds on the model uncertainty satisfy $F(x,t) < -\big(\bar{\alpha}(x,t) - \dot{x}_d\big)$ when $x > x_d(t)$.*

Proof: A feedback law for the sending rate is determined using a sliding mode control approach [162]. A sliding variable (s) is defined in terms of the tracking error (e) as follows:

$$s = e(t) = x(t) - x_d(t) \quad \Longrightarrow \quad \dot{s} = \alpha(x,t) + u(t) - \dot{x}_d(t).$$

Asymptotic tracking (i.e., $e(t) \to 0$ as $t \to \infty$) of the queue length is achieved by requiring the squared distance to the sliding surface ($s = 0$) decreases along all trajectories:

$$\frac{1}{2}\frac{ds^2}{dt} \leq -\eta|s|, \ \eta > 0 \quad \Longrightarrow \quad s\dot{s} \leq -\eta|s| \tag{4.5}$$

The best approximation to the control input that would achieve $\dot{s} = 0$ is represented by $\hat{u}$, and is given by, $\hat{u} = -\bar{\alpha}(x,t) + \dot{x}_d$. Consider the control law of the form

$$u(t) = \hat{u} - k\,\mathrm{sgn}(s) = -\bar{\alpha}(x,t) + \dot{x}_d - k\,\mathrm{sgn}(s). \tag{4.6}$$

The discontinuity is added across the sliding surface to account for model uncertainties [162]. To determine the gain parameter, k, consider,

$$\begin{aligned} s\dot{s} &= s\big(\alpha(x,t) + u(t) - \dot{x}_d\big) && (4.7)\\ &= s\big(\alpha(x,t) - \bar{\alpha}(x,t) - k\ \mathrm{sgn}(s)\big) && (4.8)\\ &= s\big(\alpha(x,t) - \bar{\alpha}(x,t)\big) - k|s| && (4.9) \end{aligned}$$

If we chose $k \geq F(x,t) + \eta$, then the sliding condition in Eq. (4.5) is satisfied, and therefore the resulting control law in Eq. (4.6) is guaranteed to asymptotically track the desired queue length. However, the control input needs to be non-negative since it represents the sending rate from the source. Therefore, the feedback law is modified as:

$$u(t) = \max\big(\hat{u} - k\,\mathrm{sgn}(s), 0\big) \triangleq \big(\hat{u} - k\,\mathrm{sgn}(s)\big)^{+}. \tag{4.10}$$

Next, we show that the resulting closed loop dynamics asymptotically tracks the desired queue length even with the saturated control input under certain conditions.

We make use of the Lyapunov theorem (Theorem 1) to show that the queue length tracks the desired value using the control input given in Eq. (4.10). Consider the Lyapunov function candidate, $V(s,t) = s^2$. Here, $V(s,t)$ is a positive definite function and decrescent. The time derivative of V is given by

$$\dot{V} = s\dot{s} = s\big(\alpha(x,t) + \big(\hat{u} - k\ \mathrm{sgn}(s)\big)^+ - \dot{x}_d\big). \tag{4.11}$$

When the control input is not saturated ($\hat{u} - k\ \mathrm{sgn}(s) \geq 0$), the sliding condition ensures that $\dot{V}$ is negative definite.

Next, we consider the case when the control input is saturated. Using Eq. (4.10), the condition for control input saturation is given by

$$\big(-\bar{\alpha}(x,t) + \dot{x}_d - (F(x,t) + \eta)\ \mathrm{sgn}(s)\big) < 0. \tag{4.12}$$

The control input can saturate only when $s > 0$, if we choose $\eta > \max\Big((\bar{\alpha} - \dot{x}_d - F(x,t)), 0\Big)$. For the case of saturated control input,

$$\dot{V} \;\leq\; F(x,t)|s| + s\big(\bar{\alpha}(x,t) - \dot{x}_d\big). \tag{4.13}$$

From the above inequality, $\dot{V} \leq -\eta|s|$ if $F(x,t) = -\big(\bar{\alpha}(x,t) - \dot{x}_d\big) + \eta$ for any $\eta > 0$ and $s > 0$. Therefore, $\dot{V}$ is negative definite even with control input saturation if $F(x,t) < -\big(\bar{\alpha}(x,t) - \dot{x}_d\big)$. Hence, the equilibrium $s = 0$ (that corresponds to $x = x_d$), is asymptotically stable, guaranteeing perfect tracking.

□

Remark: The bounds on the uncertainty in the dynamics ($F(x,t)$) becomes more conservative if the desired queue length, $x_d(t)$, has larger fluctuations.

4.2.2 Case with propagation delays

Consider the case in which there is a travel time, τ, to move from the source to the server. Assume that the travel time is a known constant. The queue length is given by

$$\dot{x} = \bar{\alpha}(x,t) + u(t-\tau). \tag{4.14}$$

Applying the delay-free controller that we developed earlier on the time-delay system can lead to poor tracking. To illustrate this, let us consider a simple example where the desired queue length to be tracked is of the form $x_d = 8 + 2\sin(0.1t)$. The system response with the delay-free controller that we developed earlier is shown in Fig. 4-2. If there are no delays, the tracking is perfect with the robust controller as intended. However, the observed queue length deviates from the desired value with the introduction of time-delay, indicating the need to develop a controller that explicitly accounts for it.

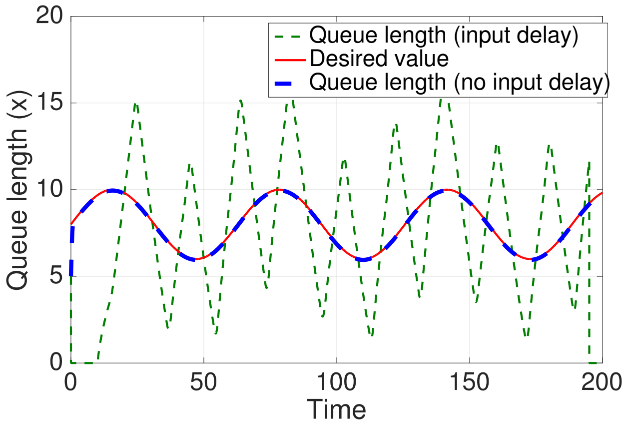


Figure 4-2: Queue length obtained when the delay-free controller is applied on the time-delay system. Parameters: $x_d = 8 + 2\sin(0.1t))$, $\tau = 5,\ C = 1,\ \mu = 1$.

Predictor-based feedback control

A predictor based feedback controller is used to compensate for the time-delays in the system [97]. The system is forward-complete, that is, the state trajectories are well-defined for all $t \geq 0$, for every initial condition and bounded input signal. Forward-completeness ensures that the state does not become unbounded before the control acts on the state due to delays. Let $P_t(t+\tau)$ be the predictor for the state at time $(t+\tau)$, computed at time t. The predictor, $P_t(\theta),\ \forall\theta \in (t, t+\tau]$ is obtained by integrating the model equations with input delays (Eq. (4.14)) forward in time with the initial condition, $P_t(t) = x(t)$, as:

$$P_t(t+\tau) = P_t(t) + \int_{t-\tau}^{t} \Big(\bar{\alpha}\big(P_t(\theta+\tau), \theta+\tau\big) + u(\theta)\Big) d\theta. \tag{4.15}$$

Using the feedback law for the delay-free system (Eq. (4.10)), the predictor-based feedback law for the delayed system is obtained by replacing the current state values with the corresponding predicted states, as:

$$u(t) \;=\; \big(-\bar{\alpha}(P_t(t+\tau), t+\tau) + \dot{x}_d(t+\tau) - k\ \operatorname{sgn}(P_t(t+\tau) - x_d(t+\tau))\big)^{+} \tag{4.16}$$

Next, we determine the tracking guarantees with the above predictor-based feedback law. We determine bounds on the prediction errors, and use those bounds to obtain guarantees for tracking the queue length. The following lemma provides the bounds for the prediction errors.

Lemma 3 *Consider the fluid-flow model for a single queue served by a single server with a known propagation delay (τ) to move from the source to the server (i.e., dynamics given by Eq.* (4.14)*, and bounded model uncertainties as in Eq.* (4.3)-(4.4)*). The error between the predicted queue length (given by Eq.* (4.15)*) and its actual value is bounded by*

$$|P_{t-\tau}(t) - x(t)| < a\tau. \tag{4.17}$$

Proof: For the ease of notation, we denote $\hat{P}(\theta) = P_{t-\tau}(\theta)$, the predictions for $x(\theta)$ computed at time $(t-\tau)$. Using Eq. (4.15), the dynamics of $\hat{P}(\theta)$ is as follows:

$$\dot{\hat{P}}(\theta) = \bar{\alpha}\big(\hat{P}(\theta), \theta\big) + u(\theta - \tau);\ \theta \in [t-\tau, t], \tag{4.18}$$

with the initial condition $\hat{P}(t-\tau) = x(t-\tau)$. Suppose that the actual queue dynamics evolves according to

$$\dot{x} = \alpha(x,t) + u(t-\tau). \tag{4.19}$$

Similar to the delay-free case, we assume that the function $\alpha(x,t)$ is unknown but bounded by $|\alpha(x,t) - \bar{\alpha}(x,t)| < F(x,t)$, and that the time delay (τ) is known. The error in predicting $x(t)$ at time $(t-\tau)$ is given by $\hat{e}(\theta) = e_{t-\tau}(\theta) = \hat{P}(\theta) - x(\theta)$. From Eqs. (4.18)-(4.19), the error dynamics is given by:

$$\dot{\hat{e}}(\theta) = \bar{\alpha}\big(\hat{P}(\theta),\theta\big) - \alpha(x(\theta),\theta); \quad \theta \in [t-\tau, t], \tag{4.20}$$

with the initial condition $e(t-\tau) = 0$. Simplifying further, we obtain

$$\dot{\hat{e}}(\theta) = -\gamma(\theta)\hat{e}(\theta) + d(x(\theta)), \tag{4.21}$$

where $\gamma(\theta) = \mu(\theta)C(\theta)/\Big((C(\theta)\hat{P}(\theta)+1)(C(\theta)x(\theta)+1)\Big)$, and $d(x(\theta)) = \big(\bar{\alpha}(x(\theta),\theta) - \alpha(x(\theta),\theta)\big)$. The error dynamics is stable since $\gamma(\theta) > 0$, and the solution is given by

$$\begin{aligned} \hat{e}(t) &= \exp\Big(-\int_{t-\tau}^{t}\gamma(\theta)d\theta\Big)\hat{e}(t-\tau) \\ &\quad + \int_{t-\tau}^{t}\exp\Big(-\int_{r}^{t}\gamma(\theta)d\theta\Big)d(x(r))dr \leq a\tau, \end{aligned}$$

since $\exp\Big(-\int_r^t \gamma(\theta)d\theta\Big) \leq 1$ (since $\gamma(\theta) > 0$) and $d(x(t)) \leq F(x,t) < a$. Therefore, the error bound (D) for the prediction of $x(t)$ computed at time $t-\tau$ is given by

$$|\hat{P}(t) - x(t)| < D = a\tau. \tag{4.22}$$

□

The above result indicates that the prediction error is proportional to the time delay (τ) and magnitude of uncertainty in the dynamics (a). Next, we investigate the impact of the prediction error on tracking the queue length.

Tracking performance of the controller

Using the results from Lemma 3, we can show that with predictor-based feedback control, the queue length always converges to a set around the desired value. To do so, we define the concept of ultimate boundedness [51].

Definition 7 (Ultimate boundedness for time-delay systems [174]) *The solutions* $x_t(t_0, \phi)$ *of* $\dot{x} = f(t, x_t)$ *are said to be uniformly ultimately bounded if there exists an* $\eta > 0$ *and a* $\hat{t} = \hat{t}(\eta, \delta) > 0$ *independent of* t_0 *such that* $\|x_t(t_0, \phi)\| \leq \eta$ *for all* $t \geq t_0 + \hat{t}$*, when* $|\phi| < \delta$*. Here,* $x_t(t_0, \phi)$ *refers to the solutions of* $\dot{x} = f(t, x_t)$ *with the initial condition* ϕ *at* t_0*.*

The following theorem presents the key result for tracking guarantees using predictor-based feedback control:

Theorem 4 *Consider the fluid-flow model for a single queue served by a single server, with a known propagation delay* (τ) *to move from the source to the server (i.e., dynamics from Eq.* (4.14) *and bounded model uncertainties given by Eq.* (4.3)-(4.4)*). Then, the control input*

$$u(t) = \big(- \bar{\alpha}(P_t(t+\tau), t+\tau) + \dot{x}_d(t+\tau) - k \; sgn(P_t(t+\tau) - x_d(t+\tau))\big)^+ \tag{4.23}$$

with

$$P_t(t+\tau) = \int_{t-\tau}^{t} \Big(\bar{\alpha}\big(P_t(\theta+\tau), \theta+\tau\big) + u(\theta)\Big) d\theta, \;\; P_t(t) = x(t); \tag{4.24}$$
$$\text{and } k = F(x,t) + \eta + \mu C a \tau; \;\; \eta > \big(\bar{\alpha} - \dot{x}_d - (F + \mu C a \tau)\big)^+$$

guarantees that the queue length is uniformly ultimately bounded if $F(x,t) < (-\bar{\alpha}(x,t) + \dot{x}_d)$ *when* $x > x_d + a\tau$*. Furthermore, the ultimate bounds are given by* $\|x - x_d\| < a\tau \; \forall \; t \geq t_0 + T$*, where* $T = \frac{(x_0 - x_d)^2 - a^2\tau^2}{ka\tau}$*.*

Proof: Consider the Lyapunov function candidate, $V(s,t) = s^2$, where $s = (x - x_d)$, as before. The time derivative of V is given by

$$\dot{V} = s\Big(\alpha(x,t) + \big(- \bar{\alpha}(\hat{P}(t),t) + \dot{x}_d - k\,\mathrm{sgn}(\hat{P}(t) - x_d(t))\big)^+ - \dot{x}_d\Big).$$

When $x > x_d + D$, we have $\hat{P}(t) - x_d > 0$ from Eq. (4.17) and $s > 0$. Then,

$$\begin{aligned} \dot{V} &= s\Big(\alpha(x,t) - \bar{\alpha}(\hat{P}(t),t) - k\Big) \\ &\leq sF(x,t) + s\Big(\bar{\alpha}(x,t) - \bar{\alpha}(\hat{P}(t),t)\Big) - sk. \end{aligned}$$

However, since $\big(\bar{\alpha}(x,t) - \bar{\alpha}(\hat{P}(t),t)\big) = \frac{-\mu C(x-P)}{(Cx+1)(CP+1)} \leq \mu CD$, we obtain

$$\dot{V} \leq sF(x,t) + s\mu CD - sk \tag{4.25}$$

For the case when $x < x_d - D$, we have $\hat{P}(t) - x_d < 0$ from Eq. (4.17) and $s < 0$. Then, similarly:

$$\dot{V} \leq |s|F(x,t) + |s|\mu CD + sk. \tag{4.26}$$

Eqs. (4.25)-(4.26) imply that $\dot{V} < 0$ for $|x - x_d| > D$ when the gain, k, is appropriately chosen. With the above results, we can show that queue length always converges to a set around the desired value with predictor-based feedback control, or equivalently, the trajectories of the closed-loop system are uniformly ultimately bounded.

If the gain k in Eqs. (4.25)-(4.26) is chosen such that $k = \mu CD + F(x,t) + \eta,\ \eta > 0$, then we obtain $\dot{V} \leq -\eta|s|$ for $\|s\| > D$. Therefore, for $\|s\| > D$, $\dot{V} \leq -\eta D$, which implies that the trajectory behaves as if the origin ($s = 0$) is asymptotically stable and satisfies an inequality of the form, $s^2 \leq s_0^2 - \eta D(t - t_0)$. The trajectories that start in or those that reach the set, $\{s : \|s\| \leq D\}$, will remain within that set since $\dot{V}$ is negative on the boundary of that set. Therefore, the system is uniformly ultimately bounded with an ultimate bound D, which implies $\|x - x_d\| < D\ \forall\ t \geq t_0 + T$, where $T = \frac{(x_0 - x_d)^2 - D^2}{\eta D}$. Therefore, the queue length converges to a set around the desired queue length, $\{x : \|x - x_d\| < D = a\tau\}$, in finite time. This set depends on the time-delay and model uncertainty.

The above results are valid even when the controller saturates under certain conditions. From Eq. (4.16), the control input saturates if:

$$-\bar{\alpha}(\hat{P}(t), t) + \dot{x}_d - (F + \mu CD + \eta)\ \mathrm{sgn}(\hat{P}(t) - x_d(t)) < 0.$$

If $\eta > \max\Big(\bar{\alpha} - \dot{x}_d - (F + \mu CD), 0\Big)$, then the control input does not saturate when $x < x_d - D$ (from Eq. (4.17)). For the case when the controller saturates and $x > x_d + D$, we have:

$$\dot{V} \leq F(x,t)|s| + s\big(\bar{\alpha}(x,t) - \dot{x}_d\big).$$

From the above inequality, $\dot{V}$ is negative for $x > x_d + D$ if $F(x,t) \leq -\big(\bar{\alpha}(x) - \dot{x}_d\big)$. The condition on $F(x,t)$ is similar to what we had obtained earlier for the case without time-delays. Therefore, even with controller saturation, the above system is uniformly ultimately bounded with these additional conditions. $\square$

The predictor in Eq. (4.24) is computed through numerical integration. To eliminate chattering (i.e., the control input switching at high frequencies at the sliding surface), the *sgn(.)* function in the control input is replaced by a saturation function, $\{\mathrm{sat}(x) = x \text{ if } |x| < 1, \text{ or } \mathrm{sgn}(x) \text{ otherwise}\}$, as is standard practice for sliding mode controllers [162].

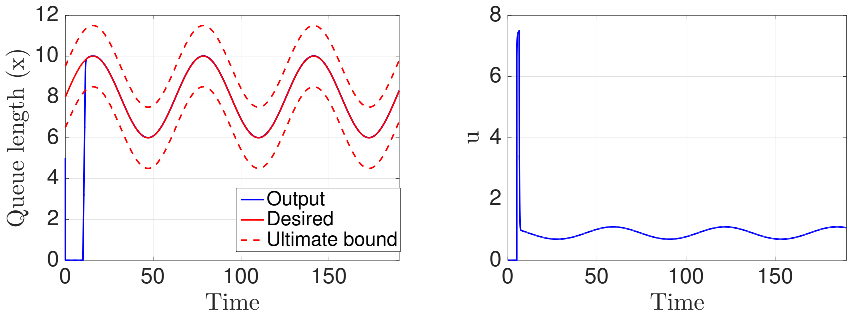
4.2.3 Numerical experiments

To illustrate the tracking performance of the proposed approach, we consider the following sinusoidal reference signal for the desired queue length: $x_d(t) = 8 + 2\sin(0.1t)$, and the parameter values $\mu = 1, C = 1, \tau = 5, a = 0.3$. The sinusoidal reference signal is used to illustrate that the control algorithm can be used to maintain a time-varying desired queue length. The parameters represent a server with exponential service time distribution with a mean equal to one; however, the analysis can be adapted to other general service time distributions by appropriately changing the value of C.

Figure 4-3 shows the control input and the achieved queue length (along with the ultimate bounds) for the case with no model uncertainties. We see that one obtains perfect tracking, and that the control input is continuous (no chattering). Next, we consider the case with model uncertainty. Figure 4-4 shows the resulting queue length for the three cases that correspond to instances when the uncertainty in the model dynamics is either lesser than, equal to, and greater than the assumed value. In these examples, we assume that the actual mean service rate deviates from the model. The output queue length deviates from the desired value, but lies within the ultimate bound when the actual uncertainty is within the assumed range. We note that the ultimate bound ($x_d \pm a\tau$) for the queue length that we have obtained is very tight in this example.

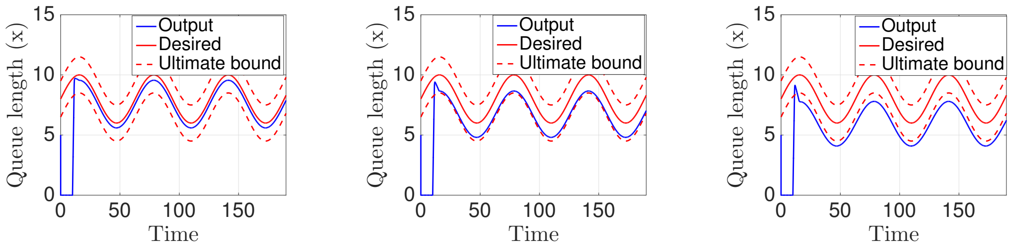
The condition that the model uncertainty being less than the assumed value guaranteeing uniform ultimate boundedness is a sufficient condition but not necessary. We could have trajectories within the ultimate bound even if the model uncertainty is greater than the assumed value when the prediction error over the time-horizon is smaller than $a\tau$. This could happen when the errors in the assumed mean service rate fluctuates over time, resulting in an instantaneous deviation higher than the assumed value, but the prediction error over the time horizon is smaller than the threshold. This observation is particularly important when we apply this control framework to discrete stochastic queuing systems, wherein, the instantaneous deviation of the model might be large but the prediction errors over a horizon are small. Also, note that the ultimate bound collapses to x_d when there are no time-delays. This shows that the robust controller guarantees nearly perfect tracking even in the presence of model uncertainties when there are no delays in the system.

Next, the control law is tested on a queuing simulator instead of the analytical model to test the performance of the controller on a discrete queuing system. The queuing simulator advances in discrete time steps and the customers released at the source join the queue after a time delay that corresponds to the travel time. The customers in the queue are served on a first-come-first-serve basis and the service times of the server are sampled from an exponential distribution. From an implementation point of view, we need to note that the control law (Eq. 4.23) provides a continuous release rate into the queue. However, the



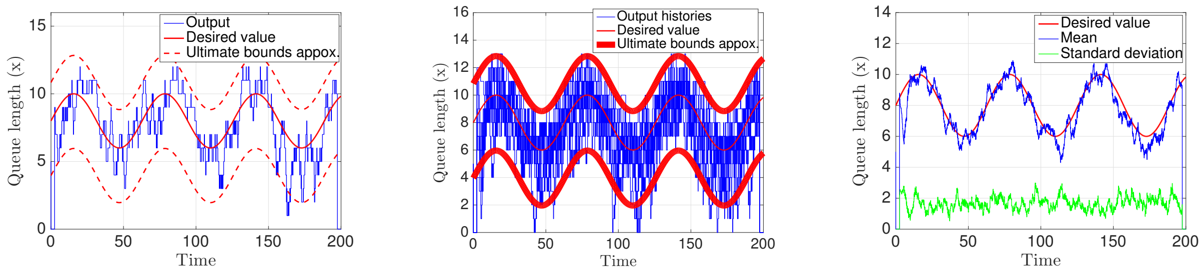
(a) Queue length (b) Control input

Figure 4-3: Output queue length when there are no model uncertainties.



(a) $\mu_a = 1.1$; $|\alpha(x) - \bar{\alpha}(x)| < F(x)$ (b) $\mu_a = 1.3$; $|\alpha(x) - \bar{\alpha}(x)| = F(x)$ (c) $\mu_a = 1.5$; $|\alpha(x) - \bar{\alpha}(x)| > F(x)$

Figure 4-4: Tracking accuracy when the actual service rate (μ_a) is different from the nominal service rate assumed in the model ($\mu = 1$).



(a) Queue length (single realization) (b) Queue length (multiple realizations) (c) Simulation statistics

Figure 4-5: Queue length from the queuing simulation with exponential service time distribution.

simulator (and any physical queuing system) requires the number of customers that need to be released into the queue as the input. So one needs to convert the continuous sending rate, that is the output of the feedback control policy, into discrete number of customers that need to be released into the queue at every time instant. The way we handle this issue is by integrating the release rate with time and releasing a customer when the integral crosses an integer value. Let Δt be the discrete time step of the simulator and $u(t_n)$ be the release rate determined from the feedback law (Eq. (4.23)) at discrete time step t_n. The number of customers released into the queue at time t_n is represented by $U(t_n)$, which is found using,

$$U(t_n) = \left\lfloor \sum_{i=1}^{n} u(t_i)\Delta t - \sum_{i=1}^{n-1} U(t_i) \right\rfloor \tag{4.27}$$

Here, $\lfloor . \rfloor$ represents the floor function. The objective in the simulation experiments is to release customers into the queue to track a desired queue length. The ultimate bounds obtained from the analytical model, $x_d \pm a\tau$, essentially depends on the error bounds for the state predictions ($a\tau$). We obtain tighter empirical bounds for the predictor errors using the simulated data and model predictions. Let E be the empirical distribution of the predictor error obtained from the simulation. We then define an approximate ultimate bound from the simulation data as $[x_d - E_1, x_d - E_{99}]$, where, E_k denotes the k^{th} percentile of the error distribution. Fig. 4-5(a) shows the profile of the desired queue length, achieved queue length and the ultimate bounds for a single realization of the simulation. We can see that the queue length tracks the desired trajectory and stays almost within the bounds. To highlight the fact that the output trajectories remain within the ultimate bounds, we present multiple output trajectories in Fig. 4-5(b). The ensemble mean and standard deviation over multiple realizations (50 in this case) is shown in Fig. 4-5(c). The ensemble mean of the queue length shows better tracking since our analytical model is derived based on capturing the ensemble mean. We can also see that the standard deviation of the resultant trajectory is relatively low as desired. In an operational application, we can also compute the prediction errors of the model using the actual operational data to obtain the error distributions, and then compute the approximate ultimate bounds using the 1^{st} and 99^{th} percentile of the error distribution.

4.3 Control of queues in tandem

To illustrate the methodology for a queuing network, we first consider the simplest queuing network: two queues in tandem as shown in Fig. 4-6. Let x_1 and x_2 be the length of the first and second queue, respectively. Assume that the queues have infinite capacity to hold customers (no blocking). Let μ_1 and μ_2 be the mean service rates of the first and second server, respectively. For the present discussion, assume that the service time distributions are time-invariant. Let the sending rate at the source be $u(t)$. The time taken by a customer to reach the first queue from the source is τ_1 and similarly the time taken to reach the second queue from after completing service in the first queue is τ_2. The objective is to control the arrivals into the first queue to maintain a desired queue length in the second queue. For this objective to be feasible, we assume that the mean service rate of the first server is greater than the second server. The input into the second queue depends on the out-flow rate from the first queue, that is upper bounded by mean service rate of the first queue. Moreover, the out-flow rate from the second queue depends on its mean service rate. As a consequence, there is a physical constraint on the trajectories that can be tracked by the controller.

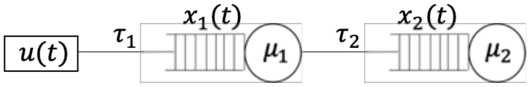


Figure 4-6: Schematic of two queues in tandem.

Using the fluid flow model discussed earlier, the queuing model for tandem queues with propagations delays is given by the following delay differential equations with appropriate initial conditions,

$$\dot{x}_1(t) = -\mu_1 \frac{C_1 x_1(t)}{(1 + C_1 x_1(t))} + u(t - \tau_1) \tag{4.28}$$

$$\dot{x}_2(t) = -\mu_2 \frac{C_2 x_2(t)}{(1 + C_2 x_2(t))} + \mu_1 \frac{C_1 x_1(t - \tau_2)}{(1 + C_1 x_1(t - \tau_2))} \tag{4.29}$$

4.3.1 Controller for dynamics without propagation delays

As we had done in the case of a single queue, we first develop a robust controller for the non-delayed system and then account for the delay terms through a predictor-based feedback. The model for the queue length without time-delay is:

$$\begin{aligned}\dot{x}_1 &= -\mu_1(t)\frac{C_1(t)x_1}{(1+C_1(t)x_1)} + u(t) \\ \dot{x}_2 &= -\mu_2(t)\frac{C_2(t)x_2}{(1+C_2(t)x_2)} + \mu_1(t)\frac{C_1(t)x_1}{(1+C_1(t)x_1)}.\end{aligned} \tag{4.30}$$

The objective is to track the second queue length at a desired value considering model uncertainties. The state that has to be tracked, x_2, needs to be differentiated twice to obtain the control input, leading to the following second-order differential equation:

$$\begin{aligned}\ddot{x}_2 &= \bar{\alpha}(\mathbf{x},t) + \bar{\beta}(\mathbf{x},t)u \\ \bar{\alpha}(\mathbf{x},t) &= -\mu_1^2\frac{C_1^2x_1}{(C_1x_1+1)^3} + \mu_2^2\frac{C_2^2x_2}{(C_2x_2+1)^3} - \mu_1\frac{\mu_2C_1C_2x_1}{(C_2x_2+1)^2(C_1x_1+1)} \\ \bar{\beta}(\mathbf{x},t) &= \frac{\mu_1C_1}{(1+C_1x_1)^2},\end{aligned}$$

where $\mathbf{x} = [x_1, x_2]^T$. We assume that the dynamics for the actual queue length is of the form $\ddot{x}_2 = \alpha(\mathbf{x},t) + \beta(\mathbf{x},t)u$, where $\alpha(\mathbf{x},t)$ and $\beta(\mathbf{x},t)$ are unknown functions. However, we assume that the error between the model and the actual dynamics is bounded. We assume the following form for the error bounds:

$$|\alpha(\mathbf{x},t) - \bar{\alpha}(\mathbf{x},t)| \leq F(\mathbf{x},t), \tag{4.31}$$

$$\tfrac{1}{c}\bar{\beta}(\mathbf{x},t) \leq \beta(\mathbf{x},t) \leq c\bar{\beta}(\mathbf{x},t),\ c > 1. \tag{4.32}$$

Since errors arise primarily due to uncertainties in the service times, we consider the following form for $F(\mathbf{x},t)$:

$$F(\mathbf{x},t) = a_1\frac{C_1^2x_1}{(C_1x_1+1)^3} + a_2\frac{C_2^2x_2}{(C_2x_2+1)^3} + a_3\frac{C_1C_2x_1}{(C_2x_2+1)^2(C_1x_1+1)}$$

where $a_i,\ i = 1, 2, 3$ are constants that determine the level of uncertainty. The goal is to design a sliding controller to have x_2 track a desired queue length, $x_{2,d}(t)$. Assume that the desired trajectory is continuous and twice-differentiable. The sliding variable (s) is defined in terms of the tracking error, $e = x_2 - x_{2,d}$, as $s = \dot{e} + \lambda e,\ \ \lambda > 0$.

$$\dot{s} = \ddot{e} + \lambda\dot{e} = \ddot{x}_2 - \ddot{x}_{2,d} + \lambda\dot{e} = \alpha + \beta u - \ddot{x}_{2,d} + \lambda\dot{e} \tag{4.33}$$

The best approximation of the control input that would achieve $\dot{s} = 0$ is given by $\hat{u} = \bar{\beta}^{-1}(\ddot{x}_{2,d} - \lambda\dot{e} - \bar{\alpha})$. Consider the control law of the form

$$u = \hat{u} - k\ \mathrm{sgn}(s). \tag{4.34}$$

The gain parameter (k) is determined from the sliding condition (Eq. (4.5)) to guarantee asymptotic tracking. Using the definitions for s, the control law Eq. (4.34), and the sliding condition (Eq. (4.5)), we get:

$$\begin{aligned}
s\dot{s} &= s\Big(\alpha + \beta(\hat{u} - k\ \mathrm{sgn}(s)) - \ddot{x}_{2,d} + \lambda\dot{e}\Big)\\
&= s\Big(\alpha + \beta(\bar{\beta}^{-1}(\ddot{x}_{2,d} - \lambda\dot{e} - \bar{\alpha}) - k\ \mathrm{sgn}(s)) - \ddot{x}_{2,d} + \lambda\dot{e}\Big)\\
&= s\Big(\alpha - \beta\bar{\beta}^{-1}\bar{\alpha}\Big) + s(-\ddot{x}_{2,d} + \lambda\dot{e})(1 - \beta\bar{\beta}^{-1}) - \beta k|s|\\
&= s\Big(\alpha - \bar{\alpha}\Big) + (\bar{\alpha} - \ddot{x}_{2,d} + \lambda\dot{e})(1 - \beta\bar{\beta}^{-1}) - \beta k|s|\\
&= s\Big(\alpha - \bar{\alpha}\Big) - \bar{\beta}\hat{u}(1 - \beta\bar{\beta}^{-1}) - \beta k|s|
\end{aligned}$$

Consider $k \geq \bar{\beta}^{-1}c(F + \eta) + (c - 1)|\hat{u}|$, then,

$$s\dot{s} \leq s\Big(\alpha - \bar{\alpha}\Big) - \hat{u}(\bar{\beta} - \beta) - \beta\Big(\bar{\beta}^{-1}c(F + \eta) + (c - 1)|\hat{u}|\Big)|s| \tag{4.35}$$

$$s\dot{s} \leq s\Big(\alpha - \bar{\alpha}\Big) - \Big(\beta\bar{\beta}^{-1}c(F + \eta)\Big)|s| - \hat{u}\beta(\bar{\beta}\beta^{-1} - 1) - \beta(c - 1)|\hat{u}||s| \tag{4.36}$$

From the assumptions on the error bounds (Eqs. (4.31)-(4.32)), we obtain the following conditions: $\beta\bar{\beta}^{-1}c \geq 1$, $(c-1) \geq (\bar{\beta}\beta^{-1}-1)$, $(\alpha - \bar{\alpha}) \leq F$, and hence Eq. (4.36) implies $s\dot{s} \leq -\eta|s|$. Therefore, the gain parameter (k) if chosen as per the above condition satisfies the sliding condition (Eq. (4.5)) to guarantee asymptotic tracking. To reduce the effect of chattering and to account for the non-negativity of the control input, we modify the feedback law (as done in the single queue case) as follows:

$$u = \big(\hat{u} - k\ \mathrm{sat}(s)\big).^{+}$$

As a result of saturating the control law, we lose the tracking guarantees obtained from the sliding mode controller. In the case of a single queue, we were able to obtain conditions for tracking even with controller saturation. While the analysis that was presented for the case of a single queue is difficult to extend to the case of multiple queues, we show that the feedback law performs well in numerical experiments.

Evaluating controller performance

To evaluate the performance of the feedback controller, we consider the case in which the service time of the servers follows an exponential distribution. We first illustrate the performance of the controller by applying the feedback law to the model equations with uncertain service rates. The state derivative that is required for the feedback law is estimated using the model equations. Fig. 4-7 shows the control input and the resulting queue length for the case in which the desired queue length is sinusoidal ($x_{2,d} = 8 + 2\sin(0.1t)$). The model parameters used to develop the feedback control law are as follows: $\mu_1 = 2,\ \mu_2 = 1,\ C_1 = C_2 = 1$. In this case, the actual service rates are considered to be 1.2 times the nominal service rates that are used to derive the feedback controller. We can see that the controller offers very good tracking even though the service rate used in the model differs from the actual service rate, demonstrating robustness to model uncertainties.

Next, the feedback controller is tested on a stochastic queuing simulator to access its performance on the discrete queuing system. The continuous sending rate obtained from the control law is converted into discrete sending times in the same way as done earlier for the

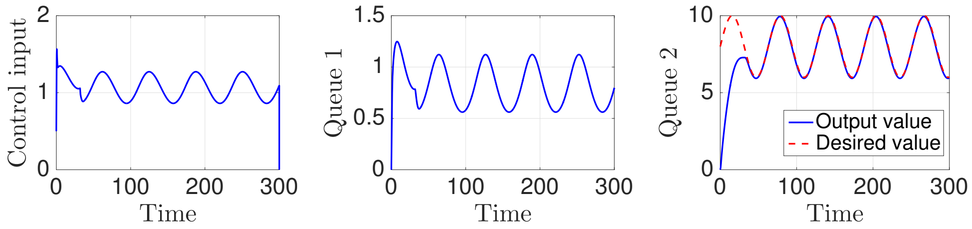
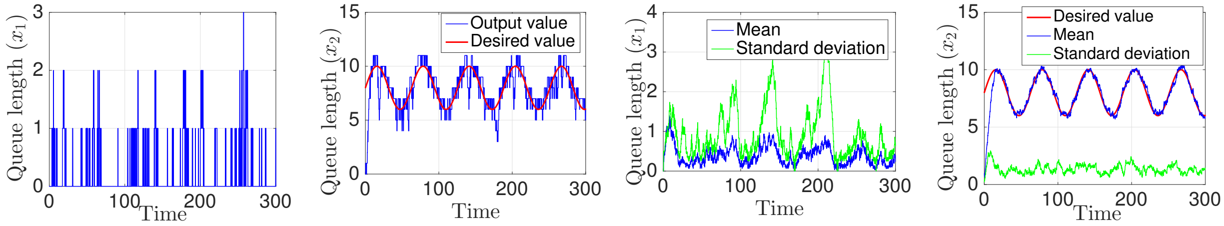


Figure 4-7: Controller performance for tracking the length of the second queue.



(a) Single realization - x_1 (b) Single realization - x_2 (c) Simulation statistics of x_1 (d) Simulation statistics of x_2

Figure 4-8: Single realization and statistics over multiple realizations of the queue length. Simulation parameters: $\mu_1 = 2$, $\mu_2 = 1$, $\lambda = 23$, $\eta = 0.5$, $\Delta t = 0.1, x_{2,d} = 8 + 2\sin(0.1t)$

single queue case. In addition, we drop the state dependence on the nominal value of the control gain and consider $\bar{\beta}(x) = \bar{\beta} = 3$. Figures 4-8(a)-4-8(b) show a single realization of the first and second queue length obtained from the simulations. We see that the length of the second queue remains close to the desired value. The mean and standard deviation of the queue length calculated from 30 sample realizations are shown in Figs. 4-8(a)-4-8(b). We can see that the mean queue length is close to the desired value and the standard deviation is relatively small.

Comparison with an exact solution

The solution obtained using the sliding mode controller is compared with an exact solution obtained using Markov Decision Process (MDP). We can represent the queuing process as a Markov chain (Fig. 4-9) for the case when the service time is exponentially distributed and there are no propagation delays ($\tau_1 = \tau_2 = 0$). The state of the Markov chain is $X = (i, j)$, with i being the length of the first queue and j being the length of the second queue. The control input, u, determines the number of customers released to the first queue.

The transition probability (P) of the Markov chain conditioned on the control input, $u = k$, is as follows,

$$p_{(i,j)\to(l,m)|u=k} = \begin{cases} \mu_1 \Delta t \delta_i, \text{if } (l,m) = (i-1+k, j+1) \& (i \neq 0 | k > 0) \\ \mu_2 \Delta t \delta_j, \text{if } (l,m) = (i+k, j-1) \& (j \neq 0) \\ 1 - (\mu_1 \delta_i + \mu_2 \delta_j) \Delta t, \text{if } (l,m) = (i+k, j) \\ 0, \text{otherwise.} \end{cases} \tag{4.37}$$

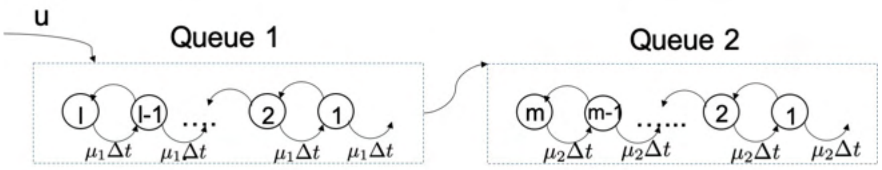


Figure 4-9: Markov chain representation for tandem queue

Here, $\delta_m = 0$ if $m = 0$, or 1 otherwise. The aim of the control problem is to track the length of the second queue at some desired value (x_d). Here, we consider the desired value to be a constant to obtain a static policy. The cost function ($C(X, u)$) for the MDP is considered to be the following,

$$C(X, u) = (j - x_d)^2 \tag{4.38}$$

The cost function penalizes the deviation from the desired queue length. The $s - horizon$ cost under a stationary control policy π is given by,

$$J_s(X, \pi) = E_\pi \Big[\sum_{n=0}^{s-1} \gamma^n C(X_n, u_n) | X_o = X \Big] \tag{4.39}$$

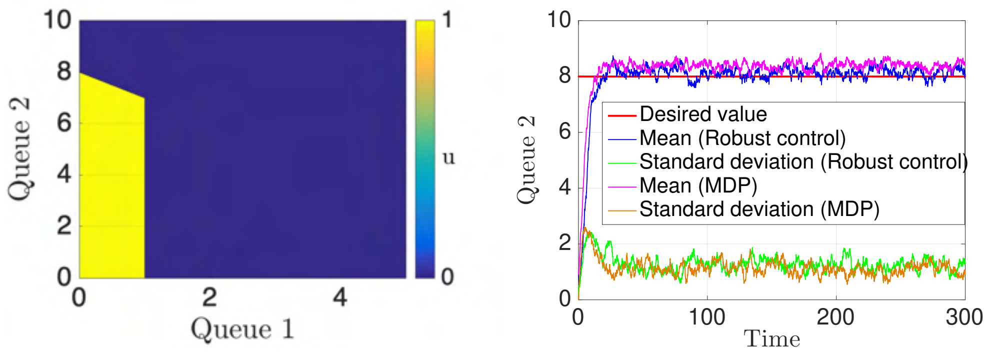
where, E_π denotes the expectation over the path of the process under policy π, γ is the discount factor and (X_n, u_n) denotes the state and control input pair at time $n\Delta t$. The $infinite - horizon$ cost under policy π is given by,

$$J(X,\pi) = \lim_{s\to\infty} J_s(X,\pi) \tag{4.40}$$

The optimal stationary policy π^* satisfies the Bellman equation,

$$J^*(X) = \min_u \Big[C(X,u) + \gamma \sum_{X'} P_{X\to X'} J^*(X')\Big] \tag{4.41}$$

The optimal policy, that is the solution to the above equation, is obtained using value iteration. Fig. 4-10(a) shows the control policy as a function of the queue length for $x_d = 8,\ \mu_1 = 2,\ \mu_2 = 1$. It is interesting to note that the control policy is to release one customer only when there is one or less customers in the first queue and the lesser than the desired value in the second queue. The mean and standard deviation of the second queue length obtained from the stochastic simulations (over 30 sample realizations) with the MDP control policy is shown in Fig. 4-10(b). The figure also shows the mean and standard deviation of the queue length obtained from the sliding mode controller that was presented earlier. The mean queue length obtained from both the control methodologies are close to the desired value. The standard deviation of the queue length obtained from the sliding controller is only slightly more compared to the optimal solution obtained from the MDP, indicating that the performance of the sliding controller closely matches the optimal solution in this case. One needs to note that the performance of the two solutions can slightly vary depending on the parameters (cost function for MDP and gain parameters for the sliding mode controller). The main drawback of the MDP solution is that the methodology cannot be extended to general cases where the queues have a service time distribution that is not exponential or time-varying. It can also be challenging to extend the methodology to a large network of queues due to the issue of curse of dimensionality or to incorporate time-delays due to propagation. These disadvantages can be overcome using the proposed sliding mode controller with a small loss in performance.

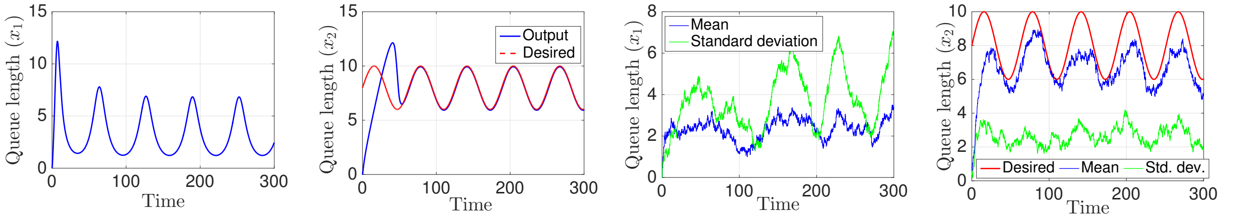


(a) Control input as a function of the queue length (b) Mean and variance of the queue length

Figure 4-10: Control policy and queue length obtained from simulation ($x_d = 8, \ \mu_1 = 2, \ \mu_2 = 1, \ \Delta t = 0.1, \ \gamma = 0.99$).

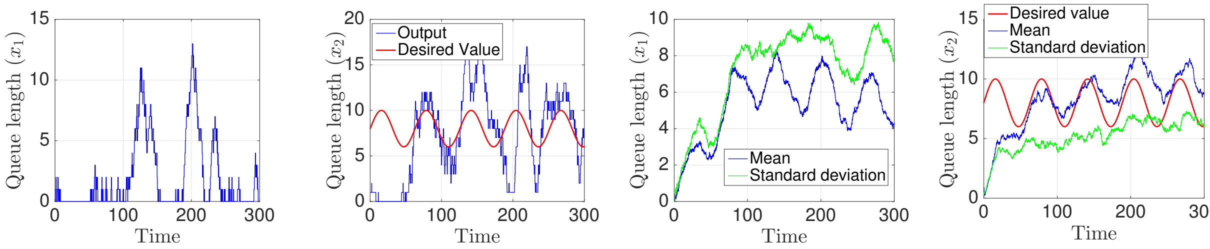
Case when the service rates are comparable

In the earlier cases, we had considered the mean service rate of the first server to be significantly greater than the mean service rate of the second server. This resulted in negligible queuing in the first queue while tracking the queue length of the second queue. Consider the case when the mean service rate of the first server is comparable to the second server (but still needs to be smaller than the second one for the tracking problem to be feasible). The reason for considering this particular case is to analyse the impact of the tracking performance when there is significant queuing even in the first queue. Fig. 4-11 shows the queue length obtained with the feedback controller for the analytical model as well as the simulations (50 trials). We can see a higher queue length for the first queue compared to the previous case as expected. We also notice perfect tracking for the analytical model. However, there is some tracking error even for the mean queue length in the simulations. This is primarily due to the fact that when the first queue is not empty (or the first server has a high utilization), then the outflow rate from the first queue which is the inflow rate into the second queue is determined by the service rate of the first server and it cannot be manipulated much in order to track the second queue length.



(a) Analytical model-x_1 (b) Analytical model-x_2 (c) Simulation statistics-x_1 (d) Simulation statistics-x_2

Figure 4-11: Queue length for tandem queue without propagation delays.$x_{2,d} = 8+2\sin(0.1t)$. Simulation parameters: $\mu_1 = 1.25$, $\mu_2 = 1$, $\lambda = 0.3$, $\eta = 0.5$, $\Delta t = 0.1$



(a) Single realization - x_1 (b) Single realization - x_2 (c) Simulation statistics - x_1 (d) Simulation statistics - x_2

Figure 4-12: Queue length for tandem queue obtained by using a controller without considering the first queue length information in the feedback.

Controller without considering the first queue length information in the feedback

Although the second queue is the primary bottleneck, controlling the second queue without considering the first queue length information in the feedback leads to poor tracking performance. This is more pronounced when there is significant queuing in the first queue that occurs when the service rate of the first server is close to the service rate of the second server. Fig. 4-12 shows the queue length obtained when only the second queue length is considered in the feedback using the controller developed for a single queue. In this case, the mean service rate of the first server is 1.25 and the mean service rate of the second server is 1. We see that the tracking performance is poor in this case (compared to Fig. 4-11), noticeable from the higher standard deviation of the second queue length.

4.3.2 Controller for tandem queues with propagation delays

The model for tandem queues with propagation delays was shown earlier in Eqs. (4.28)-(4.29). The system has delays in the state and control input. Also, note that the system is forward complete. We can use a predictor-based feedback controller for tracking the queue length. Using feedback controller for the non-delayed dynamics presented earlier (Eq. 4.34), we can write the predictor based controller for time-delayed system as follows,

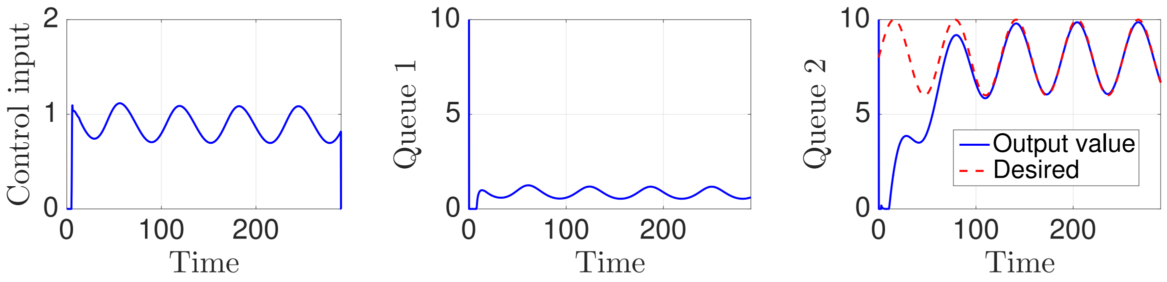
$$u(t) = \max\Big(\bar{\beta}^{-1}(\mathbf{x_p})[\hat{u}(\mathbf{x_p}) - k(\mathbf{x_p})\ \mathrm{sgn}(s(\mathbf{x_p}))], 0\Big) \tag{4.42}$$

Here, $\mathbf{x_p}(t) = (P_{1,t}(t+\tau_1), P_{2,t}(t+\tau_1+\tau_2))$, with $P_{i,t}(.)$ defined as follows:

$$\begin{aligned}
P_{1,t}(t+\tau_1) =& x_1(t) + \int_{t-\tau_1}^{t} \Bigg(-\mu_1 \frac{C_1 P_{1,t}(\theta+\tau_1)}{(1+C_1 P_{1,t}(\theta+\tau_1))} + u(\theta) \Bigg) d\theta, \\
P_{2,t}(t+\tau_1+\tau_2) =& x_2(t) + \int_{t-\tau_2-\tau_1}^{t} \Bigg(-\mu_2 \frac{C_2 P_{2,t}(\theta+\tau_1+\tau_2)}{(1+C_2 P_{2,t}(\theta+\tau_1+\tau_2))} \Bigg) d\theta + \int_{t-\tau_2}^{t} \Bigg(\mu_1 \frac{C_1 P_{1,t}(\theta+\tau_1)}{(1+C_1 P_{1,t}(\theta+\tau_1))} \Bigg) d\theta \\
&+ \int_{t-\tau_1}^{t} \Bigg(\mu_1 \frac{C_1 x_1(\theta)}{(1+C_1 x_1(\theta))} \Bigg) d\theta
\end{aligned}$$

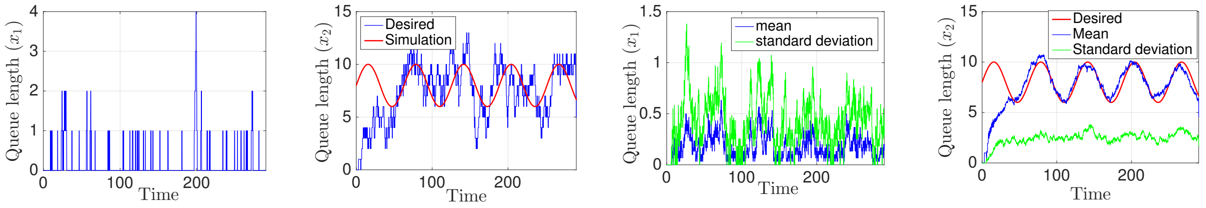
with appropriate initial conditions for the state predictors, $P_{1,t}(t) = x_1(t)$ and $P_{2,t}(t) = x_2(t)$. If the server parameters are time varying, the integrands in the above equations need to be modified to account for this factor.

The performance of the predictor based control law for tracking the second queue length is shown in Fig. 4-13. For this case, we consider the following model parameters: $\mu_1 = 2,\ \mu_2 = 1, \tau_1 = \tau_2 = 2.5, C_1 = C_2 = 1$. The control parameters (λ, η) are picked to avoid overshoot. We see that output queue length tracks the desired sinusoidal value after a small initial transient. Fig. 4-14 shows the tracking performance for discrete stochastic queuing simulations and the queue length statistics are computed over 30 realizations. We can see that the ensemble mean queue length of the second queue closely matches the desired value. However, the queue length has higher variability compared to the case without time delays as one would expect. Next, we apply the robust controller for a queuing network that represents the traffic movement on an airport surface to reduce congestion.



(a) Control input (b) Queue 1 (c) Queue 2

Figure 4-13: Tandem queues with delays ($\mu_1 = 2,\ \mu_2 = 1, \tau_1 = \tau_2 = 2.5, \eta = 0.1, \lambda = 0.04, x_{2,d} = 8 + 2\sin(0.1t)$).



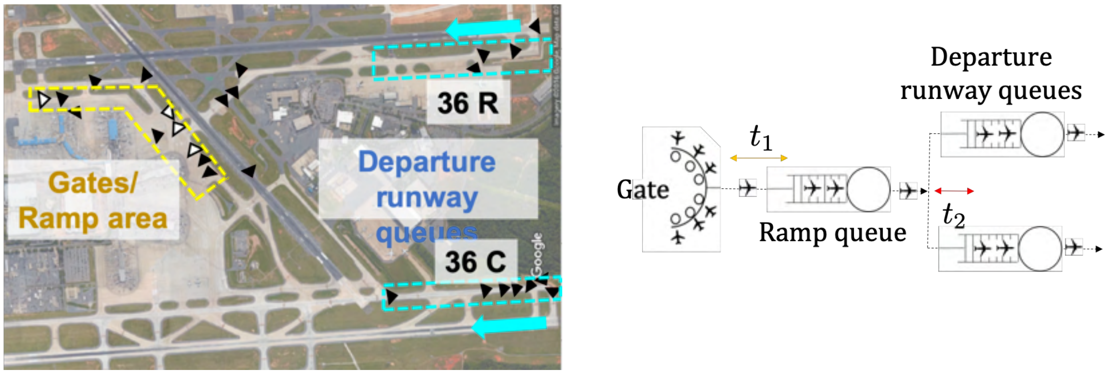
(a) Single realization of first queue (b) Single realization of the second queue (c) Statistics over multiple realization- first queue (d) Statistics over multiple realization- second queue

Figure 4-14: Queue length for tandem queue with propagation delays.

4.4 Application to airport surface operations

We utilize the queuing network model presented earlier for CLT-NF to illustrate the application of the robust control strategy for departure metering. In particular, we focus on the movement of taxi-out flights since that is of primary interest for departure metering.

We provide a brief summary of the model before presenting the control framework. Figure 4-15(a) shows a snapshot of aircraft positions at CLT. The black triangles represent the taxi-out flights (departures) and the white triangles represent the taxi-in flights (arrivals). We observe that the taxi-out flights form long queues in the ramp area (close to the airport terminals) as well as near the departure runways. The movement of departures can be represented as a queuing network as shown in Fig. 4-15(b). The departures pass through a taxi-out ramp queue and one of the two runway queues, depending on their runway assignment (36C or 36R). The corresponding queuing model for the departure process is given by



(a) Layout of CLT with a focus on departure operations.

(b) Queuing representation for the departure operations at CLT.

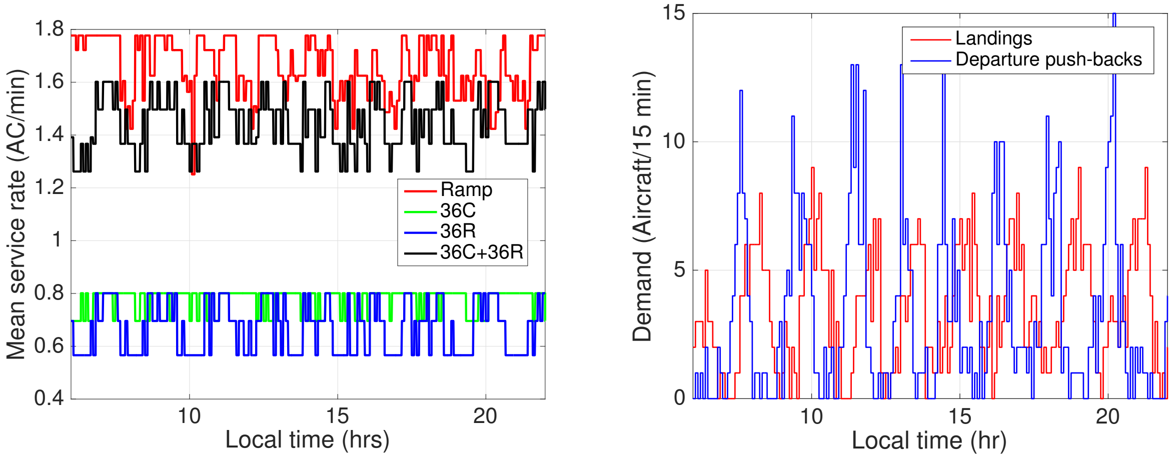
Figure 4-15: Airport layout and queuing representation.

the following set of delay differential equations,

$$
\begin{aligned}
x_s(t) &= x_{s_1}(t) + x_{s_2}(t) && (4.43)\\
\dot{x}_{s_i}(t) &= -\mu_s(t)\frac{C_s(t)x_{s_i}(t)}{1 + C_s x_s(t)} + u_{r_i}(t - t_1), \quad i = 1, 2; && (4.44)\\
\dot{x}_{r_i}(t) &= -\mu_{r_i}(t)\frac{C_{r_i}x_{r_i}(t)}{1 + C_{r_i}x_{r_i}(t)} + \mu_s(t - t_1)\frac{C_s(t - t_2)x_{s_i}(t - t_2)}{1 + C_s(t - t_2)x_s(t - t_2)} && (4.45)
\end{aligned}
$$

Here, x_{s_i} represents the number of flights in the ramp queue headed to runway i, x_{r_i} represents the number of flights in the i^{th} departure runway queue, (C_s, μ_s) are the server parameters of the ramp server, (C_{r_i}, μ_{r_i}) are server parameters of the i^{th} departure runway server, t_1 is the average unimpeded time from the gate to the spot (exit of the ramp), t_2 is the average unimpeded time from the spot to the departure runway, and u_{r_i} is the pushback rate to the i^{th} departure runway. The pushback rate is the number aircraft in a given time interval (5 min intervals in this work) that pushback from the gate before the start of taxi.

We assume that the service time distributions are piece-wise constant over 5 min intervals. Fig. 4-16(a) illustrates the variation of the mean service rate for the ramp and runway servers on a typical good weather day. The fluctuation in the mean service rate in this case is because of the variation in the arrival traffic level over the course of the day. The service rate of the ramp and runway servers can vary by about 20%. One can also notice that the service rate of the ramp server is slightly more than the sum of the service rates of the two runway servers. This reaffirms that the critical bottleneck are the runway servers. The arrival and departure demand at the airport also fluctuates throughout the day as shown in Fig. 4-16(b). This highlights the need for a dynamic policy to reduce congestion that can account for these factors.



(a) Mean service rate of ramp and runway servers

(b) Departure and arrival demand

Figure 4-16: Variation of the mean service and demand for a typical good weather day (07/12/2015).

4.4.1 Departure metering: controlling departure queue length

Lower taxi-out times can be achieved by maintaining smaller queues on the airport surface. In the current scenario, pilots pushback whenever they are ready to go. There are intervals during the day when many aircraft pushback around the same time, leading to the formation of large queues. The objective is to tactically control departure push-backs in order to maintain a desired runway queue length. Ideally, one would like to have a queue length of zero, so that the departing aircraft can take-off without waiting on the airport surface. However, this ideal goal is almost impossible to achieve without delaying take-offs (or under-utilizing the runway) because of various uncertainties present in the system. Hence, the trade-off is between having a smaller queue length and not under-utilizing the runway. We will later discuss how to pick the desired queue length so as to not under-utilize the runway.

First, we present a methodology for determining the pushback times to achieve a desired runway queue length. We use a similar approach that we had presented earlier for single and tandem queues, a sliding mode controller is developed for the non-delayed dynamics, followed by a predictor based control law for handling the time delays. The key difference here is that the queueing model for the airport surface leads to a multi-input-multi-output system unlike the single-input-single-output systems discussed earlier. The outputs to be tracked are the two runway queue lengths $(x_{r_1},\ x_{r_2})$ and the inputs correspond to the pushback rate to each runway (u_{r_1}, u_{r_2}). The inputs do not directly appear in the equations of the output dynamics. Differentiating the outputs twice yields the following second order differential equations containing the inputs (for the non-delayed dynamics),

$$\ddot{\mathbf{y}} = \bar{\alpha}(\mathbf{x}, t) + \bar{\beta}(\mathbf{x}, t)\mathbf{u} \tag{4.46}$$

Here, $\mathbf{x} = [x_{s_1}, x_{s_2}, x_{r_1}, x_{r_2}]^T$ and $\mathbf{u} = [u_{r_1}, u_{r_2}]^T$. The functions $\bar{\alpha}(\mathbf{x}, t) \in R^2$ and $\bar{\beta}(\mathbf{x}, t) \in R^{2\times 2}$ are given by,

$$\bar{\alpha}(\mathbf{x}, t) = \begin{bmatrix} \mu_{r_1}^2 \frac{C_{r_1} x_{r_1}}{(1+C_{r_1} x_{r_1})^2} - \mu_{r_1}\ \mu_s \frac{C_s x_{s_1}}{1+C_s x_s} - \mu_s^2 \frac{C_s^2 x_{s_1}}{(1+C_s x_s)^2} \\ \mu_{r_2}^2 \frac{C_{r_2} x_{r_2}}{(1+C_{r_2} x_{r_2})^2} - \mu_{r_2}\ \mu_s \frac{C_s x_{s_2}}{1+C_s x_s} - \mu_s^2 \frac{C_s^2 x_{s_2}}{(1+C_s x_s)^2} \end{bmatrix} \quad \bar{\beta}(\mathbf{x}, t) = \begin{bmatrix} \mu_s \frac{C_s(1+C_s x_{s_2})}{(1+C_s x_s)^2} & -\mu_s \frac{C_s^2 x_{s_1}}{(1+C_s x_s)^2} \\ -\mu_s \frac{C_s^2 x_{s_2}}{(1+C_s x_s)^2} & \mu_s \frac{C_s(1+C_s x_{s_1})}{(1+C_s x_s)^2} \end{bmatrix} \tag{4.47}$$

The terms containing the derivatives of the server parameters in the queue model are ignored (note that the service time distribution is considered to be a piecewise constant). As done earlier, we shall assume that the actual dynamics is of the following form,

$$\ddot{\mathbf{y}} = \alpha(\mathbf{x}, t) + \beta(\mathbf{x}, t)\mathbf{u} \tag{4.48}$$

Here, $\alpha(.)$ and $\beta(.)$ are unknown functions, with the following error bounds,

$$|\alpha_i(\mathbf{x},t) - \bar{\alpha}_i(\mathbf{x},t)| < F_i(\mathbf{x},t),\ i = 1,2. \tag{4.49}$$

$$\beta(\mathbf{x},t) = (\mathbf{I} + \Delta)\bar{\beta}(\mathbf{x},t), \quad |\Delta_{ij}| < D_{ij};\ i,j = 1,2. \tag{4.50}$$

Motivated by the fact that the uncertainties arise primarily in the service times in the actual system, we consider the following form for $\mathbf{F}$,

$$\mathbf{F}(\mathbf{x},t) = \begin{bmatrix} a_1 \frac{C_{r_1} x_{r_1}}{(1+C_{r_1} x_{r_1})^2} + a_2 \frac{C_s x_{s_1}}{1+C_s x_s} + a_3 \frac{C_s^2 x_{s_1}}{(1+C_s x_s)^2} \\ \\ a_4 \frac{C_{r2} x_{r_2}}{(1+C_{r_2} x_{r_2})^2} + a_5 \frac{C_s x_{s_2}}{1+C_s x_s} + a_6 \frac{C_s^2 x_{s_2}}{(1+C_s x_s)^2} \end{bmatrix} \tag{4.51}$$

The a_is and D_{ij}s are design parameters that need to be picked appropriately depending on the level of uncertainty in the system. The goal is to track the runway queue length at a desired value, represented by $x_{r_{id}}$, for each runway i. The sliding variable ($\mathbf{s} \in \mathbb{R}^2$) is defined in terms of the tracking error ($\mathbf{e}$) as follows,

$$s_i = \dot{e}_i + \lambda e_i; \quad e_i = x_{r_i} - x_{r_{id}}, \quad i = 1,2 \tag{4.52}$$

Perfect tracking is guaranteed for the model dynamics if the sliding variable satisfies the following sliding condition,

$$\frac{1}{2}\frac{d}{dt} s_i^2 \leq -\eta_i |s_i|,\ \eta_i > 0 \implies s_i \dot{s}_i \leq -\eta_i |s_i| \tag{4.53}$$

From the definition of $\mathbf{s}$, we have,

$$\dot{s}_i = \ddot{x}_{r_{id}} - \ddot{x}_{r_{id}} + \lambda \dot{e}_i \tag{4.54}$$

$$= \alpha_i(\mathbf{x},t) + \sum_{j=1}^{2} \beta_{ij}(\mathbf{x},t) u_{rj} - \ddot{x}_{r_{id}} + \lambda \dot{e}_i \tag{4.55}$$

Consider the control input of the form,

$$\mathbf{u} = \bar{\beta}^{-1}(\bar{\mathbf{u}} - \mathbf{k} \odot \operatorname{sgn}(\mathbf{s})); \tag{4.56}$$

Here, $\bar{\mathbf{u}} = (-\bar{\alpha}(\mathbf{x},t) + \ddot{x}_{r_{id}} - \lambda\dot{e}_i)$ and $\odot$ represents element-wise multiplication operation. The value of $\mathbf{k}$ is obtained from the sliding condition to achieve perfect tracking (4.53),

$$-\eta_i|s_i| \geq s_i\dot{s}_i, \quad i = 1,2 \tag{4.57}$$

$$-\eta_i|s_i| \geq s_i\Big(\alpha_i(\mathbf{x},t) - \ddot{x}_{r_{id}} + \lambda\dot{e}_i + \big(\beta\mathbf{u}\big)_i\Big) \tag{4.58}$$

$$-\eta_i|s_i| \geq s_i\Big(\alpha_i(\mathbf{x},t) - \ddot{x}_{r_{id}} + \lambda\dot{e}_i + \big(\beta\big(\bar{\beta}^{-1}(\bar{\mathbf{u}} - \mathbf{k}\odot\operatorname{sgn}(\mathbf{s}))\big)\big)_i\Big) \tag{4.59}$$

$$-\eta_i|s_i| \geq s_i\Big(\alpha_i(\mathbf{x},t) - \ddot{x}_{r_{id}} + \lambda\dot{e}_i + \big((I+\Delta)(\bar{\mathbf{u}} - \mathbf{k}\odot\operatorname{sgn}(\mathbf{s}))\big)_i\Big) \tag{4.60}$$

$$\begin{aligned}-\eta_i|s_i| \geq s_i\Big(&\alpha_i(\mathbf{x},t) - \ddot{x}_{r_{id}} + \lambda\dot{e}_i + \bar{u}_{rj} - k_i\operatorname{sgn}(s_i)\\ &+ \sum_{j=1}^{2}\Delta_{ij}\big(\bar{u}_{rj} - k_j\operatorname{sgn}(s_j)\big)\Big)\end{aligned} \tag{4.61}$$

$$-\eta_i|s_i| \geq s_i\Big(\alpha_i(\mathbf{x},t) - \bar{\alpha}_i(\mathbf{x}) - k_i\operatorname{sgn}(s_i) + \sum_{j=1}^{2}\Delta_{ij}\big(\bar{u}_{rj} - k_j\operatorname{sgn}(s_j)\big)\Big) \tag{4.62}$$

$$-\eta_i|s_i| \geq s_i\Big(\big(\alpha_i(\mathbf{x}) - \bar{\alpha}_i(\mathbf{x})\big) + \sum_{j=1}^{2}\Delta_{ij}\bar{u}_{rj} - (1+\Delta_{ii})k_i\operatorname{sgn}(s_i) - \sum_{j=1,j\neq i}^{2}\Delta_{ij}k_j\operatorname{sgn}(s_j)\Big) \tag{4.63}$$

Using triangular inequality and constraints on the error bounds, we get,

$$F_i(\mathbf{x},t) + \sum_{j=1}^{2} D_{ij}|\bar{u}_j(\mathbf{x},t)| - \sum_{j=1,j\neq i}^{2} D_{ij}k_j + \eta_i \leq (1 - D_{ii})k_i \tag{4.64}$$

A particular value of $\mathbf{k}$ is chosen by solving the following linear equation,

$$(1 - D_{ii})k_i + \sum_{j=1,j\neq i}^{2} D_{ij}k_j = F_i + \sum_{j=1}^{2} D_{ij}|\bar{u}_{rj}(\mathbf{x},t)| + \eta_i \tag{4.65}$$

The control law in Eq. (4.56) is guaranteed to track the desired queue length if the value for $\mathbf{k}$ is chosen such that it satisfies the above equation. However, the control input needs to be saturated at zero since the pushback rate cannot be negative. The time delays in the original model dynamics is handled using a predictor-based feedback. We consider the following substitutions in the delay-free feedback law to account for time-delays: $x_{s_i}(t) \to P_{s_i,t}(t+t_1)$, $(\mu_s(t), C_s(t)) \to (\mu_s(t+t_1), C_s(t+t_1))$, $x_{r_i}(t) \to P_{r_i,t}(t+t_1+t_2)$, $x_{r_i,d}(t) \to x_{r_i,d}(t+t_1+t_2)$,

$(\mu_{r_i}(t), C_{r_i}(t)) \to (\mu_{r_i}(t+t_1+t_2), C_{r_i}(t+t_1+t_2))$. Here, $P_{y,t}(t+z)$ refers to the prediction of state $y(t+z)$ computed at time t. The derivatives of the states present in the control input are also transformed in a similar way. We use the delay differential equations to compute the predictions of the derivatives given the predictions of the states. However, there are no guarantees for tracking since we have saturated the control input and considered a predictor based feedback for time-delays. Nevertheless, we show that the control law performs well through numerical simulations.

The pushback rate decisions need to be converted into flight specific hold decisions. The day is divided into 5-min intervals and decisions are made at the beginning of every interval. At the beginning of each interval, t, the pushback rate is determined for $t + T_p$, where T_p is the planning horizon. A planning horizon is included to improve predictability in the system. The planning horizon essentially introduces additional input delay in the dynamics. The number of aircraft that can be released during each 5-min window (n) is determined from the pushback rate. The first n aircraft in the 5-min window are released as per the control decision, and remaining aircraft are pushed to the beginning of the next time window, awaiting decision for release. This approach specifies the flights that need to be released in $[t+T_p, t+T_p+5]$ time window, and postpones the remaining flights to the next time window.

4.5 Evaluating the performance of departure metering

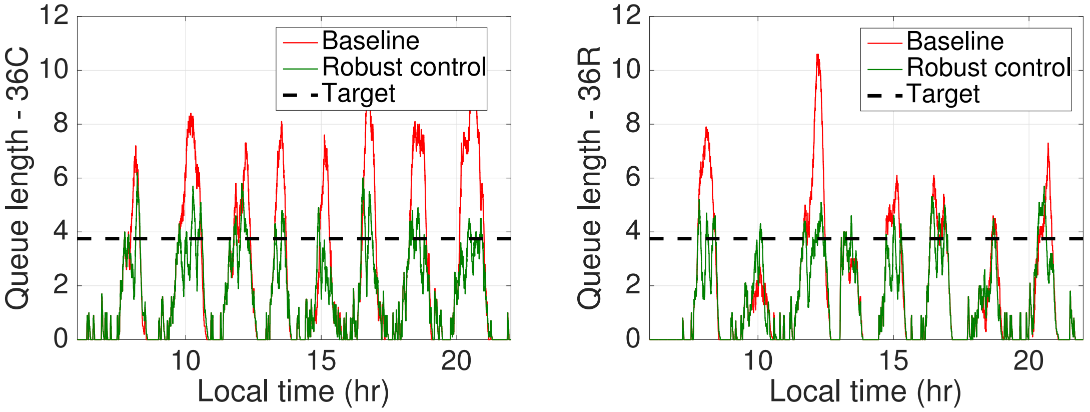
4.5.1 Benefits of departure metering

The performance of the departure metering algorithm is evaluated using simulations of the airport surface as done in the previous chapter for the other departure metering algorithms. The simulator is based on the discrete version of the queuing network model for the airport, with the service time for each server being sampled from an empirical distribution.

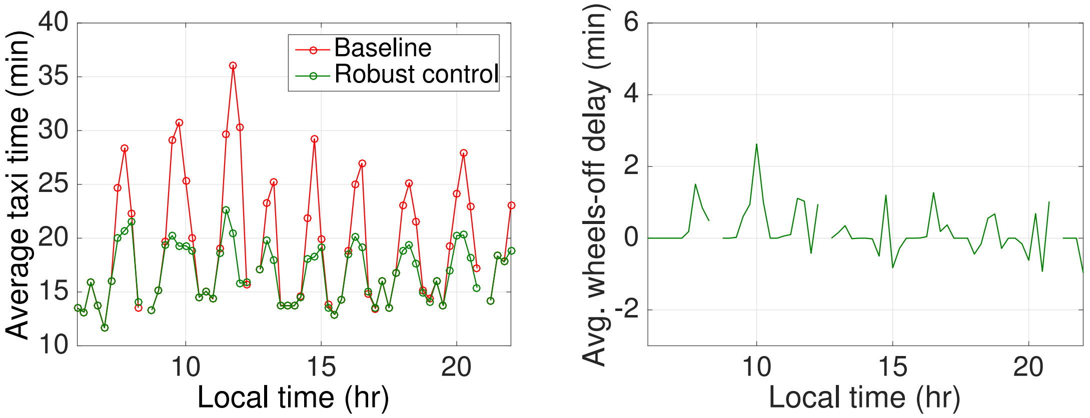
We apply the controller developed earlier to compute the release rate for departure flights to maintain a certain target queue length at the runway. Figs. 4-17(a)-4-17(b) show the mean queue length for the two runways over multiple realizations (10 samples) of the simulation obtained using the pushback control strategy and they are compared with the baseline case. The baseline case refers to the scenario without departure metering. We see that the resulting queue length with departure metering is close to the desired target value during periods of high demand (when there is a large queue in the baseline case). Note that the queue length is not close to the target value at all times because of the fact there is not enough demand. The schedule consists of banks of departures separated by banks of arrivals, a typical characteristic of hub airports such as CLT. The target queue length is set to 3.75 for each runway queue. The target queue length was set as the lowest value of the queue length that does not result in under-utilization of the runway. The runway is under-utilized if the take-off time for flights is delayed with departure metering compared to the baseline case. This occurs due to excessive gate-holds that is a result of low values of target-queue length. On the other hand, higher values of target queue length lead to decreased benefits in terms of taxi-out time reduction. The optimal target queue length is determined through a parametric analysis using simulations that ensures the average wheels-off delay over-all flights is close to zero and yields a high taxi-out time reduction. Figure 4-17(c) shows the average taxi-out time of flights with departure metering and they are compared to the baseline case. We can see that the peaks in the taxi-out time in the baseline case are reduced with departure metering. Moreover, the departure metering strategy appropriately holds the flights at the gate since it does not result in significant wheels-off delay as seen in Fig. 4-17(d). The wheels-off delay at some intervals is slightly negative because of the switch in the sequence of the order of take-off. Although the peaks in the wheels-off delay is about 1 to 2 min, the mean wheels-off delay is about 0.1 min, which we consider to be acceptable. One could obtain even lower values of wheels-off delay by appropriately increasing the target queue length, but this would come at the cost of reduced benefits in terms of taxi-out time reduction.

4.5.2 Comparison to other departure metering approaches

We compare the performance of the robust control approach with the ATD-2 logic and the optimal control approach (that were discussed in the previous chapter). Table. 4.1 shows



(a) Queue length of 36C (b) Queue length of 36R



(c) Average taxi-out time (d) Average wheels-off delay

Figure 4-17: Average queue length, taxi-out time and wheels-off delay from the simulations of the departure metering strategy for a typical day (May 7, 2015).

some key statistics obtained from the simulations comparing the two departure metering approaches over three days of operations (6 AM-9 PM local time) using a 20-min planning horizon and assuming perfect EOBT information. We notice that the reduction in taxi-out time is higher with the robust control approach compared to the other approaches and results in no significant wheels-off delay. Another distinguishing feature is that a smaller fraction of flights are held for a larger duration with the robust controller. Overall, the robust control approach performs better than the other two approaches.

Next, we consider the case when there is EOBT uncertainty. To account for the additional uncertainty, the target queue length is increased to four in order to avoid wheels-off delay.

Table. 4.2 shows a comparison of the departure metering approaches when there is EOBT uncertainty ($\sigma = 6$ min). Even in this case, we notice that the robust controller performs better and performance gap in terms of taxi-out time reduction is larger compared to the case without EOBT uncertainty.

Table 4.1: Comparison of simulations of departure metering approaches for CLT-NF (perfect EOBT case).

Mean statistics	**ATD-2 logic**	**Optimal control**	**Robust control**
Taxi-out reduction (min)	2.6	1.3	2.9
Hold time (min)	2.8	1.5	3.0
Wheels-off delay (min)	0.1	0.2	0.1
Fraction of flights held	0.6	0.3	0.4
Hold time of flights held (min)	4.3	4.5	8.4

Table 4.2: Comparison of simulations of departure metering approaches for CLT-NF (EOBT uncertainty, $\sigma = 6$ min).

Mean statistics	**ATD-2 logic**	**Optimal control**	**Robust control**
Taxi-out reduction (min)	0.9	1.2	2.2
Hold time (min)	1.0	1.3	2.3
Wheels-off delay (min)	0.0	0.1	0.1
Fraction of flights held	0.1	0.2	0.2
Hold time of flights held (min)	6.8	6.2	9.6

4.6 Discussions

In this section, we present a few extensions for the proposed approach.

4.6.1 Correcting for the prediction errors

The model uncertainties lead to tracking errors for time delay systems while using the predictor-based feedback controller, as we had seen earlier in Fig. 4-4 for tracking a single queue. We propose a heuristic to improve the tracking performance in the presence of model uncertainties. The idea is to work on the lines of the classical Smith Predictor that is used for stable linear time-delay system [2]. The Smith predictor provides feedback control not just using the predicted state but it also accounts for errors between the output and model predictions that might arise due to disturbances or model uncertainties. In a similar way, we account for the state prediction errors by estimating a correction factor online using past predictions and observations. Instead of using model predictions as inputs to the feedback law, we intend to use corrected predictions based on past observations. Let $\hat{P}_k$ be the corrected value of the state predictor, P_k, at time t_k (note a slight change in convention for the subscript, here k refers to the time index and we will illustrate the method for discrete version of the dynamics as used in the implementation of the controller). We assume the following form for $\hat{P}_k$.

$$\hat{P}_k = w_k P_k \tag{4.66}$$

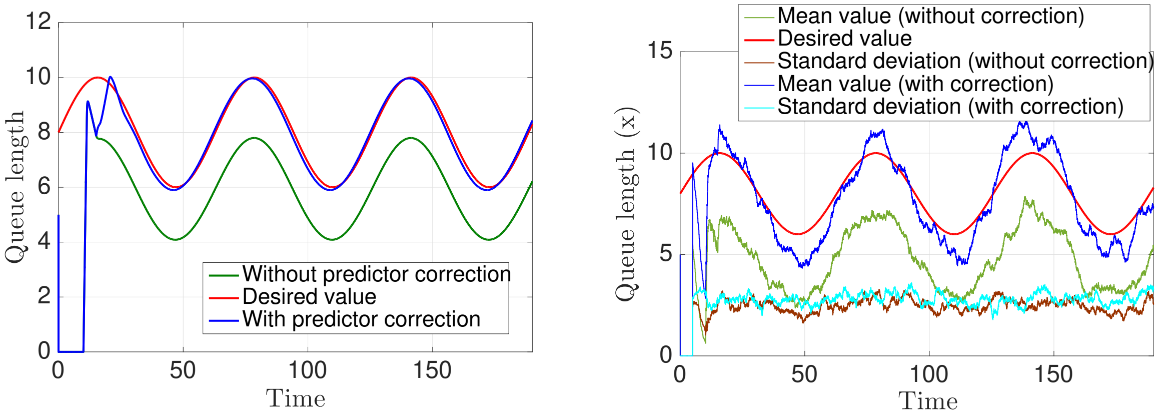
The correction factor, w_k, will be determined online using weighted recursive least squares based on past model predictions and current state observations. The recursive algorithm to obtain w_k is as follows,

$$K_k = Q_{k-1} P_k (\lambda_f + Q_{k-1} P_k^2)^{-1} \tag{4.67}$$

$$Q_k = (Q_{k-1} - K_k P_k Q_{k-1})/\lambda_f \tag{4.68}$$

$$w_k = w_{k-1} - K_k (w_k P_k - x_k) \tag{4.69}$$

Here, $\lambda_f \in (0, 1]$ is an exponential forgetting factor. A smaller value for λ corresponds to higher weighting for recent prediction errors compared to the past errors. Since the actual prediction error can be determined only after waiting for the system delay, the correction factor (w_k) that we determine is for the past prediction. However, we neglect that lag and assume that the correction factor determined is the best possible estimate for any given time. This method is effective for handling prediction errors that have a fixed bias or slowly varying. To illustrate the improvement in tracking performance, we consider queue length tracking for a single queue. The mean service rate of the of the actual dynamics is considered to be 1.5 times the service rate used in the model to design the controller. Fig. 4-18 shows the queue length obtained with a corrected estimate for the predictor and it is compared with the value obtained without the correction. The results are shown with the controller acting on the analytical model (Fig.4-18(a)) as wells as on the discrete queuing simulation (Fig.4-18(b)). We see significant improvements in the tracking performance compared to the case where naive state predictions were used (Fig. 4-4(c)). For the results presented, a higher value of the forgetting factor is chosen for the simulation ($\lambda_f = 1$) compared to the analytical model ($\lambda_f = 0.98$) to account for the stochasticity in the simulations. Another potential approach to deal with model uncertainty is to use a robust adaptive control approach to correct for the model parameters [162]. The challenge even with this approach is that one has to deal with delayed prediction errors.



(a) Analytical model

(b) Statistics from simulation

Figure 4-18: Correcting for the prediction errors using recursive least squares ($\mu = 1,\ \mu_a = 1.5,\ C = C_a = 1$).

4.6.2 Multiple arrival sources into the queueing network

Motivated by the congestion control on the airport surface, we discussed about controlling the sending rate from a single source into a queuing network. We could extend the analysis to the control of queuing networks with multiple sources as given in the general expression for the dynamics in Eq. 2.11. For example, in the tandem queue system that we had considered, instead of having a single source feeding customers into the first queue, we could have two sources feeding into each of the two queues. The model queuing dynamics for the non-delayed system would be of the form, $\dot{\mathbf{x}} = \bar{\alpha}(\mathbf{x}) + \mathbf{u}$, where, $\mathbf{x} = [x_1, x_2]^T$; $\mathbf{u} = [u_1, u_2]^T$ and $\alpha(\mathbf{x})$ as obtained from Eq. 2.11. We can compute the control inputs using the sliding mode approach to track the queue length for the two queues with the sliding variable, $\mathbf{s} = \mathbf{x} - \mathbf{x_d}$. Similar to the control input for a single queue, the control input for tracking would be of the form $\mathbf{u}(t) = \max(-\bar{\alpha}(\mathbf{x}) + \dot{\mathbf{x_d}} - \mathbf{k} \odot \operatorname{sgn}(\mathbf{s}), 0)$.

Chapter 5

Reachability analysis for queuing networks

5.1 Introduction

In the first part of this thesis, we presented a fluid flow model for queuing networks that resulted in an ordinary differential equation representation for the evolution of the ensemble mean queue length. The queuing model allowed us to use optimal and robust control techniques to regulate the queue length in a network. A closely related problem that is of interest is to analyze the performance of queuing networks.

The performance metrics, such as queue length and wait times, are generally obtained using stochastic simulations. However, simulations might not be computationally tractable for large queuing networks (as it involves sampling from a large joint distribution), particularly, if one wants to investigate the impact of uncertainty in network parameters such as the service rate of the servers. Instead, we propose reachability analysis as an alternative tool to analyse performance of large-scale uncertain queuing networks. The approach bears some philosophical similarity to the recent robust optimization approach to analyze the performance of queuing networks [28]. We note that both methods: (a) take a non-probabilistic approach by specifying the underlying randomness as uncertainty sets, and (b) determine the queuing performance measures that correspond to the extreme case scenarios. However, in contrast to the the robust optimization approach, our approach relies on control-theoretic techniques for continuous dynamical systems, and the underlying queuing model is different.

In this chapter, we present a framework to compute the range of the expected queue length in a network given the level (range) of uncertainty in the queue parameters. We utilize the queuing network model developed earlier and some recent results in reachability analysis for non-linear systems to realize this objective [48, 129, 128]. Reachability analysis is a technique that has been used in formal controller verification and model checking. To the best of our knowledge, this is the first attempt in employing reachability analysis for queuing networks.

5.2 Reachability analysis

Reachability analysis involves computing the sets of future states achievable by a system given an uncertain set of initial states and input parameters. To give an intuitive explanation, the reachable sets obtained from the analysis is equivalent to performing multiple simulations by exhaustively sampling the uncertainty set. For the queuing network example, the set of states would correspond to the length of queues in the network, and the sources of uncertainty include server service rates, arrival rates and initial queue length. To formalize the notion of reachable sets, consider the following continuous dynamical system,

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{p}) \tag{5.1}$$

here, $\mathbf{x} \in \mathbb{R}^n$ represents the state vector of the system, $\mathbf{p} \in \mathbb{R}^q$ denotes the vector of uncertain parameters (that could represent either external disturbances or control inputs), and $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^q \to \mathbb{R}^n$ is a continuously differentiable vector field. The state reached by the above system (5.1) at time t with initial state $\mathbf{x}_0$ and parameter $\mathbf{p}$ is denoted by $\mathbf{\Phi}(t; t_0, \mathbf{x}_0, \mathbf{p})$. Let $\mathbf{X}_0$ and $\mathbf{P}$ represent the set of initial states and parameters, respectively.

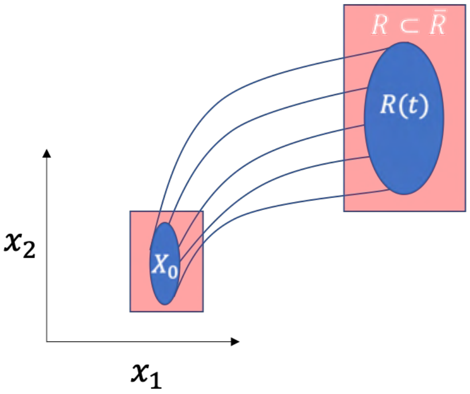


Figure 5-1: Illustration of reachable sets

Then the reachable set of (5.1) at time t is given by,

$$R(t; t_0, \mathbf{x}_0, \mathbf{p}) = \{\mathbf{\Phi}(t; t_0, \mathbf{x}_0, \mathbf{p}) | \mathbf{x}_0 \in \mathbf{X}_0, \mathbf{p} \in \mathbf{P}\}. \tag{5.2}$$

Computing the exact reachable set ($R(t)$) is often challenging, instead, one is interested in a tight over-approximation to the reachable set ($\bar{R}$), such that $R \subseteq \bar{R}$. Figure 5-1 shows an illustration for reachability analysis in a 2D space with the actual reach set and the over-approximation represented by blue ellipse and orange rectangle, respectively. Different methods have been proposed that vary by complexity and conservativeness to the over-approximation. They include level set methods, interval analysis, sampling-based methods [14, 64], contraction analysis and Taylor methods [123, 8]. A key aspect of the reachable set computation is the set representation. The set representation should scale well with dimension of the state space and one must be able to perform standard mathematical operations (such as a linear transformation) in a computationally tractable manner. Commonly used set representations include zonotopes, polytopes, high dimensional intervals and ellipsoids. Multiple software toolboxes are available for reachability computation: CORA [7], C2E2 [65], TIRA [128], etc. CORA is primarily based on Taylor approximation to the dynamics, with the reach-set being represented as a zonotope. TIRA offers different methods that are based on contraction analysis and mixed-monotonicity, and high-dimensional intervals for the set representation. In our work, we use a fairly recent technique known as

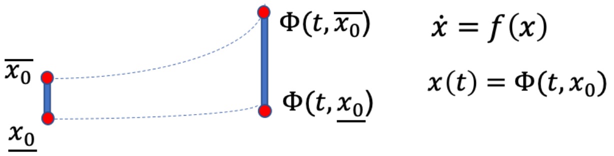


Figure 5-2: Illustration for computing the interval reach set for monotonic systems

mixed-monotonicity method for reachability analysis [47]. The primary reason for using this method is because it offers a good trade-off between computational tractability and extent of over-approximation for non-linear systems such as the one considered in this thesis.

5.3 Mixed-monotonicity method

If a system is monotonic, then computing the interval reach sets is fairly straight forward: the reach sets are obtained by propagating the lower bound and upper bound of the interval set forward in time by numerically integrating the system dynamics (as illustrated in Figure 5-2). The key idea behind the mixed monotonicity approach is to decompose a non-monotone system into a mixed monotone system (loosely, monotonically increasing in one of the function arguments and decreasing in the other function argument). To clarify this idea with more rigor, we first present some preliminary definitions and then provide a brief overview of the mixed monotonicity approach that is based on an earlier work [188].

Definition 8 *(Monotone mapping) A mapping $h : \mathcal{X} \rightarrow \mathcal{T}$ is said to be monotone if it is order preserving,*

$$\mathbf{x}, \mathbf{y} \in \mathcal{X} : \mathbf{x} \geq \mathbf{y} \implies h(\mathbf{x}) \geq h(\mathbf{y}). \tag{5.3}$$

Definition 9 *(Mixed monotone mapping) A mapping $h : \mathcal{X} \rightarrow \mathcal{T}$ is mixed monotone if there exists a decomposition function, $g : \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{T}$, satisfying the following properties: (a) $g(x,x) = h(x)$; (b) g is monotone increasing in the first argument, $x_1 \geq x_2 \implies g(x_1, y) \geq g(x_2, y)$; (c) g is monotone decreasing in the second argument, $y_1 \geq y_2 \implies g(x, y_1) \leq g(x, y_2)$.*

Remarks: (1) The decomposition function, g, may not be unique. (2) A system such as (5.1) is mixed monotone if the flow map $\Phi(t; t_0, \mathbf{x}_0, \mathbf{p})$ that maps the initial states to the final state at time t is mixed monotone. (3) Mixed monotonicity is a very generic property; a sufficient condition for mixed monotonicity requires the Jacobian of the function to be finite everywhere in the domain [188].

We follow the reachability set computation as presented in [129]. Let $c \in \{\mathbf{x}, \mathbf{p}\}$ represent a generic variable and a dimension $m \in \{n, q\}$ such that $c \in \mathbb{R}^m$. Assume that the elements of the Jacobian matrix are bounded as follows:

$$\frac{\partial f_i}{\partial c_j}(\mathbf{x}, \mathbf{p}) \in [a^c_{ij}, b^c_{ij}],\ i \in \{1, 2..n\},\ j \in \{1, 2..m\}. \tag{5.4}$$

The values of these bounds lead to the following four cases with respect to the signs of the partial derivatives: (C1) *positive*: $a^c_{ij} \geq 0$; (C2) *mostly positive*: $a^c_{ij} \leq 0 \leq b^c_{ij}$ and $|a^c_{ij}| \leq |b_{ij}|$; (C3) *mostly negative*: $a^c_{ij} \leq 0 \leq b^c_{ij}$ and $|a^c_{ij}| \geq |b_{ij}|$; (C4) *negative*: $b^c_{ij} \leq 0$. Since we have assumed that the elements of the Jacobian matrix are bounded, the function dynamics, f, is mixed monotone [188]. Let g be the decomposition function for f, which is defined as follows:

$$g_i(\mathbf{x}, \mathbf{p}, \mathbf{x}^*, \mathbf{p}^*) = f_i(X_i, P_i) + \alpha^x_i(\mathbf{x} - \mathbf{x}^*) + \alpha^p_i(\mathbf{p} - \mathbf{p}^*) \tag{5.5}$$

where $X_i = [x_{i1}..x_{in}]^T$, $P_i = [p_{i1}..p_{iq}]^T$ and $\alpha^c_i = [\alpha^c_{i1}, .., \alpha^c_{im}]$ are defined according to cases (C1)-(C4) as follows (note $c \in \{\mathbf{x}, \mathbf{p}\}$):

$$c_{ij} = \begin{cases} c_j \text{ if (C1) or (C4),} \\ c^*_j \text{ otherwise.} \end{cases} \qquad \alpha^c_{ij} = \begin{cases} -a^c_{ij} \text{ if (C2),} \\ b^c_{ij} \text{ if (C3),} \\ 0 \text{ otherwise.} \end{cases} \tag{5.6}$$

The above definition satisfies the condition for g to be a decomposition function since: (a) $g_i(\mathbf{x}, \mathbf{p}, \mathbf{x}, \mathbf{p}) = f_i(\mathbf{x}, \mathbf{p})$, (b) $\frac{\partial g_i}{\partial c_i} \geq 0$, (c) $\frac{\partial g_i}{\partial c^*_i} \leq 0$, $c \in \{\mathbf{x}, \mathbf{p}\}$. Next, define an 'embedded' dynamical system as follows:

$$\begin{pmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{x}}^* \end{pmatrix} = \mathbf{k}(\mathbf{x}, \mathbf{p}, \mathbf{x}^*, \mathbf{p}^*) = \begin{pmatrix} g(\mathbf{x}, \mathbf{p}, \mathbf{x}^*, \mathbf{p}^*) \\ g(\mathbf{x}^*, \mathbf{p}^*, \mathbf{x}, \mathbf{p}) \end{pmatrix} \tag{5.7}$$

Let $\Phi_h(t; t_0, \mathbf{x}_0, \mathbf{p}, \mathbf{x}_0^*, \mathbf{p}^*)$ be the flow map of the embedded system that maps the initial states to the final states at time, t. Let Φ_{h1} and Φ_{h2} denote the first n and last n components of Φ_h, respectively. Then the following theorem provides a way to compute a tight over-approximation interval for the reachable set at time t.

Theorem 5 *(Computing reachable sets [129]) For all bounds $\overline{x}, \underline{x} \in \mathbb{R}^n$, $\overline{p}, \underline{p} \in \mathbb{R}^q$ and for all $t \in \mathbb{R}^+$, $\mathbf{x}_0 \in [\underline{x}, \overline{x}]$, $\mathbf{p} \in [\underline{p}, \overline{p}]$, we have*

$$\Phi_{h1}(t; t_0, \underline{x}, \underline{p}, \overline{x}, \overline{p}) \leq \Phi(t; t_0, \mathbf{x}_0, \mathbf{p}) \leq \Phi_{h2}(t; t_0, \overline{x}, \overline{p}, \underline{x}, \underline{p}) \tag{5.8}$$

Note that Φ_{hi} can be obtained by a simple numerical integration of the system dynamics (Eq. (5.7)) forward in time. We have used TIRA [128], a MATLAB library for reachability analysis, for computing the reachable sets in our work.

5.4 Application for queuing networks

To illustrate reachability analysis for queueing networks, we first consider a simple queuing system, the $M/M/1$ queue. Let λ be the mean arrival rate into the queue and μ be the mean service rate of the server. Using our proposed fluid-flow queuing model, an approximation for the dynamics of the ensemble mean queue length is given by,

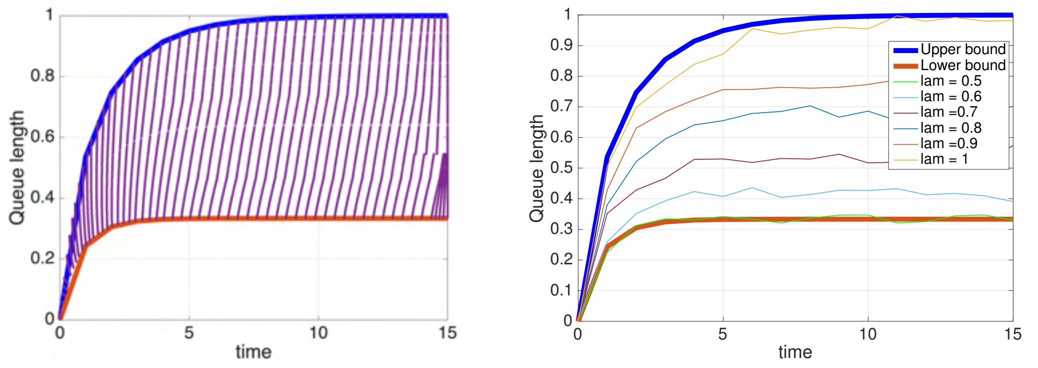
$$\dot{x} = -\mu \frac{x}{x+1} + \lambda, \quad x_0 \geq 0 \tag{5.9}$$

The goal is to compute the bounds of the expected queue length at any particular time given an uncertainty associated with the arrival rate, λ. Note that although the solution to this problem is trivial, we use this example to illustrate the methodology. The Jacobian of the dynamics is given by,

$$f_x = \frac{-\mu}{(x+1)^2}, f_\lambda = 1; \tag{5.10}$$

$$\text{sgn}(f_x) = -1; \quad \text{sgn}(f_\lambda) = 1 \tag{5.11}$$

The Jacobian is always finite over the entire domain, and hence the system is mixed-monotone [188]. Therefore, we could use Theorem 5 for computing the reachable sets. Figure 5-3(a) shows the bounds on the expected queue length obtained from the reachability computation for a particular set of the system parameters ($\mu = 2$ and $\lambda \in [0.5, 1]$) and they are compared with simulations of the analytical model. The simulations of the analytical model involve integrating the dynamics forward in time for various values of the arrival rate sampled from the interval uncertainty set. We notice that the reachable set from the analysis tightly encloses the trajectories obtained from the simulations. Figure 5-3(b) shows a comparison of the reachable sets along with the trajectories (ensemble mean queue length) obtained from discrete queuing simulations (that represents the actual queuing system instead of the analytical model). Even here we see that the trajectories are tightly enclosed within the bounds generated by the reachable set computation.



(a) Simulation samples of the analytical model compared with the reachable set

(b) Expected value obtained from discrete queuing simulations for various (λ)

Figure 5-3: Reachable set for the expected queue length for M/M/1 queue with uncertain mean arrival rate (λ).

Next, we consider a more complex queuing network topology. For the sake of illustration, we consider the queuing network representation for Charlotte airport (CLT) departure process — a single queue connected to two parallel queues, with the first queue serving two classes of customers depending on which parallel queue the customer joins (as shown earlier in Fig. 4-15(b)). Assuming the service time distributions are exponential and arrivals are Poisson, the queuing network model is given by,

$$x_T(t) = x_1(t) + x_2(t) \tag{5.12}$$

$$\dot{x}_i(t) = -\mu \frac{x_i(t)}{1 + x_T(t)} + \lambda_i, \quad i = 1, 2; \tag{5.13}$$

$$\dot{x}_i(t) = -\mu_i(t) \frac{x_i(t)}{1 + x_i(t)} + \mu \frac{x_{i-2}(t)}{1 + x_T(t)} \quad i = 3, 4; \tag{5.14}$$

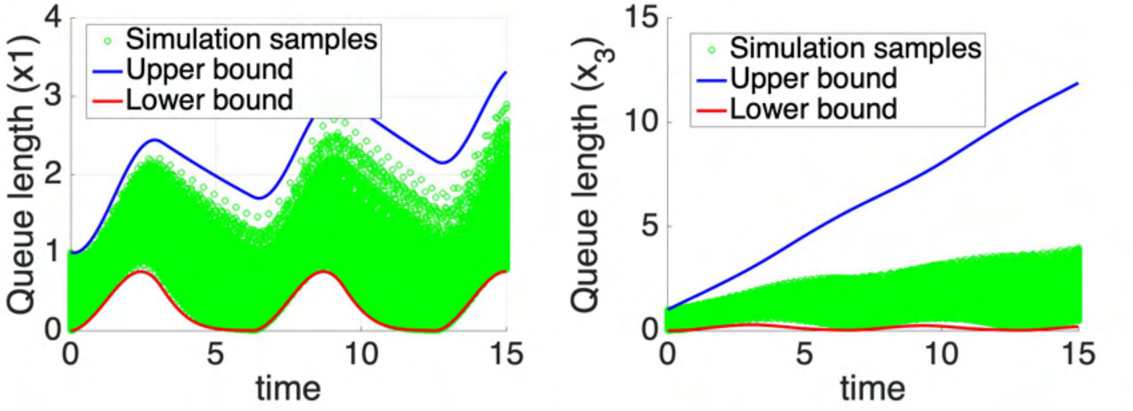
Note that the assumptions on the service time and arrival time distributions were made for the ease of analysis, and one could have the dynamics with general service time distribution as used earlier in Chapter 2. Figure 5-4 shows a comparison of the reachable sets and the trajectories obtained from the simulations of the analytical model by sampling the uncertainty set. For these computations, the uncertainty in the initial queue length ($x_i(0)$) and the network parameters ($\mu_i,\ \lambda_i$) are considered to be as follows:

$$x_i(0) \in [0, 1], i = 1, 2, 3, 4 \tag{5.15}$$

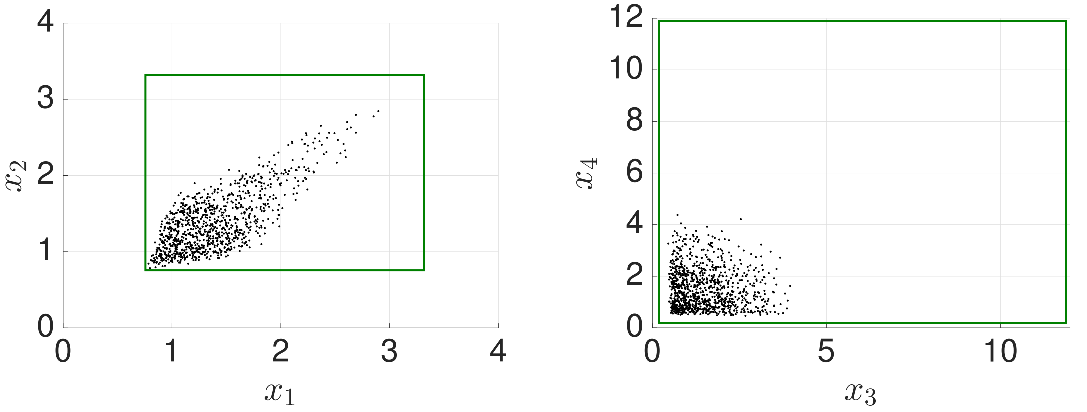
$$\mu \in [1.8, 2.2];\ \mu_3 \in [0.8, 1.2],\ \mu_4 \in [0.8, 1.2], \tag{5.16}$$

$$\lambda_i \in [0, 0.5] + \max(\sin(t), 0), i = 1, 2 \tag{5.17}$$

The computational time for determining the reachable set at the final time, $t_f = 15$, was found to be just 0.51 s for this case. Figure 5-4 shows the time series and a time-slice of the reachable sets representing the queue lengths. We notice that the over-approximation of the reachable set becomes conservative, particularly for longer time horizons and for downstream queues (x_3, x_4). The primary reason for such conservative over-approximation is because of wrapping effect. This arises when non-interval sets are over-approximated using interval enclosures as illustrated in Fig. 5-5. Some of the ways to reduce conservativeness is to use a better set representation for the reachable set (such as ellipsoids or zonotopes) or use a different method for analysis such as Taylor models [7]. However, these methods significantly increase computational time that does not scale well with increase in state space dimension.



(a) Reachable set for $x_1(t)$ (b) Reachable set for $x_3(t)$



(c) Reachable set in $x_1 - x_2$ plane at $t = t_f = 15$ (d) Reachable set in $x_3 - x_4$ plane at $t = t_f = 15$

Figure 5-4: The reachable set for the mean queue length in the network.

5.5 Partitioning the initial set

A straight forward approach to obtain a less conservative outer-approximation is to partition the initial uncertainty set before computing the reachable sets. The set partitioning should be such that the union of the partitioned sets should be equal to the original set. In the analysis, reachable sets are computed for each of the partitioned sets and their union gives

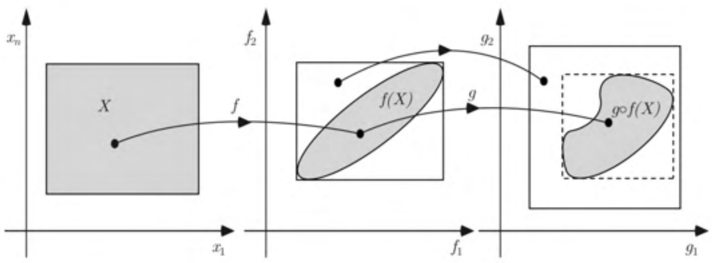
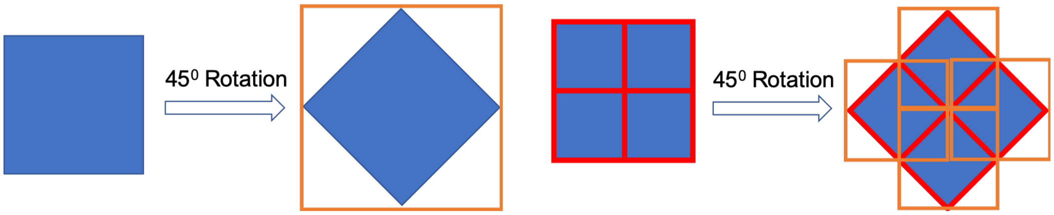


Figure 5-5: Illustration of wrapping effect: the resulting interval enclosure of the transformed interval sets can be significantly conservative (source: [7])

the reachable set for the complete initial uncertainty set. An illustration of the benefits of set partitioning is shown in Fig. 5-6. In this example, the initial uncertainty set undergoes a rotational transformation. We can see that the interval enclosure using the partitioned set gives a tighter over-approximation for the reachable set.

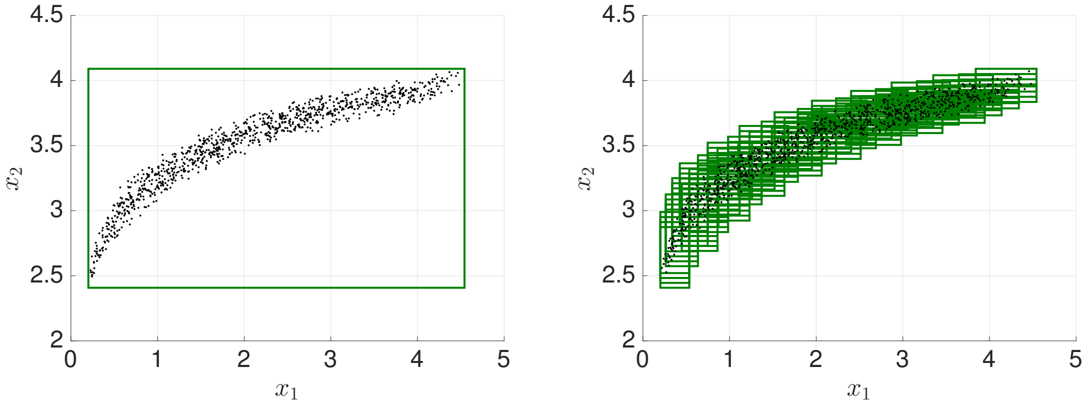
We also show the benefits of set partitioning for computing the reachable sets of a queuing system. Consider two servers with exponential service time distributions in series, and with the following parameters: mean service rate, $\mu_1 = 1.1, \mu_2 = 2$; arrival rate $\lambda \in [0, 1]$; initial queue length $x_{01} = [3, 4], x_{02} = [3.4, 3.6]$. Figure 5-7 shows the reachable set obtained with and without set partitioning. For the set partitioning case, the initial set was divided into 144 parts (4 equal parts for the states and 9 equal parts for the control input). We can see that set partitioning gives a tighter outer-approximation for the reachable set. In terms of computational time, the set partitioned case takes 1.75 s, whereas the case without set partitioning takes 0.08 s. An extreme case of set partitioning is simulation by "exhaustive" sampling of the uncertainty set. One should also note that naive set partitioning does not scale well with increase in dimension of the state space. This motivates the question on how we can intelligently partition the state space in a dynamic manner so as to avoid over-approximation while keeping the computations tractable.



(a) Interval enclosure using complete set

(b) Interval enclosure after partitioning initial set

Figure 5-6: Comparing interval enclosure with and without partitioning the initial set.



(a) No partitioning

(b) Partitioning initial set

Figure 5-7: Reachable set for tandem queue system at $t = 5$.

Chapter 6

Data analytics for airport operations: Beyond queuing models

6.1 Introduction

One of the earliest academic papers on the demand-capacity imbalance at airports and the resulting flight delays was published way back in the 1940s [33]. Although the analyses in those early papers were elegant and highlighted the underlying principles, the framework relied on theoretical distributions of the key system parameters (such as runway separation times) due to the lack of operational data, limiting the accuracy of those models. However, the research landscape has dramatically changed in recent years, where researchers have access to multiple sources of operational data. One can proclaim that we have entered the era of "big data" even in aviation wherein large amounts of data are collected from a variety of sources including sensors on the aircraft as well as on the ground. A good review of the various aviation data sources can be found in [112]. These data sources play a critical role in developing accurate models and practical decision support tools.

In the first part of this thesis, we presented airport-specific queuing models that were developed using flight trajectories and flight schedules. In addition to flight trajectories and schedules, there are other sources of operational data that one could exploit to characterize operations and develop decision support tools. In this chapter, we present models to analyse airport operations that were developed using other emerging sources of aviation data with two illustrative examples. First, we present a statistical model to estimate aircraft fuel burn in the taxi phase using flight trajectories, which is intended for environmental monitoring of airport operations. Next, we present an approach to extract operational information such as flight ID and runway number from air traffic controller voice communications, which could potentially be used in many applications such as improving system safety and enhancing air traffic controller training. Further, information such as call-ready time is available primarily through ATC voice data. Therefore, models to automatically transcribe ATC voice data and extract such critical operational information will help characterize operations as well as develop decision support tools for traffic management (such as departure metering).

6.2 Environmental impact assessment of airport operations

Aviation accounts for nearly 12% of all the emissions from transport related sources [3]. Emissions of carbon monoxide, unburned hydrocarbons and particulate matter are higher at the surface level where aircraft taxi with low engine thrust settings [125]. Due to the significant impact of fuel burn, and the resulting emissions on our health and environment, there is a growing interest among regulatory agencies to monitor them at major airports. Although aircraft fuel burn is archived in flight data recorders (FDR, commonly referred to as the 'black-box'), airlines do not share this data with regulatory agencies because of various privacy concerns. Therefore, policy makers need good estimates of fuel burn and emissions to set reasonable targets, which has motivated research into modeling these quantities [40, 16].

Industry standard models such as FAA's Aviation Environment Development Toolbox (AEDT) use taxi-time as the main input to estimate fuel burn [66]. However, such simple models do not consider the second order effects such as spikes in the aircraft fuel consumption during stop-and-go traffic, which is becoming more prominent with increasing congestion at airports. This motivates the need to develop more granular and accurate aircraft fuel burn models using flight trajectories.

Aircraft trajectories on the airport surface can be obtained primarily from two sources: aircraft's Flight Data Recorder (FDR) and airport surface surveillance systems. A trajectory here is specified by latitude, longitude, heading and velocity with an associated time stamp. The data from FDR includes the aircraft trajectory as well as other critical flight parameters such as engine fuel flow rate, airspeed, gross weight, ambient temperature and pressure. The trajectory information from the FDR data has been used earlier to develop fuel burn models for various phases of flight, and their predictive performance has been shown to be better than some industry standard tools [153, 100, 40].

Unlike the FDR data, the trajectory information from the airport surface surveillance system is easier to access, particularly for regulatory agencies such as the FAA that operate them. Note that the trajectory information in the FDR data is based on an onboard GPS, whereas the ASDE-X data is a fusion from multiple external sensors such as the airport ground radars, multilateration sensors and Automatic Dependent Surveillance — Broadcast (ADS-B) sensors. The inputs from multiple external sensors make the ASDE-X data more noisy when compared to the FDR data. The other major drawback of the ASDE-X is its poor coverage around the airport ramp areas.

Even though there are a couple of drawbacks with the ASDE-X data, there is significant interest among regulatory agencies in utilizing archives of ASDE-X data for estimating fuel burn and emissions because of the accessibility to this dataset. Another important aspect to note is that the trajectory-based fuel burn models that have been developed earlier [100] using FDR data have not been tested using ASDE-X trajectories because of the lack of availability of ASDE-X trajectories corresponding to the FDR data (that contain the fuel burn information). In this section, we develop a new statistical model to estimate aircraft fuel burn using flight trajectories and present a framework that ensures that the models work well even with noisy ASDE-X trajectories.

6.2.1 Aircraft trajectory and fuel burn

The FDR data presented in this section was obtained from a major European airline. The dataset contains multiple aircraft types operating from different airports around the world [40]. We first discuss the modeling methodology using data from A330-343 aircraft, and then present the results using the same procedure for all the aircraft types in our dataset. The A330 dataset consists of 183 departures from seven major US airports (BOS, EWR, JFK, LAX, MIA, ORD, SFO), and we focus our attention on the taxi-out phase of flight.

Figure 6-1 shows the time-series of speed and fuel flow rate during the taxi-out phase for a sample flight. We notice that the fuel flow rate is constant for most periods and spikes during some intervals. On an average over 183 A330 flights in our dataset, the fuel flow rate spikes during 4.5% of the entire duration, and the spikes correspond to nearly 3.5% of the total fuel burn during the taxi-out phase. Notice that the intervals of spikes in the fuel flow rate correspond to the periods of acceleration. Moreover, the height and duration of these spikes vary, and they correlate well with the duration and value of acceleration. The constant fuel flow rate value (also referred to as the baseline fuel flow rate) corresponds to the idle throttle setting of the engines, whereas spikes occur whenever the throttle setting is increased, resulting in an acceleration. The aircraft could also accelerate without increasing the throttle when the pilots release the brake from a stop. However, the magnitude of acceleration during brake release or turns is smaller when compared to the acceleration obtained by increasing the throttle. The other unlikely scenario to note is when the pilots brake while increasing the throttle, resulting in a spike in the fuel flow rate without any acceleration.

The total fuel burn during the taxi-out phase of a flight can be split into a contribution from the baseline fuel flow rate component and a contribution from the spikes. The constant fuel flow rate component depends on the total taxi-out time of the flight. The spikes depend on the acceleration, which needs to be estimated from the trajectory information. The idea is to estimate the intervals of acceleration from the trajectory and use that to develop a model for the total fuel burn. Note that the trajectory data contains speed at a low resolution (0.41 m/s), and therefore numerical differentiation of the speed to estimate acceleration does not yield good results. A simple moving average smoothing of the speed measurements, followed by numerical differentiation is also not beneficial because it will under-estimate the acceleration. To overcome these issues, some of the trajectory-based fuel burn models proposed earlier use the number of stops as a surrogate for acceleration events [100]. Instead, in our work, we propose using a multi-model smoothing algorithm for detecting the periods of acceleration, and use that as the input feature in our models to obtain more accurate estimates of the total fuel burn. Moreover, the trajectory data from the ASDE-X is noisy when compared to the FDR data, making it challenging to estimate the periods of acceleration. However, we show that our models perform well even with ASDE-X trajectories by tuning some of the parameters in the smoothing algorithm to preserve the correlation between the fuel burn spikes and acceleration.

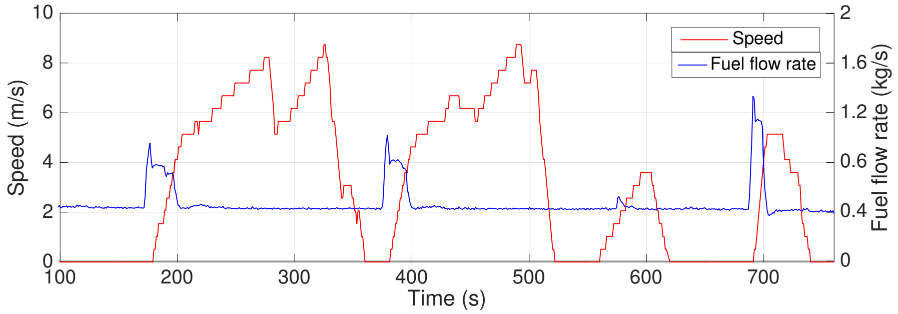


Figure 6-1: A sample profile of aircraft speed and total fuel flow rate during the taxi-out phase of flight for A330-343.

6.2.2 Detecting acceleration events

A multi-model smoothing algorithm is used to determine the acceleration events from the raw trajectory measurements. The smoothing algorithm is based on combining estimates produced by an Interacting Multiple Model (IMM) filter with a backward-time recursion [116]. The details of the smoothing algorithm are presented in Appendix B. In addition to smoothed estimates of the state variables (trajectory), the smoothing algorithm outputs the 'most likely mode' of the system at any time instant. We consider three modes to represent the aircraft dynamics: (a) constant speed mode ($a = 0, \omega = 0$), (b) acceleration mode, $(a = a_t, \omega = 0)$, and (c) deceleration mode, $(a = -a_t, \omega = 0)$. Here a_t is acceleration threshold parameter that needs to be chosen and ω is the turn rate. The modes were motivated by the fact that the fuel burn spikes correspond to the periods of acceleration, whereas turns do not have any significant impact [100]. One could consider additional modes such as turns or accelerated turns [99], however, it was decided to keep the number of modes to a minimum to obtain a robust predictor for the fuel burn spikes.

The first plot in Fig. 6-2 shows the estimated speed profile for the flight trajectory along with the raw data. The second plot in the same figure shows the normalized likelihood or mode fraction of the three modes obtained from the smoothing algorithm, which represents the probability of the aircraft being in one of the three modes considered in our analysis (constant speed, acceleration or deceleration). We notice that periods of acceleration correspond to a higher mode fraction for the acceleration event. Every instant of the trajectory is classified into one of the three modes (constant speed, acceleration or deceleration) based on the maximum value of the mode fraction. The third plot of Fig. 6-2 shows the periods in the acceleration mode (indicated by 1), along with the total fuel flow rate into the engines. We define an acceleration event as an extended period (at least 4 s) of the dynamics being in the acceleration mode. Similarly, a fuel burn event corresponds to an extended period of a spike in the fuel flow rate. We notice that the acceleration events correspond to fuel burn events with a small time lag. The lag corresponds to the response time for the thrust to increase with the fuel intake for a jet engine.

The acceleration threshold (a_t) has a significant impact on the mode detection. A high value of the threshold might not capture any acceleration event, whereas a low value would detect small accelerations or even falsely report accelerations. For our application to predict fuel burn, the threshold needs to be tuned such that the reported acceleration events correspond to the fuel burn events. For example, in Fig. 6-2, although there is a slight acceleration between 300 and 350 s, it does not correspond to a fuel burn spike. Such variations in speed could be because of turns or braking that do not correspond to throttling (a higher fuel flow rate). Therefore, we choose an optimal acceleration threshold, $a_t = 0.4\ ms^{-2}$, which ensures that a slight increase in speed is not flagged as an acceleration event and the value is

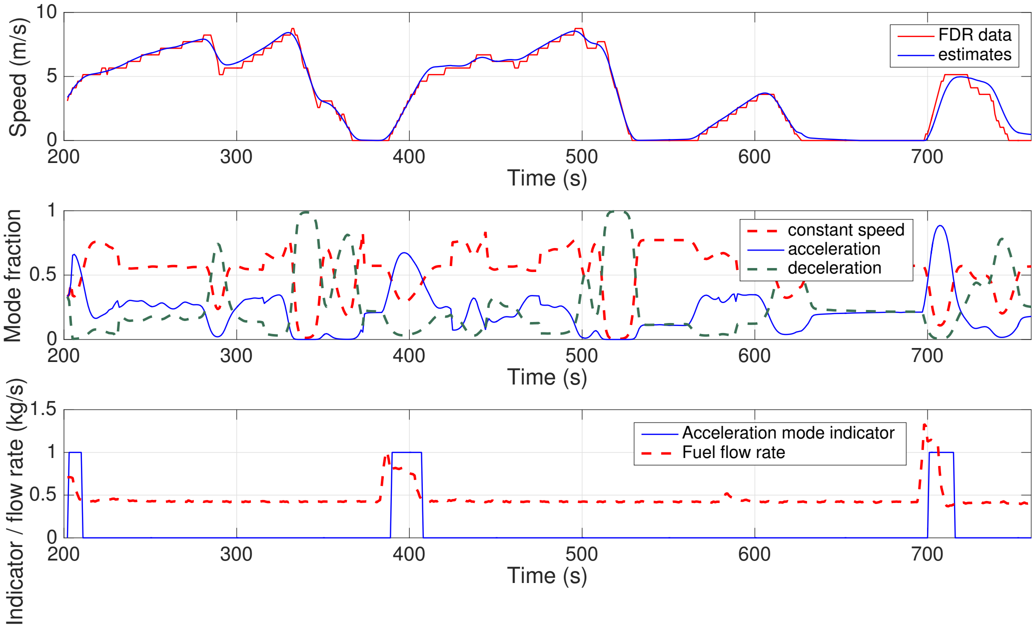


Figure 6-2: The plot on the top shows a comparison of the speed profile obtained from the smoothed estimate with the actual FDR data. The middle plot shows the normalized likelihood (mode fraction) for the differet modes. The bottom plot shows the indicator function for the acceleration event obtained from the mode fraction along with the total fuel flow rate.

sufficiently large to capture the fuel burn spikes correctly. The details on the methodology to choose the optimal acceleration threshold is skipped here and presented in Appendix B. At the optimal threshold of 0.4 ms^{-2}, 82% of the total duration of acceleration in our dataset is mapped to fuel burn events and 78% of the total duration of fuel burn spike is mapped to the acceleration events.

6.2.3 Fuel burn prediction using aircraft trajectories

In this section, we present three fuel burn models, each with a different set of predictors, to determine the total fuel burn during taxi operations. For the sake of discussion, we call the three models: Model A, Model B and Model C. The model equations are as follows:

- Model-A: $f/\sqrt{T_{amb}} = c_1 t$.
- Model-B: $f/\sqrt{T_{amb}} = c_2 t + d_2 n_a$.

- Model-C: $f/\sqrt{T_{amb}} = c_3 t + d_3 t_a$.

Here, f denotes the total fuel consumption for an aircraft during its taxi-out phase, T_{amb} is the ambient temperature, t is the taxi-out time, n_a is the number of acceleration events, t_a is the total duration of acceleration and $(c_i,\ d_i)$ are constants that are determined using least squares regression. The reason for not including the intercept term in the linear equations is to ensure that the models are physically interpretable in a sense that the total fuel burn should be zero when the taxi-out time is zero. Note that the fuel burn models are for the taxi-out phase after pushback, and therefore the fuel flow rate consists of just the baseline fuel flow rate with intermittent spikes, without any added component at pushback. In all the models, the total fuel burn is normalized with $\sqrt{T_{amb}}$ because the engine specific fuel consumption is proportional to this factor [100]. The taxi-out time is a natural choice as a predictor because the total fuel burn depends on the time spent by the aircraft on the airport surface with its engines turned on. The taxi-out time is considered to be the primary explanatory variable to account for the constant fuel flow rate component, which is the key contributor. To capture the fuel burn spikes, we consider the number of acceleration events in Model-B and the total duration of acceleration in Model-C. In the past, researchers have investigated using a linear model with the number of stops as a predictor (a surrogate for the number of acceleration events), in addition to the taxi-out time [100]. In this work, we investigate the benefits of using the duration of acceleration as the predictor (in Model-C) instead of using just the number of acceleration events. The primary reason for not considering aircraft gross weight as another explanatory variable in these models is because there have been studies that have shown that the idle thrust (and hence the fuel flow rate) does not vary significantly with aircraft mass [176]. Further, we wish to use predictors that are accessible via openly available data sources, which is the other reason for not including the aircraft mass as predictor because it is available only through the FDR data.

The models were trained using 60% of the sample points in the dataset and the rest were used for testing. The number and duration of acceleration events is determined using the optimal acceleration threshold in the smoothing algorithm. The predictive performance of the three models are compared using the mean percentage error (MPE) and mean absolute percentage error (MAPE), with the error being calculated by subtracting the actual value from the model prediction for each individual flight. Table 6.1 shows the model equations, p-values of the regression coefficients and error statistics for different aircraft types. The MAPE values indicate that Model-C performs the best compared to the other approaches, followed by Model-B, which is better than Model-A. The predictive performance is better for models with higher fidelity as one would expect (in a sense that the duration of acceleration has more information than the number of acceleration events). The relative improvement of the model varies for different aircraft types. The benefits of the higher fidelity model, in terms of accuracy, depends largely on the fraction of the fuel burn spike duration in the dataset.

Table 6.1: Model equation for the fuel burn and the error statistics for different aircraft types. The number of data points in the test set is shown below the aircraft type in brackets. The numbers below the model equation indicate the p-value of the model coefficients, and the row below shows (R^2, MPE, MAPE).

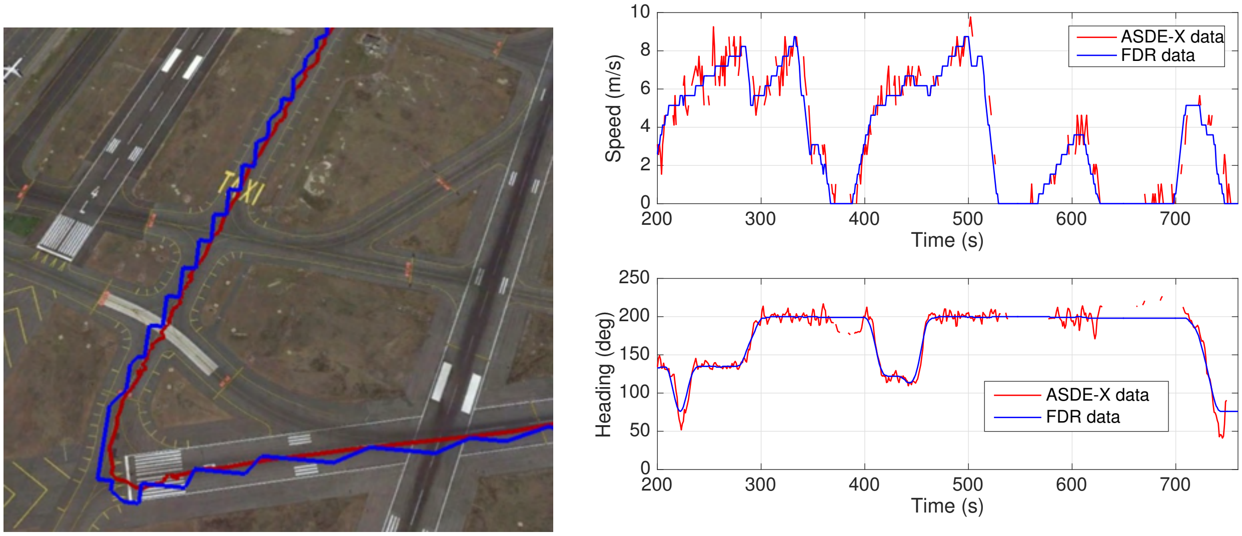
Aircraft (engine type)	Model-A	Model-B	Model-C
A330-343 (Trent 772B-60) (84)	$0.0259\ t$ (0.0) (0.9962, 1.49, 3.29)	$0.0252\ t + 0.3778\ n_a$ (0.0) (0.0) (0.9980, 0.05, 2.63)	$0.0253\ t + 0.0298\ t_a$ (0.0) (0.0) (0.9981, -0.14, 2.07)
A320-214 (CFM56-5B4/2) (68)	$0.0120\ t$ (0.0) (0.9598, 0.73, 6.98)	$0.0116\ t + 0.0759\ n_a$ (0.0) (0.0394) (0.9614, 0.94, 6.72)	$0.0115\ t + 0.0073\ t_a$ (0.0) (0.0006) (0.9642, 0.77,6.44)
A321-111 (CFM56-5B4/2) (29)	$0.0120\ t$ (0.0) (0.9600, -1.16, 5.01)	$0.0112\ t + 0.1663\ n_a$ (0.0) (0.0001) (0.9701, 0.13, 4.45)	$0.0113\ t + 0.0084\ t_a$ (0.0) (0.0005) (0.9703, -1.34, 4.10)
B777-300ER (GE90-115BL) (51)	$0.0349\ t$ (0.0) (0.9975, 1.42, 2.14)	$0.0346\ t + 0.0848\ n_a$ (0.0) (0.0416) (0.9977, 1.27, 1.94)	$0.0346\ t + 0.0135\ t_a$ (0.0) (0.0007) (0.9979, 1.24, 1.82)
CS100 (PW1542G) (59)	$0.0089\ t$ (0.0) (0.9948, -1.86, 4.40)	$0.0089\ t + 0.0049\ n_a$ (0.0) (0.67) (0.9948, -1.77, 4.34)	$0.0089\ t + 0.0007\ t_a$ (0.0) (0.37) (0.9949, -1.76, 4.32)

6.2.4 Fuel burn estimation using airport surface surveillance data

The idea is to build fuel burn models with the FDR data and confirm that these models work well with the openly available ASDE-X data. For this idea to work well, the trajectory estimates from the noisy ASDE-X data have to closely match the FDR estimates.

The trajectory information in ASDE-X data is more noisy compared to the FDR data. Figure 6-3 shows a comparison of the trajectory information for a sample flight obtained from the raw FDR data and ASDE-X data. One can notice that the heading and speed measurements in the ASDE-X data are more noisy, with several missing data points. The mean error in the speed measurements in ASDE-X data is 0.31 m/s and the mean absolute error is 0.69 m/s, considering FDR data to be the ground truth. Note that both FDR and ASDE-X have a speed resolution of 0.51 m/s. The ASDE-X measurements are reported at 1Hz frequency, however, speed measurements are missing on an average for about 46% of the taxi-out duration. These statistics were obtained from 104 A330 ASDE-X tracks in our dataset that could be mapped to the FDR data. Since the fuel burn models developed earlier need the duration of acceleration as one of the inputs, we need to ensure that we are able to estimate the acceleration information from the ASDE-X data with the same level of accuracy as we had done earlier with the FDR data. The acceleration threshold in the smoothing algorithm is adapted to ensure that we have the same mode detection statistics for the ASDE-X data as we had earlier for the FDR data (more details on this in Appendix B).

Table 6.2 shows the error statistics for the three models using the same set of flights that was used for computing the FDR error statistics. The statistics indicate that Model C performs better than the other two models even with the ASDE-X trajectories. We see a relative improvement of 22.26% in the MAPE for Model-C relative to Model-A, indicating the benefits of using the trajectory information for the fuel burn prediction. However, the relative improvement is negative for Model-B indicating a slight drop in the model performance. This shows that the duration of acceleration is a more robust predictor compared to number of acceleration events while using ASDE-X trajectories. This could be because a falsely detected acceleration event generally lasts for a smaller duration, making Model-C less sensitive to false events than Model-B. The prediction errors using ASDE-X data are slightly higher than those obtained using FDR data because the ASDE-X data is inherently more noisy. The last column in the table shows the percentage increase in MAPE using ASDE-X data relative to the error obtained using FDR data. We see that the error (MAPE) increases by about



(a) Trajectory (FDR shown in blue and ASDE-X shown in red).

(b) Speed and heading

Figure 6-3: Comparison between measurements obtained from FDR and ASDE-X data for a sample flight.

23.51% when we use ASDE-X trajectories instead of FDR data.

Table 6.2: Error statistics computed from ASDE-X trajectories using fuel burn models developed from FDR data for A330 data set

Model	Mean percentage error (MPE)	Mean absolute percentage error (MAPE)	Relative improvement in MAPE (%)	Percentage increase in MAPE relative to FDR
Model-A	-1.77	3.40	0.00	0.00
Model-B	1.71	3.46	-1.81	31.74
Model-C	-0.15	2.63	22.26	23.51

6.3 Extracting operational information from ATC voice communications

Air traffic control (ATC) voice communications contain rich operational information that has been largely underutilized. Automatic transcription of ATC communications has the potential to improve system safety, operational performance, conformance monitoring, and enhance controller training [108]. However, automatic speech recognition (ASR) systems proposed to date for the ATC applications have not yet demonstrated the levels of accuracy needed for practical deployment [137]. Developing an ASR system for the ATC domain is challenging because of factors such as noisy radio channels, high speech rates, and diverse accents. On the other hand, ATC communications contain domain-specific vocabulary and standard phraseology that can be exploited to develop algorithms tailored to the domain. Recent improvements in machine learning techniques such as deep neural networks have led to more accurate speech recognition algorithms [9]. These techniques present the potential to develop better ASR algorithms to transcribe ATC voice communications. The transcribed ATC voice data could be used to extract valuable operational information to analyse and model airport operations. For example, information such as departure call-ready time is available exclusively through ATC voice data.

In this section, we present an automatic speech recognition (ASR) model that transcribes ATC voice communications to text. The proposed model is based on an end-to-end speech recognition architecture with a deep neural network, which offers several advantages over traditional approaches, such as Hidden Markov Model (HMM). This modeling approach is easier to train, and has been shown to yield a better accuracy than traditional methods for conversational speech [9]. Additionally, we present a methodology to accurately extract operational information such as aircraft call signs and runway assignments from transcripts, using state-of-the-art techniques from natural language processing. In this thesis, we present a brief description of the model and some of the key results; and readers can find a more detailed description of this work in our paper [19].

6.3.1 Related work

ASR techniques have several potential applications in the ATC domain, including safety monitoring of live operations [170, 42, 113] and identification of anomalous aircraft trajectories [170]. In another example of safety monitoring, Chen et al. [42] developed a framework to automatically flag pilot-controller miscommunications (read back error detection) to prevent untoward incidents. With the introduction of electronic flight strips in ATC towers, speech assistants have been shown to reduce air traffic controller workload in early demonstrations [85]. ASR systems can also be used instead of pseudopilots for training air traffic controllers [163, 171], and for human-in-the-loop simulations and workload measurements in air traffic management research [120, 50]. Additionally, the increasing demand for unmanned aerial vehicle operations has stimulated the need for ASR systems [119]. However, the lack of sufficiently accurate ASR models for the ATC domain has remained a significant barrier to its deployment in these applications.

ATC applications of ASR have traditionally used HMM-based approaches, with limited consideration of end-to-end speech recognition approaches [113]. In the recently-concluded Airbus ATC speech recognition challenge, the top performing team used a HMM-based hybrid model that yielded a word error rate (WER) of 7.6% on the Airbus test data [56], with the second ranking team using an end-to-end speech recognition architecture that yielded a WER of 8.4% [143]. In another recent study, a comparison of hybrid model (HMM/DNN in Kaldi) and end-to-end speech recognition for ATC voice in English and Chinese was evaluated by Lin et al. [113]. For English ATC voice, they found that the end-to-end speech recognition model yielded a WER of 6%, performing better than the hybrid model that yielded a WER of 9%. One needs to note that the performance of the models, in terms of the WER, also depends on the quality of test data. If the test data is more representative of the training data and is less noisy, one expects better performance from the speech recognition model. Therefore, one cannot compare models based on WER if they are reported on different test sets.

6.3.2 ASR model for ATC communications

Automatic speech recognition model

The automatic speech recognition model is based on Deep Speech [80], an end-to-end speech recognition model. We use Mozilla's implementation of Deep Speech for our analysis [132]. A brief description of the model is presented in Appendix C. Additionally, training an automatic speech recognition model requires large amounts of transcribed audio data. For our analysis, we use AIRBUS-ATC dataset [56] that contains about 40 hours of transcribed ATC voice communications.

Performance metrics

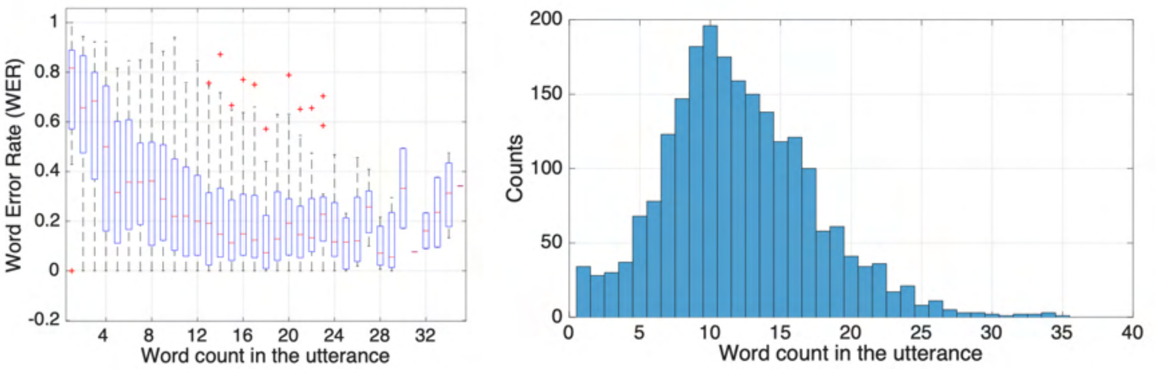
The Word Error Rate (WER) is a widely-used measure of ASR model accuracy. It is defined as the sum of the number of words in the transcribed text that are either substituted (S), deleted (D), or inserted (I) relative to the reference text, divided by the number of words in the reference (N_r):

$$WER = \frac{S + D + I}{N_r}. \tag{6.1}$$

The WER includes errors arising from filler words or other words that are not relevant in a particular context. The speed of transcription is also an important quantity. It is measured in terms of the real-time factor (RTF), namely, the duration of input audio divided by the required time to process the input. The real-time factor for our models is around 0.3, implying that they can be deployed in live operations.

Model performance

We used approximately 36 hours of the 40-hour AIRBUS-ATC dataset for training the model, 2 hours for validation, and 2 hours for testing. The ASR model yields an average word error rate of 0.22 on the 2-hour test dataset with 1,500 utterances. Figure 6-4(a) shows a box-plot of the average WER as a function of the word count in the utterance. The WER is seen to be higher for utterances with a lower word count, and is nearly constant for word counts greater



(a) WER for different word counts

(b) Histogram of word count

Figure 6-4: Statistics of word counts and WERs computed for the test set.

than six. The reason for this behavior is that missing one or two words (even filler words, e.g., 'um' or 'ah') would lead to a higher WER for smaller sentences. Figure 6-4(b) shows the histogram of the word count of utterances in the test set. The average word count is 12 words, and utterances with fewer than six words represent 7% of the test set. Additionally, 10.3% of the utterances in the test set have a foreign word or non-intelligible word in the transcription. We therefore compute the WER after excluding utterances with fewer than six words and the ones having foreign words or non-intelligible words, and this was found to be equal to 0.17.

6.3.3 Extracting operational information

ATC communications contain extensive operational information (e.g., runway assignment, heading, flight level) that are of value in decision-making. In this section, we present a methodology to automatically extract operational information from ATC communication transcripts. Natural language processing (NLP) techniques for extracting information from unstructured text can be broadly classified into statistical models and rule-based grammar. Statistical models are obtained using machine learning algorithms on the text data. On the other hand, rule-based grammar techniques are based on a collection of hand-engineered rules to perform the NLP task. We employ both the methodologies in this paper: a statistical model is used for call-sign extraction, and a rule-based grammar approach is used to extract runway information.

Call-sign extraction

The utterances of the air traffic controllers and the pilots often contain the aircraft call-sign, which serves as a unique identifier for a flight receiving or sending the information. The call-sign typically comprises of the airline telephony designation (such as "American" for "American Airlines"; "Speedbird" for "British Airways"), followed by the flight ID, which comprises of 2-4 digits, with an optional suffix of 2-3 letters. A few examples of ATC utterances are shown below, with the aircraft call-sign highlighted:

- **air nostrum eight seven six one** contact bordeaux one three three decimal seven seven five goodbye
- five thousand feet one zero one zero **easy six eight four romeo**
- **lufthansa zero one charlie** reduce speed two two zero knots

To associate a flight with the instruction or command given by the air traffic controller or pilot, one must be able to accurately extract call-signs from the transcript. There are multiple challenges to accurate call-sign extraction: (a) controllers could use multiple airline identifiers for the same airline; (b) the number of words following the airline identifier (corresponding to the numerals/NATO phonetics) can vary; and (c) there could be call-signs with just the flight number without the airline identifier. Although the digits are expected to be pronounced as separate numbers as per ICAO nomenclature, the convention can vary in different regions (for example, in the US, "2020" is often spoken as "twenty twenty" instead of "two zero two zero"). One could also have non-standard pronunciation: for example, “0" could be pronounced as ‘oh’, “9" could be pronounced as ‘nine’ instead of ‘niner’, and “F" could be pronounced as ‘fox’ instead of ‘foxtrot’.

Given these variations, a rule-based grammar approach for call-sign extraction might not be feasible. We instead use a statistical approach to train a model to identify call-signs using transcripts of ATC communications. In order to extract call-signs, we use Named Entity

Recognition (NER), a standard NLP technique, to classify information within unstructured text into predefined categories. For the call-sign extraction problem, the objective is to identify a sequence of words in each utterance, and to categorize it as the call-sign, if it exists. We use a Python library called Spacy, which is one of many standard libraries for NER [164]. The NER model in Spacy is based on a deep convolution neural network and uses sub-word features. To categorize a particular word, the model accounts for the neighbouring three words on either side. A key advantage of the sub-word feature is that it can identify call-signs even if there are spelling errors in the transcripts, a particularly useful property for ASR-generated transcripts.

Performance metrics: The accuracy of the model is evaluated using three performance measures: precision, recall, and F-1 score. We introduce some notation to define these performance measures: Let A be the number of instances in the test set when there is no call-sign in the reference text, and the model also detects no call-sign. Similarly, let B be the number of instances when there is a call-sign in the reference text, but the model does not detect a call-sign. Let C be the number of instances when there is no call-sign in the reference text, but the model erroneously detects a call-sign. Let D and E be the number of instances when the detected call-sign is correct and incorrect, respectively, among the instances when the reference text has a call-sign. The mathematical expressions for precision (P), recall (R), and F-1 score are as follows:

$$P = \frac{D}{C+D+E}; \qquad R = \frac{D}{B+D+E}; \qquad F-1 = \frac{2PR}{P+R}. \tag{6.2}$$

Precision is the fraction of instances for which the call-sign is correctly extracted from among the instances when a call-sign is extracted. Recall is the fraction of instances for which the correct call-sign is extracted from among the instances when the call-sign is present in the reference text. The F-1 score is the harmonic mean of precision and recall.

Results: The model was trained using 25,000 samples of labelled data, which contained the transcript of the utterance, and the corresponding call-sign in the utterance (if present). The independent test data set contained approximately 3,000 utterances. For the call-sign extraction task from the reference transcript, the precision was 0.97, the recall was 0.93, and the F-1 score was 0.95. In other words, 97% of the call-signs that were extracted matched the reference call-sign, and 93% of the call-signs in the reference text were extracted correctly. Table 6.3 shows a few example transcripts of the utterances in the test set, along with the reference call-sign contained within the transcript, and the extracted call-sign from the model. The first three rows in the table are instances in which the call-sign was correctly extracted. The next three rows illustrate cases wherein the call-sign present in the text was not extracted, which would impact the recall score. In these cases, the primary reason for the error was that either the airline identifier in the test set was very different from those seen in the training set, or that the call-sign did not follow the usual convention. The last row in the table shows a case in which the model incorrectly extracted a call-sign when there was none in the reference transcript, which would impact the precision score. Overall, a F-1 score of 0.95 is a good initial step, and can be improved by increasing the amount of training data. One needs to note that even if the call-sign was not seen in the training data, the model is capable of correctly identifying the call-sign based on neighboring words that gives it context.

Next, we look at the performance of the call-sign extraction task on the transcribed voice data that is output from the speech recognition model. For the call-sign extraction task from the transcribed voice data, the precision is 0.81, recall is 0.57, and the F-1 score is 0.67, on an independent test set. The performance here is worse than in the case where we had the actual transcripts, because of inaccuracies in the voice transcription. It is worth noting that even when the transcription of voice is not completely accurate (as reflected by the word error rate), it may be possible to accurately extract the call-sign from the transcript if the call-sign part of the utterance is transcribed correctly.

Some limitations of this statistical approach to call-sign extraction must be noted: (a) detecting airlines unseen in the training set can be challenging, even though the approach considers neighboring words for context; (b) call-sign digits are pronounced differently in the US and in Europe (for example, flight number “2020" is pronounced ‘two zero two zero’ in Europe as per ICAO standard, but ‘twenty twenty’ in the US). These limitations can be

Table 6.3: Extracting call-signs from reference transcripts.

Reference call-sign	**Reference Transcript**	**Extracted call-sign**
easy four seven tango lima	roger we call you huh short final easy four seven tango lima	easy four seven tango lima
beeline four papa golf	report huh short final maximum one sixty beeline four papa golf	beeline four papa golf
swiss one juliet bravo	keep rolling and vacating via mike eight swiss one juliet bravo	swiss one juliet bravo
easy _ whiskey	toulouse tower @ easy _ whiskey established on the ils three two right	
airbus delta sierra	airbus delta sierra after takeoff you will maintain runway axis	
binair seven alpha	# huh good huh @ huh binair seven alpha approaching november eight	
	we leave via sierra three huh india lima	sierra three huh india lima

overcome by including more diverse data sources for training.

Extracting runway information

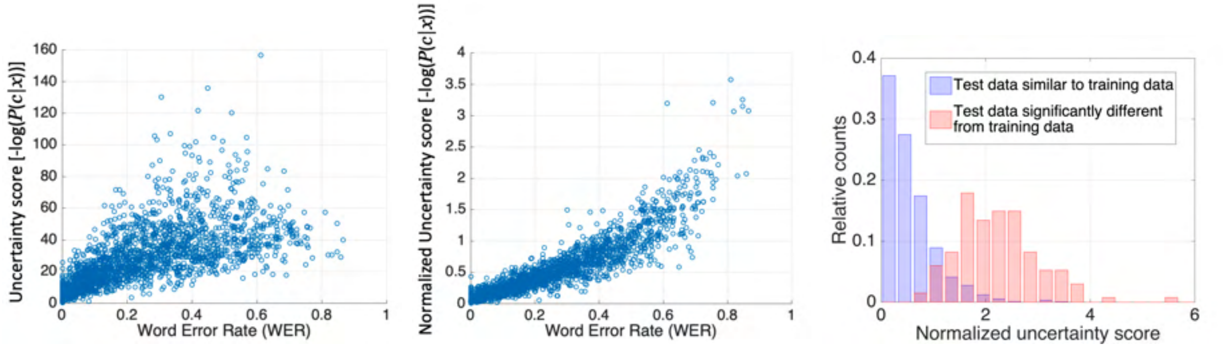
A rule-based grammar was used for extracting runway information from the transcript. Reasons for employing a rule-based grammar approach rather than a statistical approach for runway extraction include: (a) the datasets do not contain the labelled runway information needed to build a statistical model; and (b) relatively few runway numbers are uttered in the datasets (because the voice recordings are from a small number of airports), which is not sufficient to obtain a statistical model that generalizes well. Furthermore, the utterances of runway numbers are highly structured, motivating the use of rule-based grammar. Runway numbers are uttered in the following manner: one or two digits followed by 'left', 'right' or 'center'. The transcript is searched for such patterns to obtain the runway information for extraction. We hand-labelled about 200 utterances to test the performance of a rule-based grammar approach for extracting runway information. The precision, recall and F-1 score is 1 on the reference transcript, indicating that the rule-based approach is perfectly accurate in extracting runway information. Runway information extraction using the transcribed text shows very good performance, achieving a precision of 0.97, recall of 0.93 and F-1 score of 0.95.

6.3.4 Evaluating transcription accuracy in the absence of ground truth data

Calculation of the word error rate requires the availability of ground truth data (i.e., the actual transcript). However, the practical deployment of decision-making using ASR requires an estimate of the likely accuracy of the transcription, even in the absence of ground truth data. In other words, it is valuable to know when the ASR models are expected to be accurate, and when they are not. To this end, we propose to use the *uncertainty score*, defined as the negative logarithm of the confidence score ($Q(c)$), as a surrogate for the word error rate. Note that the confidence score is obtained directly from the speech recognition model and one can find more information about this metric in Appendix C. The confidence score of an utterance, and therefore its uncertainty score, depends on the character probabilities of the transcribed text, as determined by the acoustic model and language model of the ASR system. Figure 6-5(a) shows a scatter plot of the uncertainty score and word error rate, for each utterance in the test set. For the purpose of illustration, the uncertainty score was computed using only the acoustic model. The figure shows a clear correlation between the uncertainty score and the word error rate. This correlation increases further when one considers the normalized uncertainty score, defined as the uncertainty score divided by the number of characters in the transcription of the utterance. Figure 6-5(b) shows a scatter plot of the normalized uncertainty score and the word error rate, for the utterances in the test set. The normalized uncertainty score increases with the word error rate, and can be used to identify instances in which one would expect a high WER (i.e., low accuracy of the ASR model).

For applications in which the ASR model is used as part of a decision-support tool, one could flag, or even exclude, transcriptions that have a higher-than-average normalized uncertainty score. The uncertainty score can also be evaluated at the word-level, when the accuracy of the extracted operational information is a quantity of interest. The uncertainty score can also be used to identify off-nominal events during which we expect the performance of the speech recognition model to degrade (for example, because the speech-rate or the phraseology are significantly different from the nominal periods over which the model was trained).

Figure 6-5(c) shows the distribution of the normalized uncertainty score computed over two test datasets: (a) one that is similar to the dataset used for training the ASR model (shown in blue, and representative of conversations during nominal events); (b) a test dataset that is significantly different (in terms of accent and speech rate) from the training data set (shown in red, and representative of conversations during off-nominal events). We observe that the distribution of the normalized uncertainty score for the off-nominal data is significantly different (with a higher mean value, indicative of higher WERs) from the distribution obtained for the test dataset that is similar to the training data. Hypothesis testing techniques can be used to determine if an utterance (or a series of utterances) lies outside the nominal distribution in a statistical sense, and to flag those periods as off-nominal events.



(a) Uncertainty score

(b) Normalized uncertainty score

(c) Distribution of normalized uncertainty score

Figure 6-5: Uncertainty score of the transcription

Chapter 7

Conclusions

7.1 Summary

In this thesis, we presented new queuing network models of airport surface operations. The proposed models reflect the time-varying nature of capacity and demand seen at airports, and could be used to predict the queue lengths at congested locations on the airport surface and the taxi times of aircraft. The key highlight of the queuing network model is that it can account for multiple hot spots of congestion on the airport surface and the queuing dynamics could be represented using a set of simple ordinary differential equations. The modeling approach was applied to multiple major airports with different operating characteristics in order to demonstrate its general applicability. We also demonstrated that the airport surface model could be augmented with a terminal departure airspace model to predict the transit time from the departure gate to the final departure fix.

Further, the models were used to develop and validate algorithms for departure metering, an approach that leads to lower taxi-out delays, resulting in lower operating cost and emissions. In particular, we proposed two new algorithms for departure metering based on optimal and robust control techniques, and they were compared with NASA's ATD-2 logic that is currently under field trials. Using stochastic simulations of the airport operations, we showed that our robust control algorithm performs better (by about 11%[1]) than the

---

[1] Obtained using simulations for CLT; varies for other airports depending on the congestion level

ones currently planned by the industry for future operational use. Moreover, the proposed queuing network models could also be used to estimate the optimal value of the excess queue time buffer, which is an important parameter within the ATD-2 logic.

The departure metering algorithms require information on departure demand as inputs, usually in the form of the Earliest Off-Block Times or EOBTs. In this thesis, we evaluated the accuracy of the EOBT information currently published by the airlines, and conducted a parametric analysis of the impacts of EOBT uncertainty on the performance of departure metering algorithms. The results showed that EOBT uncertainty can significantly reduce the departure metering benefits in terms of taxi-out time and fuel burn savings. For example, reducing the standard deviation of the EOBT error from the current value of around 6 min to 4 min at a 20-min planning horizon would yield an additional annual fuel burn savings of 200 metric tonnes at EWR, 1,300 metric tonnes at DFW, and 800 metric tonnes at CLT[2]. This analysis can potentially incentivize airlines to improve the accuracy of reported EOBTs. More importantly, several other downstream traffic management initiatives are planned to utilize EOBTs, and therefore characterizing EOBT uncertainty is valuable for such applications.

In addition to developing algorithms for improving airport operations, the tools and techniques for analyzing queuing networks that were presented in this thesis also have a broader applicability. Using point-wise steady state approximation, we were able to represent the dynamics of the ensemble mean queue length using a simple ordinary differential equation that could account for propagation delays between servers and general service time distribution. The dynamical system representation allowed us to develop robust controllers to track the queue length at a desired value even in the presence of parameter uncertainties. For the case of a single queue, we were able to determine the error bounds for tracking the queue length using the predictor-based feedback controller. Additionally, the dynamical system representation of the queuing process allows us to use reachability analysis to analyse the performance of queuing networks.

The adaptation of queuing models to airport operations relied on operational data such

---

[2] Simulation results with the ATD-2 logic.

as flight tracks and flight schedules. Such aviation data sources play a critical role in developing accurate models and practical decision support tools. In addition to flight tracks and schedules, we have shown how one could utilize other emerging aviation data sources to characterize airport operations. In particular, we have developed a trajectory-based fuel burn model to analyse the environmental impacts of aviation. Our models perform better than the traditional fuel burn models, and they also have the potential to estimate taxiway noise and emissions at airports. Additionally, we have proposed a methodology to extract operational information such as runway number and flight ID from ATC voice communications data using recent techniques in automatic speech recognition and natural language processing.

7.2 Practical implementation: opportunities and challenges

Departure metering is a part of FAA's Terminal Flight Data Manager (TFDM), which is an airport surface management solution for NextGen in the US [69]. The departure metering concept has undergone several field trials at major airports [160, 169, 127], with the most recent one conducted at Charlotte airport under the ATD-2 program (during 2017-2019) [149]. The concept is scheduled for operational deployment across major US airports in a phased manner in the near future (originally scheduled for 2020, but delayed due to the COVID-19 pandemic) [62]. In addition to the US, departure metering is currently being deployed at major airports in Europe and Asia [60, 39].

The underlying departure metering logic across all the surface management programs (in the US as well as in Europe and Asia) is similar to the NASA's ATD-2 logic. Therefore, the analysis of the operational benefits of the ATD-2 logic that was presented in this thesis is of potential interest to many operators. For example, choosing the optimal buffer parameter, which decides the departure metering benefits, can be obtained using our numerical analysis. Such an analytical approach can be efficient and cost-effective instead of running several field trials across different airports that can be prohibitive for various reasons. Additionally, we showed that our proposed robust control approach that explicitly considers model uncertainty performs better than the ATD-2 logic in numerical simulations. The next natural step is to run the algorithm in field trials to test the efficacy in the real-world environments. Further, we would like to highlight that the backbone required for operational implementation of any new departure algorithm such as surface surveillance and digital data exchange is already (or soon to be) deployed across major airports. Therefore, implementing our proposed departure metering logic would require just a software update.

In addition to departure metering algorithms, the EOBT analysis presented in this thesis provides insights on the benefits of improving the EOBT accuracy. Airlines update the time a flight would be ready to pushback at irregular intervals close to the scheduled departure time and this is published as EOBT. The EOBT information, which is essentially an airline's estimate of when a flight would be ready, can deviate from the actual pushready time because of various operational reasons. Departure metering algorithms use EOBTs as inputs to make scheduling decisions and our analysis was able to quantify the marginal improvement in departure metering benefits due to reduction in EOBT uncertainty. For example, we found that by reducing the standard deviation in EOBT error by 2 min at DFW, the benefits in terms of taxi-out time savings from departure metering increases by about 75%, which results in an additional fuel savings of about 3.4M kg/yr. The airlines can use such estimates to decide on investment decisions to develop better models to reduce the EOBT uncertainty.

There are a few challenges in translating the departure metering algorithms presented in this thesis to the real-world. First, we need to recognize that the models were trained using historical operational data. As one might expect, the airport operations change with time and this might influence the performance of these models. Certain aspects such as traffic demand are ingrained in the model, therefore the queuing model is expected to work well even if the volume and nature (demand-profile) of the traffic changes at an airport. However, the model parameters would need refinement if there are other operational changes such as changes in fleet mix or infrastructure. Along with the model, one might have to update the parameters in the departure metering algorithm (such as the buffer time in the case of ATD-2 logic or the target queue length in the case of robust control approach). The parameters associated with the model and the control algorithm could be revised by monitoring the predictive performance of the model. Another potential solution to this problem, is to design an adaptive controller to update the model parameters with measurements. Further, one could also update the model predictions based on the errors as shown in Section 4.6.1.

Another aspect that has been overlooked in this thesis is the need to incorporate flights with other constraints such as delays assigned due to Traffic Management Initiatives (TMIs). Additionally, departure metering posses a challenge in the US because of jurisdictional issues associated with gate metering. FAA only has jurisdictional control over the active movement area in the US, and the ramp areas (including the gate pushbacks) are controlled by the ramp controllers (usually, the airline). However, there is push to accommodate gate metering by coordinating with the ramp controllers.

Apart from aiding the development of control algorithms for departure metering, the proposed queuing models of airport traffic movements can be used in industry software tools to assess fuel burn and emissions. We are currently working with the FAA to propose enhancements to the Aviation Environmental Design Tool (AEDT), a tool that is widely-used by the regulatory agencies for predicting aviation environmental impacts. The enhancements involve updating the airport queuing models to improve taxi-time predictions as well as proposing accurate fuel burn models, such as those presented in Section 6.2 [23].

7.3 Opportunities for future research

In this section, we present opportunities for future research ranging from improvements to the departure metering algorithm to exploring other potential applications of the queuing network model.

7.3.1 Improvements to departure metering algorithm

Integrated control of arrival and departure queues

The departure metering algorithms presented in this thesis focused on reducing the departure queues without accounting for arrival queuing. One could improve the efficiency of operations further by considering integrated control of arrivals and departures. The objective would be to minimize the weighted sum of all queue lengths on the airport surface by controlling the pushback rate and the arrival rate, accounting for the number of holds in the air (arrivals) and at the gate (departures). The weights in the cost function could be chosen to penalize airborne holds more than the ground holds. In addition to assigning gate-holds, the algorithms can be extended to assign runways to ensure that the runways are equally loaded and utilised, thereby increasing the overall airport throughput. However, one needs to note that there is less flexibility in assigning runways because there are constraints arising from the runway length required by aircraft, departure fix assignments, etc.

Incentives to improve EOBT accuracy

The analysis in Chapter 3 assumed that the EOBT uncertainty distribution was the same for all airlines at an airport. However, as seen in Section 3.3 in Chapter 3, the uncertainty in EOBT varies by airline. An interesting question is whether an airline would be rewarded for improved EOBT accuracy by achieving greater taxi-out time reduction. A related question would involve opportunities for airlines to manipulate the system by publishing inaccurate EOBTs to obtain more benefits.

To incentivize airlines to publish more accurate EOBTs, the ATD-2 program categorizes flights into a "planning" group and an "uncertainty" group based on the historical accuracy of their EOBT information. The idea is to give priority to the flights in the planning group so that they are able to get more benefits. Challenges remain with this approach: (a) Deciding on a criterion to categorize flights into two groups (airlines could game by reporting accurate EOBTs during periods with less congestion and incorrect EOBTs during periods with congestion to obtain more benefits); and (b) developing strategies to disincentivize the uncertain group. One possible approach is to penalize airlines if they do not meet the TOBT assigned by the controller. For example, under the Airport Collaborative Decision Making concept, if a flight is unable to pushback within TOBT+5 min, its existing slot is cancelled and it receives a new TOBT [39].

Impact of EOBT information updates

This thesis did not consider the impact of EOBT updates by the airlines. As we had discussed in Section 3.3, airlines update the EOBT information at irregular intervals until a flight pushes back from the gate. If gate-hold decisions are made 20 min prior to a flight's EOBT which continues to be updated, then one would have to determine the most efficient way to accommodate flights with updated EOBTs.

If an airline updates a flight's EOBT, we have one of the following two cases: (a) the new EOBT is prior to the assigned pushback time, in which case, the flight needs to wait at the gate before pushing back (equivalent to an additional hold); or (b) the new EOBT is after the assigned pushback time. In the latter case, the controller considers the flight to be entering the system at $\max((t + T_p), \mathrm{EOBT})$, and reassigns the gate-hold for that flight. In the above discussion, t is the time at which the EOBT was updated and T_p is the planning horizon. Both cases penalize the airline, incentivizing airlines to report a more stable EOBT. This is just one approach by which EOBT updates can be accommodated; we could also devise alternative strategies.

7.3.2 Emergence of advanced air mobility

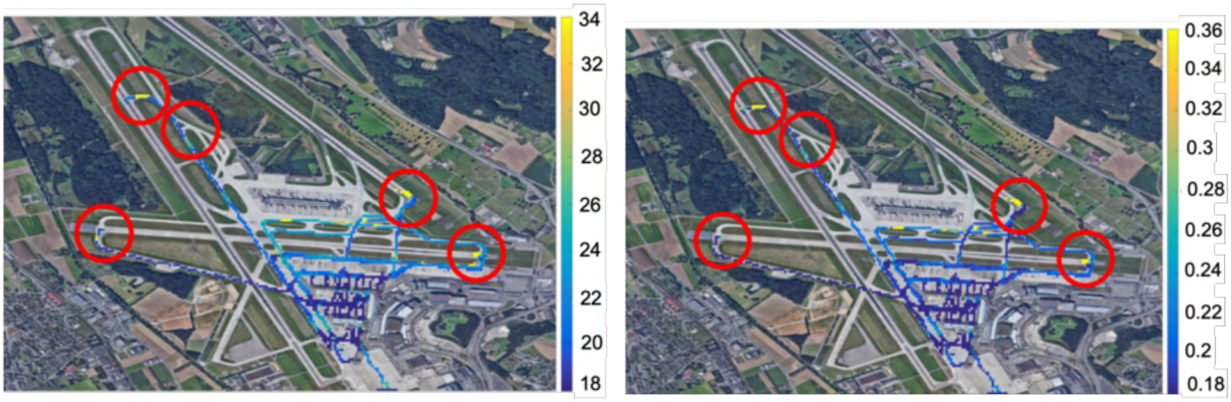
The aviation domain is currently undergoing a significant transformation with multiple technological advances, operational changes and an increasing focus on environmental sustainability. Advanced air mobility (AAM) aims to transport people and cargo between places previously under-served, both in urban and rural areas, using new cost-effective vehicle technologies incorporating various levels of electrification and autonomy. The AMM concept includes promising ideas such as drone delivery and urban air mobility. The AAM revolution is regarded as the next disruption in the aviation industry, with some projections that have estimated the AAM market value to reach $115 billion US dollars annually by 2035 [114]. The current AAM concepts call for using either existing ground infrastructure (heliports on tops of building, parking lots, under-utilised piers, etc.) or developing purpose built veriports. However, there is limited study on integrating AAM operations at or near airports, and their impact on conventional air traffic operations [177]. Recent airline investments indicate that the airlines are interested in operating a new generation of small electric vertical take-off and landing (VTOL) aircraft to move their passengers to and from their hub airports in congested areas [5]. These trends indicate that one can expect higher traffic demand from these unconventional aircraft at or near airports, requiring airport planners to think ahead to devise strategies to improve integrated operations.

7.3.3 Data analytics to evaluate airport operations

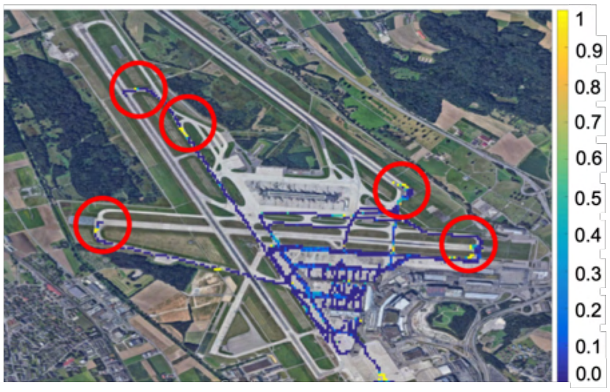
Environmental impact assessment of airport operations

The trajectory-based fuel burn models proposed in Chapter 6 have the potential to provide spatial distribution of fuel burn on the airport surface. Figure 7-1 shows heat maps of average value of engine fan speed (N1%), fuel flow rate and the estimated acceleration indicator value, obtained from FDR data for 69 A320 flights operating at Zurich Airport (LSZH). The engine fan speed (N1%) is closely coupled to the engine thrust, and it is a good indicator of the thrust level. We see a strong correlation between these quantities. These correlations could be used to estimate heat maps of thrust level and fuel flow rates from the acceleration indicator values that can be obtained from openly available surface surveillance data such as ASDE-X.

Moreover, the spatial distribution of fuel burn is essential to accurately model the dispersion of emissions around the airport, which is of significant interest to the regulatory agencies. The amount of emissions is generally modelled as a function of the fuel flow rate and the engine thrust setting. Additionally, the acceleration events presented in our analysis correspond to the breakaway thrust conditions, and serve as a measure for change in thrust



(a) Average engine fan speed (N1%) (b) Heat-map of average fuel flow rate (kg/s)



(c) Average value of acceleration indicator

Figure 7-1: Heat maps obtained by averaging values over 69 A320 taxi-out flight trajectories in LSZH.

setting.

Our trajectory-based fuel burn models can also aid in airport noise modeling. Airport noise impacts residential areas that are close by, particularly important for airports located near urban areas such as in Boston (BOS) and Chicago (MDW). The noise profile of an aircraft on the surface depends on the engine thrust level, that is in turn related to the fuel flow rate. Spatial distribution of fuel burn is directly related to the distribution of noise sources on the airport surface, which can be used to analyze airport noise levels due to taxi operations.

Automatic speech recognition of ATC voice communication

Alternative language models: In our work, we used a standard word N-gram language model for simplicity, and focused primarily on the acoustic model and the extraction of operational information. However, prior studies have shown that more sophisticated language models such as the class N-gram or neural-network-based language models can improve the accuracy of ASR systems [136, 113]. Another potential benefit of the class N-gram model is that one can efficiently incorporate out-of-vocabulary data, such as new airlines or navigational aids that are specific to a particular region.

Semi-supervised learning: The availability of sufficient amounts of high-quality data is a key factor in improving the performance of ASR models. Our experiments have shown that the use of diverse datasets for model training results in more general models. However, there are not many openly-available corpora for ATC communications, and manual transcription of ATC conversations requires significant effort. A potential approach is to use semi-supervised learning, namely, to utilize widely-available but untranscribed ATC audio data [1] to improve the accuracy of the ASR model. The proposed approach would be as follows: (a) train a preliminary model with the existing ATC speech corpora; (b) use this preliminary model to transcribe utterances of ATC voice communications that does not have transcripts; (c) evaluate the accuracy of each of the utterances using the uncertainty score[3] as a metric; and (d) re-train the ASR model by including the utterances that yielded a low uncertainty score in the training data. Although semi-supervised learning for ATC applications has been previously attempted [165], we believe that the use of the normalized uncertainty score to select data for retraining models is a promising direction for further investigation.

7.3.4 Queuing network representations of other systems

In this thesis, we have presented several techniques to represent queuing networks, analyse their performance and developed algorithms to reduce congestion in queuing networks, with an emphasis on airport surface operations. In general, queuing networks can be used to

---

[3] Uncertainty score as defined in Section 6.3.4

model congestion in various other domains. We first present a couple of other systems in air transportation that could be represented using queuing networks, and this is followed by a discussion on potential applications in other domains.

For other areas in air transportation

Traffic flow model for the National Airspace System (NAS): Congestion-related delays at an airport can propagate to other airports due to the networked nature of the air transportation system. The entire air traffic network can be modeled using our analytical queuing network model, with the nodes representing individual airport queues and edges corresponding to the traffic flows between the airports. Such queuing representation for the NAS has been used earlier to predict flight delays, however, those models relied on queuing simulations [147, 186]. The dynamics of the queuing network being represented as a differential equation (as in our analytical model) makes it amenable to develop robust control strategies to efficiently manage traffic over the entire network, in addition to predicting flight delays. However, we acknowledge that there are several challenges associated with modeling the complexities of the NAS.

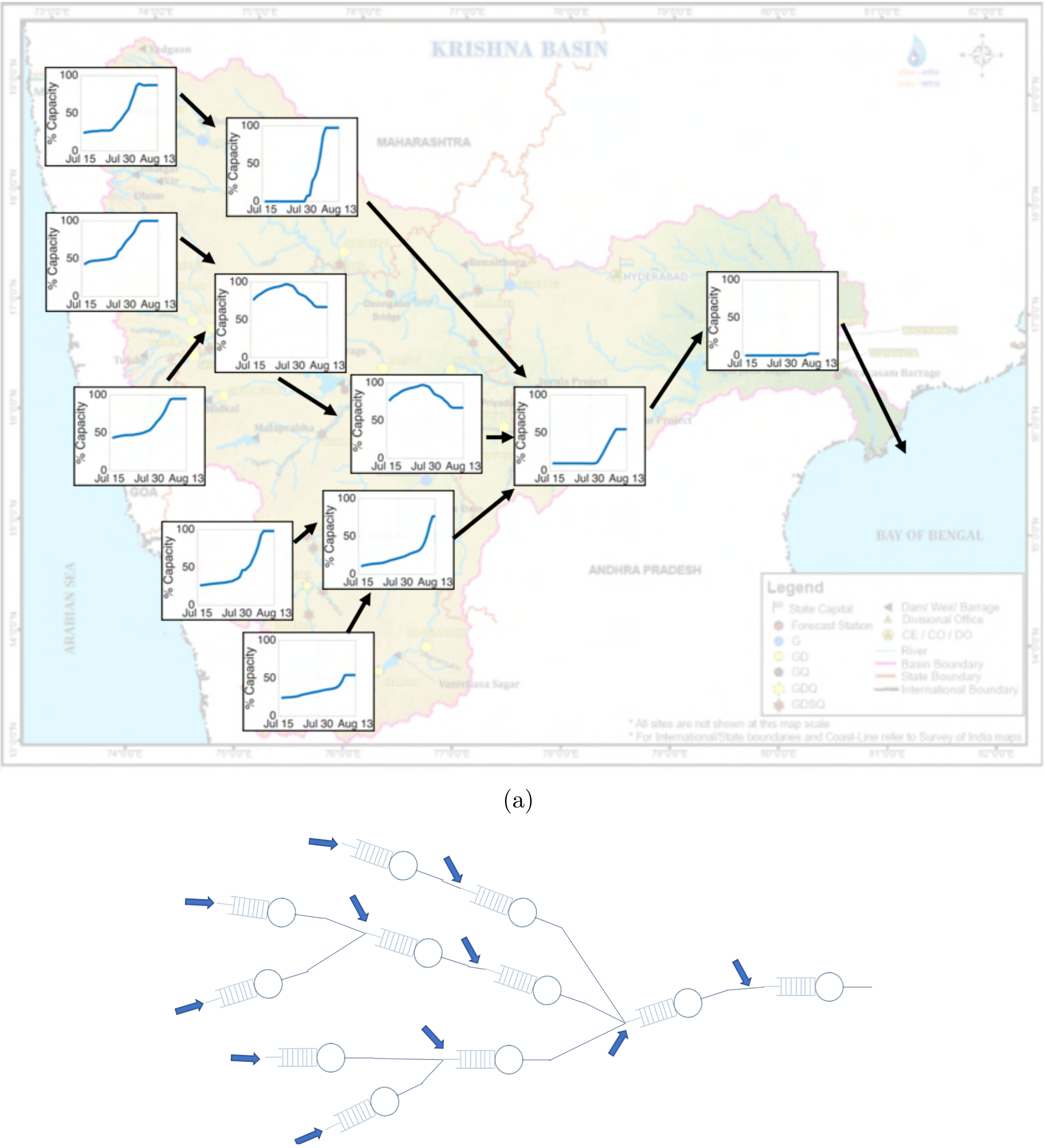
Passenger queuing in airport terminals: The primary focus in this thesis has been on aircraft queuing on the airport surface, which is a part of the airside operations. However, queuing is also quite evident in the landside operations, such as passenger queuing at the check-in counters and security lanes. The wait time, that represents the quality of service, often depends on staffing at these check points. For example, we did a preliminary analysis using operational data (for the period Jan 11 - Feb 12, 2014) from Shenzhen Bao'an International airport (ZGSZ), the 5^{th} busiest airport in China in terms of passenger counts [46, 184]. Our analysis indicated significantly high median queue length of passengers in the security lanes (about 30), as a result of inefficient use of the available security lanes. An interesting problem of practical importance is to determine the optimal staffing policy accounting for the time-varying passenger arrival rate at an airport. There has been prior work on characterizing and modeling passenger queuing at airport terminals [150]. However, there has been very limited work on staffing policy, particularly determining an optimal policy for the security lanes [43, 44].

Queuing networks in other domains

For over a century, queuing networks have been used to represent congestion in a wide variety of systems such as communication networks, urban traffic flows, manufacturing and health care systems [178, 181, 140, 131, 130]. These domains are well studied and there is extensive literature on modeling such systems as queuing networks. Here, we would like to highlight an interesting application of optimizing water reservoir networks, which is not typically represented as queuing networks.

Optimal operation of water reservoir networks is critical with the increase in water-use as well as more frequent occurrence of extreme weather events. There has been extensive work on modelling and operation of reservoir networks [38, 141, 182]. The potential benefits of multi-reservoir optimization has been demonstrated through case studies in various regions such as Upper Rhone River basin in Switzerland [94] and Lake Como water system in Italy [37]. Most often, the discharge rates from the reservoirs in the network are determined using dynamic programming or other stochastic optimization techniques. However, the challenge has been to accommodate larger reservoir networks that lead to a higher dimensional state space. Additionally, the focus has been on controlling reservoirs to maximize power generation or water for irrigation. Here, we present a slightly different motivation that focuses on flood management. More importantly, we present an analysis that motivates the use of queuing networks to represent water reservoir networks.

To motivate this particular problem, we consider an example of a recent flood event in India, that was exacerbated by the mismanagement of reservoirs. In the first two weeks of August 2019, the upstream regions of the Krishna river basin received heavy rainfall, leading to flooding. This caused significant damage to property and loss of life, impacting millions of people [173]. The management of reservoirs has largely been overlooked, which could have reduced or even prevented flooding in the region. Figure 7-2(a) shows the storage levels of the major reservoirs in the Krishna Basin as a percentage of their capacity, with the arrows indicating river connectivity and direction of flows. We can see a sharp spike in the storage



(a)

(b)

Figure 7-2: (a) Reservoir storage levels as a percentage of capacity (%) over the days leading up to the floods [10], (b) Queuing network representation of the reservoir network.

levels in the upstream reservoirs as a result of heavy rains during early August, and the storage levels reach their capacity, leading to flooding. However, the downstream reservoirs remain largely empty during this period. This clearly shows that proactive decisions were not taken to release water upstream when the forecasts had indicated heavy rainfall. Instead, a very myopic decision to release water downstream only after filling up the upstream dams led to the disastrous flood situation.

From an operational view-point, there is an interesting trade-off for releasing water from the reservoirs: a higher discharge might lead to insufficient storage at the end of monsoon for the dry season, whereas, a lower discharge might lead to flooding when there is a period of heavy downpour. However, one can consider the rainfall forecasts to decide on the optimal release rate from the reservoirs.

We propose modelling the reservoir system using our queuing network model. Figure 7-2(b) shows the queuing network representation for the Krishna river system considering the major reservoirs. The queue length represents the storage value, the network topology is based on the river connectivity, the service rate represents the reservoir discharge rate and the arrival rate represents the effective exogenous inflow into the reservoir. The reservoir management problem is to determine an optimal service rate (discharge policy) that is robust to uncertainties in the arrival rate (rainfall inflows) such that the total queue length is maximized at the end of the horizon (maximize storage), while ensuring system constraints on queue buffer (storage capacity) and service rate (reservoir discharge).

We developed algorithms to control the arrivals into a queuing network for the departure metering application. The reservoir management problem provides a good motivation to develop algorithms to control the service rate of servers in a queuing network.

Appendix A

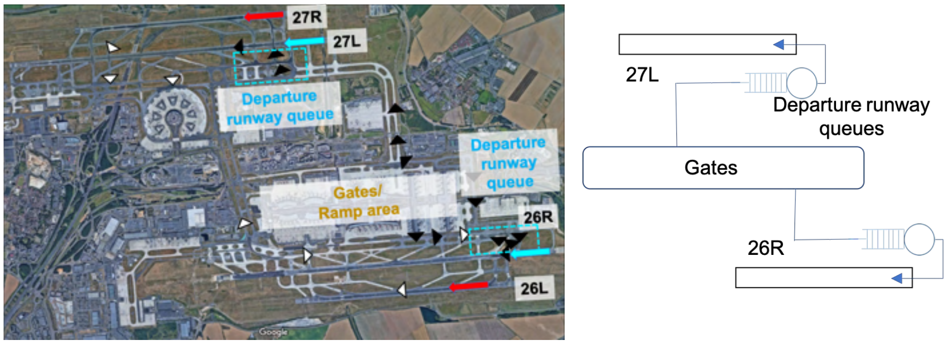
Queuing model and departure metering for Paris airport (CDG)

Overview of CDG operations

CDG is the 2^{nd} busiest airport in Europe and the 11^{th} busiest airport in the world in terms of aircraft movements, with 1,300 flights/day and 66 million passengers in 2016 [91]. The airport has four parallel runways, and operates under two broad runway configurations: West-flow $(26L, 27R|26R, 27L)$ and East-flow $(09L, 08R|09R, 08L)$. Our work focuses on the more frequently-used West-flow configuration (75% of operations in July-August 2017). Figure A-1(a) shows the CDG layout along with a snapshot of the traffic in the West-flow configuration. Departing flights are represented by black triangles and arriving flights represented by white ones. The departure and arrival runways are indicated using blue and red arrows, respectively. We note the queues of aircraft near the departure runways.

Comparison of CDG and CLT

Although CLT and CDG handle approximately the same number of aircraft movements, their fleet mixes are significantly different, with CDG handling a larger percentage of 'heavy' aircraft (25%) compared to CLT (2%). CDG operates under instrument meteorological conditions (IMC) capacity even in visual conditions, unlike CLT. Both airports have same number of departure runways, but CLT has mixed operations. Consequently, the declared departure capacities in good weather conditions are similar at both airports. Another differentiating feature is the nature of demand. Fig. A-2 shows the number of pushbacks (per 15 mins), declared departure capacity and total runway queue length for a typical day at CDG (Fig. 2-2 showed a similar plot for CLT). Departure demand is significantly banked at CLT compared to CDG, resulting in periods of increased congestion and the formation of larger queues. One can also notice that the demand at CDG rarely exceeds capacity since it is slot-constrained, unlike CLT. The higher imbalance between demand and capacity at CLT leads to higher taxi-out delays. The average taxi-out delay is 4.2 min at CDG and 9 min at CLT. We would therefore expect lower benefits from departure metering at CDG compared to CLT.



(a) Airport layout with a snapshot of traffic movement.

(b) Queuing representation.

Figure A-1: CDG in West-flow runway configuration.

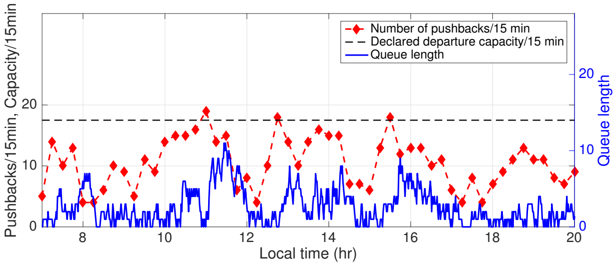


Figure A-2: Number of pushbacks (per 15 min), declared departure capacity and queue length for a typical good weather day at CDG (July 10th, 2017).

Queuing network model of CDG

The runways are the primary bottleneck at CDG, leading to the formation of departure runway queues. The taxi-out process is represented using a single queue, one for each departure runway as shown in Fig. A-1(b). After pushback, an aircraft enters the departure runway queue after spending an unimpeded gate-to-runway time. The dynamics for the evolution of the departure runway queues are given by

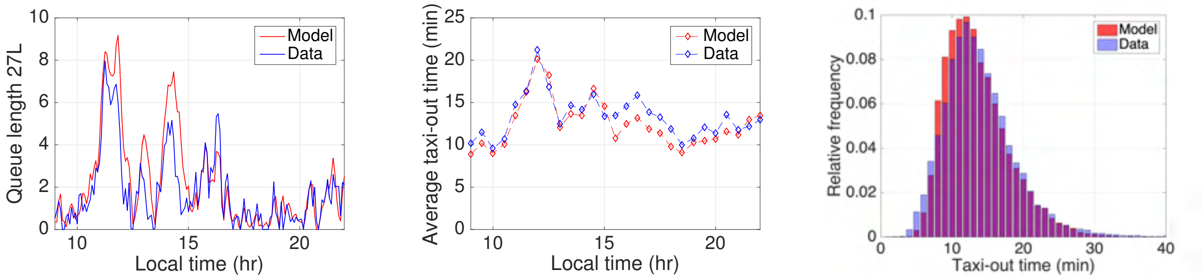
$$\dot{x}_{r_i} = -\mu_{r_i}(t)\frac{C_{r_i}(t)x_{r_i}(t)}{C_{r_i}(t)x_{r_i}(t)+1} + u_{d_i}(t-\tau_{gr_i}),\ i = 1, 2 \tag{A.1}$$

where x_{r_i} represents the queue length of the i^{th} departure runway, and τ_{gr_i} is the average unimpeded travel time from the gate to the i^{th} departure runway, u_{d_i} represents the pushback rate to the i^{th} departure runway. The pushback rate is computed as the number of aircraft pushing back from the gate in a given time interval (5 min in this paper). The time delay in the dynamics accounts for the travel time from the gate to the departure runway.

The queue length can be predicted by integrating the dynamics forward in time with appropriate server parameters and pushback rate. The wait times of aircraft entering the queue are determined using the predictions of queue length and time-varying mean service rates. The taxi-out time is then determined as the sum of the unimpeded gate-to-runway time plus the waiting time in the queue. The unimpeded times are computed as the 10^{th} percentile of the empirical taxi-time distribution obtained from data.

Predictive performance of the queuing model

Fig. A-3(a) shows a comparison of the predicted and observed departure runway queue lengths at CDG for a typical day. The data corresponds to a time-based definition of queue length, in which an aircraft is said to be in the runway queue if it has spent unimpeded gate-to-runway time after pushback but is yet to take-off. Our analysis has found that this time-based definition is consistent with the observed physical queue. The taxi-out times for this particular day, averaged over 15-min windows, are shown in Figure A-3(b). These figures show a good match between the predictions and observed values.



(a) Queue length - 26L (July 10, 2017)

(b) Average taxi-out time (July 10, 2017)

(c) Taxi-out time distribution (14,100 flights)

Figure A-3: Comparison between model predictions and data for CDG.

Table A.1: Error statistics using analytical queue models.

Airport	**Number of departures**	**Taxi-out time (min)**			**% of flights**
		Mean	**ME**	**MAE**	**|error| < 5 min**
CDG	14,100	13.3	-0.3	3.0	82.4

Table A.1 shows the aggregate error statistics of taxi-out time prediction for individual flights, computed for an independent test set of 14,100 departures for CDG. Flights with taxi-out times greater than 50 min were not included while computing the statistics. The mean errors (ME) and mean absolute errors (MAE) are found to be small relative to the mean taxi-out times. A good overlap can be seen between the predicted and actual taxi-out time distributions (Fig. A-3(c)). The mean error is slightly negative. From the point of view of departure metering, positive errors are not desirable since they correspond to an over-prediction of taxi-out times, leading to overly-aggressive holds and unnecessary wheels-off delays.

Comparison of departure metering algorithms

Table A.2 validates the simulations by comparing the taxi-out time predictions from the simulator in the baseline case (without any metering) to actual data over multiple days. The results indicate that the simulations are quite accurate, and that the errors are small relative to the mean taxi-out times. The stochastic simulations are used to evaluate the benefits of departure metering for three days of operations (6AM-10PM local time). This case corresponds to 1,934 departures at CDG set (baseline mean taxi-out time of 12.8 min). Table A.3 shows the departure metering statistics for CDG from the three days of simulation. One can see that the benefits are lower than CLT as expected because of lower levels of congestion.

Table A.2: Error statistics for stochastic simulations of baseline.

Airport	**Number of departures**	**Taxi-out time (min)**			**% of flights**
		Mean	**ME**	**MAE**	**|error| < 5 min**
CDG	14,202	13.3	-0.3	3.0	84.0

Table A.3: Comparison of simulations of departure metering approaches for CDG.

Mean statistics	**ATD-2 logic**	**Optimal control**	**Robust control**
Taxi-out reduction (min)	0.52	0.39	0.53
Hold time (min)	0.61	0.52	0.65
Wheels-off delay (min)	0.09	0.12	0.12
Fraction of flights held	0.26	0.17	0.17
Hold time of flights held	2.36	3.09	3.96

Appendix B

Multi-model smoother for aircraft trajectories

A multi-model smoother is used to estimate the aircraft trajectory from the measurements (FDR/ASDE-X data), primarily to determine the acceleration events. A model for the aircraft dynamics is presented first, followed by the smoothing algorithm.

Aircraft dynamics

The aircraft trajectory on the surface during taxi operations is modeled using the following dynamics [99]:

$$\dot{\bar{\mathbf{x}}} = \bar{\mathbf{f}}(\bar{\mathbf{x}}) + B_w \mathbf{w}, \tag{B.1}$$

$$\text{where } \bar{\mathbf{x}} = \begin{bmatrix} x(t) \\ y(t) \\ V(t) \\ \theta(t) \end{bmatrix}; \quad \bar{\mathbf{f}}(\bar{\mathbf{x}}) = \begin{bmatrix} V\sin(\theta(t)) \\ V\cos(\theta(t)) \\ a(t) \\ \omega(t) \end{bmatrix}; \quad B_w = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}; \quad \mathbf{w} = \begin{bmatrix} w_a \\ w_\theta \end{bmatrix} \tag{B.2}$$

Here, (x, y) is the aircraft position on the ground with respect to a fixed origin, V is the speed, θ is the heading angle, a is the acceleration, ω is the turn rate and $\mathbf{w} \sim \mathcal{N}(0, W)$ is the process noise. The process noise accounts for the uncertainty in the acceleration and turn rate. Here, w_a and w_θ are assumed to be uncorrelated and independent.

The trajectory measurements (either from FDR or ASDE-X) provide full information about the state vector, $\bar{\mathbf{x}}$, that needs to be estimated. Note that the trajectory data contains latitude and longitude coordinates, which are converted into local (x, y) coordinates. The measurement vector ($\bar{\mathbf{z}}$) is given by,

$$\bar{\mathbf{z}} = \bar{\mathbf{x}} + \bar{\mathbf{v}} \tag{B.3}$$

Here, $\bar{\mathbf{v}} \sim \mathcal{N}(0, R)$ is the additive noise, assumed to be independent and uncorrelated with the process noise.

Smoothing algorithm

The state estimation is done using a fixed interval smoother since we need to estimate the state at a particular time instant given the measurements for the entire trajectory [34]. For estimating the state variables, the aircraft dynamics is approximated by a hybrid system of the following form:

$$\dot{\bar{\mathbf{x}}} = \bar{\mathbf{f}}^i(\bar{\mathbf{x}}) + B_w \mathbf{w}; \quad \bar{\mathbf{f}}^i(\bar{\mathbf{x}}) = \begin{bmatrix} V \sin(\theta(t)) \\ V \cos(\theta(t)) \\ a_i \\ \omega_i \end{bmatrix}; \; i \in 1, 2..M \tag{B.4}$$

The model dynamics are specified by M modes, each correspond to a set specified by the acceleration and turn rate (a_i, ω_i). The hybrid state vector consists of the original state variables ($\bar{\mathbf{x}}$) as well as the mode of operation. Multi-model dynamics provides better tracking accuracy compared to a single model dynamics. Another advantage of using multi-model dynamics is that we can identify the ‘most likely mode’ of the system at any instant. Mode identification would allow us to correlate variations in engine fuel flow rate to specific modes or discrete events (such as acceleration events).

The estimation is performed on the discretized version of this system with time discretization equal to the sampling rate of the measurements (1Hz). The smoothing algorithm is based on combining estimates produced by an Interacting Multiple Model (IMM) filter with a backward-time recursion [116]. The algorithm gives out smoothed estimates of the state variables (trajectory) as well as the probability of the system being in a particular mode. The 'most likely mode' of the system at any time instant is determined from the maximum a posteriori probability. The smoothing algorithm consists of two parts: a forward pass using an Interacting Multiple Model (IMM) filter and a Rauch-Tung-Striebel (RTS) backward recursion [116]. The algorithm is briefly discussed below and more details can be found in the original paper [116]. The IMM filter is described first, followed by the backward recursion step.

Forward pass using an IMM filter

The IMM algorithm is used for tracking states in systems that have multiple modes of operation [30]. The main advantage of the IMM algorithm is that it provides good tracking accuracy with a lower computational complexity. We assume the following: (a) the initial state and mode distributions are known, (b) the mode transition is Markovian and transition probabilities (π_{ij}) are known.

Few notations before we proceed further: x_k represents the state vector $\bar{\mathbf{x}}$ at time k. Similarly, m_k^i refers to the system being in the i^{th} mode at k. All the measurements until and including time k are represented by $z_{1:k}$. The probability $(P(m_k^i|z_{1:k}))$ of being in the i^{th} mode at time k conditioned on all the measurements until k is denoted by $\mu_{k|k}^i$. The filtering algorithm proceeds forward in time. The steps at each iteration are as follows:

1. First, the posterior filtering probability density of the state at k conditioned on the mode at $k+1$, $p(x_k|m_{k+1}^i, z_{1:k})$, is computed using filtered state and mode estimates available until k. Here, $p(x_k|m_k^i, z_{1:k}) \approx \mathcal{N}(x_k; \hat{x}_{k|k}^i, P_{k|k}^i)$ and $p(x_k|m_{k+1}^i, z_{1:k}) \approx \mathcal{N}(x_k; \bar{x}_{k|k}^i, \bar{P}_{k|k}^i)$.

The first two moments of $p(x_k|m^i_{k+1}, z_{1:k})$ are obtained using,

$$\bar{\mu}^i_{k+1|k} = \sum_{j \in M} \pi_{ij} \mu^i_{k|k} \quad i = 1,\ 2 ... M \tag{B.5}$$

$$\bar{x}^i_{k|k} = \sum_{j \in M} \pi_{ij} \mu^j_{k|k} \hat{x}^j_{k|k} / \bar{\mu}^i_{k+1|k} \tag{B.6}$$

$$\bar{P}^i_{k|k} = \sum_{j \in M} \pi_{ij} \mu^j_{k|k} \Big[P^j_{k|k} + [\hat{x}^j_{k|k} - \hat{x}_{k|k}][.]^T \Big] / \bar{\mu}^i_{k+1|k} \tag{B.7}$$

2. The second step involves using Unscented Kalman Filter (UKF) to obtain filtered estimates for all the M modes [180]. A prior estimate is obtained by propagating the dynamics and then a posterior estimate is found by using measurements,

$$p(x_k|m^i_{k+1}, z_{1:k}) \xrightarrow{\text{propagate}} p(x_{k+1}|m^i_{k+1}, z_{1:k}) \tag{B.8}$$

$$p(x_{k+1}|m^i_{k+1}, z_{1:k}) \xrightarrow{\text{measurements}} p(x_{k+1}|m^i_{k+1}, z_{1:k+1}) \tag{B.9}$$

3. The mode probabilities are then updated from the innovations of the M Kalman filters using,

$$\mu^i_{k+1|k+1} = c \bar{\mu}^i_{k+1|k} ||Q^i_{k+1}||^{-1/2} \exp^{e^i_{k+1} ||Q^i_{k+1}||^{-1} e^i_{k+1}} \quad i = 1,\ 2 ... M \tag{B.10}$$

here, c is a normalizing constant, $e^i_{k+1} = z_{k+1} - \hat{x}^i_{k+1|k+1}$ and $Q^i_{k+1} = P^i_{k+1|k+1} + R$

4. Finally, the filtered estimate is a weighted sum of mode conditioned state estimates, with the weights being equal to the mode probabilities. The first two moments of the estimate are given by,

$$\hat{x}_{k+1|k+1} = \sum_{j \in M} \mu^j_{k+1|k+1} \hat{x}^j_{k+1|k+1} \tag{B.11}$$

$$P_{k+1|k+1} = \sum_{j \in M} \mu^j_{k+1|k+1} \Big[P^j_{k+1|k+1} + [\hat{x}^j_{k+1|k+1} - \hat{x}_{k+1|k+1}][.]^T \Big] \tag{B.12}$$

Backward recursion using RTS algorithm

The outputs from the filter are used in the backward recursion step to obtain smoothed estimates. The steps for the backward recursion algorithm starting from the final time are as follows:

1. The smoothed mixing density, which is $p(x_k|m^i_{k+1}, z_{1:T})$, at time step k is determined from the smoothed state density at $k+1$ *i.e.* $p(x_{k+1}|m^i_{k+1}, z_{1:T})$. Here, $p(x_k|m^i_{k+1}, z_{1:T}) \approx N(\bar{x}^i_{k|T}, \bar{P}^i_{k|T})$ and $p(x_{k+1}|m^i_{k+1}, z_{1:T}) \approx N(\hat{x}^i_{k+1|T}, P^i_{k+1|T})$. This is obtained using the standard Rauch-Tung-Striebel (RTS) formulae for smoothing [53],

$$\bar{x}^i_{k|T} = \bar{x}^i_{k|k} + G^i_k(\hat{x}^i_{k+1|T} - \hat{x}^i_{k+1|k}) \tag{B.13}$$

$$\bar{P}^i_{k|T} = \bar{P}^i_{k|k} + G^i_k(P^i_{k+1|T} - P^i_{k+1|k}){G^i_k}^T \tag{B.14}$$

$$G^i_k = C^i_{k,k+1}(P^i_{k+1|k})^{-1} \tag{B.15}$$

$$\text{here, } C^i_{k,k+1} = \int (x_k - \bar{x}^i_{k|k})(f^i_k(x_k) - \hat{x}^i_{k+1|k})^T N(x_k; \bar{x}^i_{k|k}, \bar{P}^i_{k|k})dx_k \tag{B.16}$$

The above integral to determine $C^i_{k,k+1}$ is evaluated by using unscented transform [180].

2. The mode conditioned smoothing density is expressed as the sum of distributions conditioned on a sequence of modes over two consecutive sampling periods,

$$p(x_k|m^j_k, z_{1:T}) = \sum_{i \in M} p(x_k|m^j_k, m^i_{k+1}, z_{1:T})P(m^i_{k+1}|m^j_k, z_{1:T}) \tag{B.17}$$

Here, $p(x_k|m^j_k, m^i_{k+1}, z_{1:T})$ ($\approx \mathcal{N}(x_k; \hat{x}^{ji}_{k|T}, P^{ji}_{k|T})$) is obtained using estimates from the IMM filter and the RTS recursion. The first two moments of the distribution are as follows:

$$\hat{x}^{ji}_{k|T} = P^{ji}_{k|T}[(\bar{P}^i_{k|T})^{-1}\bar{x}^i_{k|T} - (\bar{P}^i_{k|k})^{-1}\bar{x}^i_{k|k} + (P^j_{k|k})^{-1}\hat{x}^j_{k|k}] \tag{B.18}$$

$$P^{ji}_{k|T} = [(\bar{P}^i_{k|T})^{-1} - (\bar{P}^i_{k|k})^{-1} + (P^j_{k|k})^{-1}] \tag{B.19}$$

The smoothed mixing probabilities present in the Eq. B.17 is obtained using,

$$\bar{\mu}^{i|j}_{k+1|T} = P(m^i_{k+1}|m^j_k, z_{1:T}) \tag{B.20}$$

$$= \frac{P(m^i_{k+1}|m^j_k)p(z_{k+1:T}|m^j_k, m^i_{k+1}, z_{1:k})}{p(z_{k+1:T}|m^j_k, z_{1:k})} = \frac{\pi_{ji}\Lambda_{ji}}{d_j} \tag{B.21}$$

The likelihood (Λ_{ji}) is approximated using a Gaussian distribution of the following form,

$$\Lambda_{ji} = p(z_{k+1:T}|m^j_k, m^i_{k+1}, z_{1:k}) \approx \mathcal{N}(\delta^{ji}_k; 0, \Sigma^{ji}_k) \tag{B.22}$$

$$\text{where, } \delta^{ji}_k = \hat{x}^{b,i}_{k|k+1} - \hat{x}^j_{k|k};\ \Sigma^{ji}_k = P^{b,i}_{k|k+1} + P^j_{k|k} \tag{B.23}$$

The normalizing constant d_j is given by,

$$d_j = p(z_{k+1:T}|m^j_k, z_{1:k}) = \sum_{i\in\mathcal{M}} \pi_{ji}\Lambda_{ji} \tag{B.24}$$

The smoothed mode conditioned density is then obtained using,

$$p(x_k|m^j_k, z_{1:T}) \approx \mathcal{N}(x_k; \hat{x}^j_{k|T}, P^j_{k|T}) \tag{B.25}$$

$$\hat{x}^j_{k|T} = \sum_{i\in\mathcal{M}} \bar{\mu}^{i|j}_{k+1|T}\hat{x}^{ji}_{k|T} \tag{B.26}$$

$$P^j_{k|T} = \sum_{i\in\mathcal{M}} \bar{\mu}^{i|j}_{k+1|T}[P^{ji}_{k|T} + (\hat{x}^{ji}_{k|T} - \hat{x}^j_{k|T})(\hat{x}^{ji}_{k|T} - \hat{x}^j_{k|T})^T] \tag{B.27}$$

3. Finally, the output of the smoother are the state estimates ($p(x_k|z_{1:T})$) and the mode probabilities ($P(m^j_k|z_{1:T})$). The mode probability is given by,

$$\mu^j_{k|T} = P(m^j_k|z_{1:T}) \tag{B.28}$$

$$= \frac{p(z_{k+1:T}|m^j_k, z_{1:k})P(m^j_k|z_{1:k})}{\sum_{j\in\mathcal{M}} p(z_{k+1:T}|m^j_k, z_{1:k})P(m^j_k|z_{1:k})} = \frac{d_j\mu^j_{k|k}}{\sum_{j\in\mathcal{M}} d_j\mu^j_{k|k}} \tag{B.29}$$

The state estimates are found using,

$$p(x_k|z_{1:T}) \approx \mathcal{N}(x_k; \hat{x}_{k|T}, P_{k|T}) \tag{B.30}$$

$$\hat{x}_{k|T} = \sum_{j \in \mathcal{M}} \mu^j_{k|T} \hat{x}^j_{k|T} \tag{B.31}$$

$$P_{k|T} = \sum_{j \in \mathcal{M}} \mu^j_{k|T} [P^j_{k|T} + (\hat{x}^j_{k|T} - \hat{x}_{k|T})(\hat{x}^j_{k|T} - \hat{x}_{k|T})^T] \tag{B.32}$$

The mode detection is based on maximum a posteriori probability. The most likely mode ($\hat{j}_k$) at time step k is computed as follows,

$$\hat{j}_k = \arg\max_{j \in \mathcal{M}} \mu^j_{k|T} \tag{B.33}$$

Parameter values

For the analysis, we consider the process noise and measurement noise covariance matrix to be a diagonal with the following components for FDR data: $W_{11} = 0.04\ m^2 s^{-4},\ W_{22} = 36\ deg^2 s^{-2},\ R_{11} = R_{22} = 25\ m^2,\ R_{33} = 0.048\ m^2 s^{-2},\ R_{44} = 1\ deg^2$. We consider the same process noise, but consider a slightly different measurement noise for the ASDE-X data: $R_{11} = R_{22} = 16\ m^2,\ R_{33} = 0.0625\ m^2 s^{-2},\ R_{44} = 16\ deg^2$. The mode transition probability matrix for the smoothing algorithm is chosen such that there is a higher probability for remaining in the same mode and we consider the following values: $\pi_{ii} = 0.5$, $\pi_{ij,i \neq j} = 0.25$.

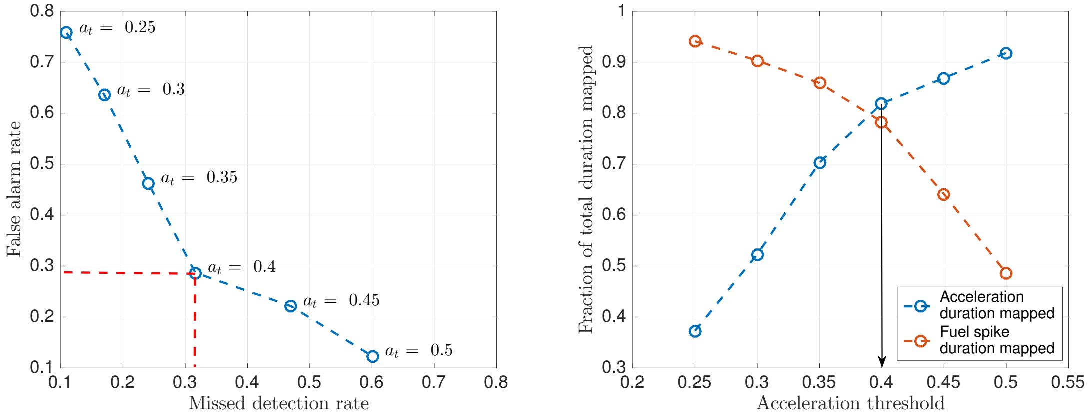
Optimal acceleration threshold

Analysis for FDR data

We first define and present a few event detection statistics that enable us to pick an optimal acceleration threshold considering the entire FDR dataset. The total taxi-out time considering all flights in the dataset is 2,406 min, containing 411 fuel burn events. The acceleration events are determined using the trajectories and they are mapped to the fuel burn events.

An acceleration event is said to be mapped to a fuel burn event if the difference between the start time of the acceleration event and the fuel burn event is less than 30 s. A 30 s duration was chosen to account for the time lag between the fuel flow rate spike and acceleration. The mapping between acceleration events and the fuel burn events is quantified using missed detection rate and false alarm rate. Missed detection rate is defined as the fraction of unmapped fuel burn events to the total fuel burn events. This represents the probability of a fuel burn event not corresponding to an acceleration event. False alarm rate is defined as the fraction of the unmapped acceleration events to the total acceleration events. This represents the probability of an acceleration event not corresponding to a fuel burn event. For the fuel burn events to correlate well with the acceleration events, we need to have low values of missed detection rate and false alarm rate.

The impact of acceleration threshold on event detection is presented in Table B.1. For low threshold values, such as 0.25 ms^{-2}, the number of acceleration events detected is significantly higher than the number fuel burn events (spikes). Most of the fuel burn spikes can be mapped to one of the acceleration events but many of the acceleration events do not correspond to a fuel burn spike. This leads to a low missed detection rate and a high false alarm rate. For higher acceleration thresholds, fewer instances are classified as an acceleration event. This results in a drop in false alarm rate, but the number of unmapped fuel burn events increases, leading to an increase in the missed detection rate. Fig. B-1(a) shows the trade-off between missed detection rate and false alarm rate which is obtained with different acceleration thresholds. The missed detection rate increases as the false alarm rate decreases. A threshold of 0.4 ms^{-2} can be considered as the optimal value since both missed detection and false alarm rates are equally low. Fig. B-1(b) shows the fraction of the total duration of acceleration that is mapped to fuel burn events (F_a) and also the fraction of total duration of fuel burn spikes that is mapped to acceleration events (F_b). For low thresholds, only a small fraction of the acceleration events correspond to fuel burn events, resulting in a low F_a. On the other hand, most of the fuel burn events are mapped to acceleration events, yielding a high F_b. As the threshold is increased, the fraction of the total acceleration duration that is mapped increases whereas the fraction of the total fuel burn spike duration that is mapped



(a) Missed detection rate vs false alarm rate for different acceleration thresholds

(b) Fraction of event duration mapped

Figure B-1: Analysis with different acceleration thresholds

decreases. At the optimal threshold of 0.4 ms^{-2}, 82% of the total duration of acceleration is mapped to fuel burn events and 78% of the total fuel burn spike duration is mapped to the acceleration events.

Table B.1: Event detection statistics for different acceleration thresholds (a_t) using FDR data

Thres. (a_t, ms^{-2})	Acc. events	Acc. events mapped to fuel burn events	Unmapped acceleration events	Unmapped fuel burn events	Missed detection rate	False alarm rate	Fraction of acceleration duration mapped	Fraction of fuel spike duration mapped
0.25	1507	366	1141	45	0.11	0.76	0.37	0.94
0.30	934	341	593	70	0.17	0.63	0.52	0.90
0.35	580	312	268	99	0.24	0.46	0.70	0.86
0.40	394	281	113	130	0.32	0.29	0.82	0.78
0.45	280	218	62	193	0.47	0.22	0.87	0.64
0.50	187	164	23	247	0.60	0.12	0.92	0.49

Analysis for ASDE-X data

We present the ASDE-X analysis using the data from 104 A330 flights departing from the US airports. The total taxi-out time for this subset of flights is 1,226 min, containing 216 fuel burn events and 200 acceleration events. The fuel burn and acceleration events mentioned above are obtained from the FDR data for these flights using the optimal acceleration threshold. The acceleration events detected using the ASDE-X data are then mapped with the acceleration events from the FDR data. Table B.2 shows the acceleration detection statistics with different acceleration thresholds for ASDE-X data. The number of acceleration events is high for low thresholds and decreases as the acceleration threshold is increased, similar to the detection statistics that we had seen earlier with the FDR data. As the threshold is increased, the number of unmapped acceleration events detected using the ASDE-X data decreases, while the number of unmapped acceleration events detected with the FDR increases. Therefore, the missed detection rate, defined here as the fraction of unmapped acceleration events obtained with FDR data, increases with increase in acceleration threshold. On the other hand, the false alarm rate, defined here as the fraction of the unmapped acceleration events obtained with ASDE-X, reduces with increase in threshold. We notice that beyond a threshold of 0.5 ms^{-2}, the drop in false alarm rate is small compared to the increase in missed detection rate. This difference can also be observed in the fraction of acceleration duration mapped using the FDR and ASDE-X. Hence we consider 0.5 ms^{-2} as the optimal acceleration threshold for ASDE-X. The last column in the table shows the fraction of the duration of acceleration in ASDE-X to that in FDR. We notice that for a threshold of 0.5 ms^{-2}, the fraction is close to one. The fraction being close to one is desirable because the fuel burn model relies on the duration of acceleration as one of the predictors, so we need the duration of acceleration picked up with the ASDE-X data to be same as that obtained from the FDR data. Given that ASDE-X data is more noisy, one would have expected the optimal acceleration threshold for the ASDE-X to be slightly higher than that for FDR to reduce the false alarm rates.

Fig. B-2 shows a comparison of the ASDE-X and FDR estimates obtained from the smoother using the optimal acceleration threshold for a sample trajectory. We see that the speed profile from the ASDE-X estimate closely matches the FDR estimate. For this particular trajectory, three acceleration events are detected using the FDR data, out of which, only two of them are captured by the ASDE-X estimate. The missed detection rate and false alarm rate statistics shown earlier can be used to quantify these errors over the all

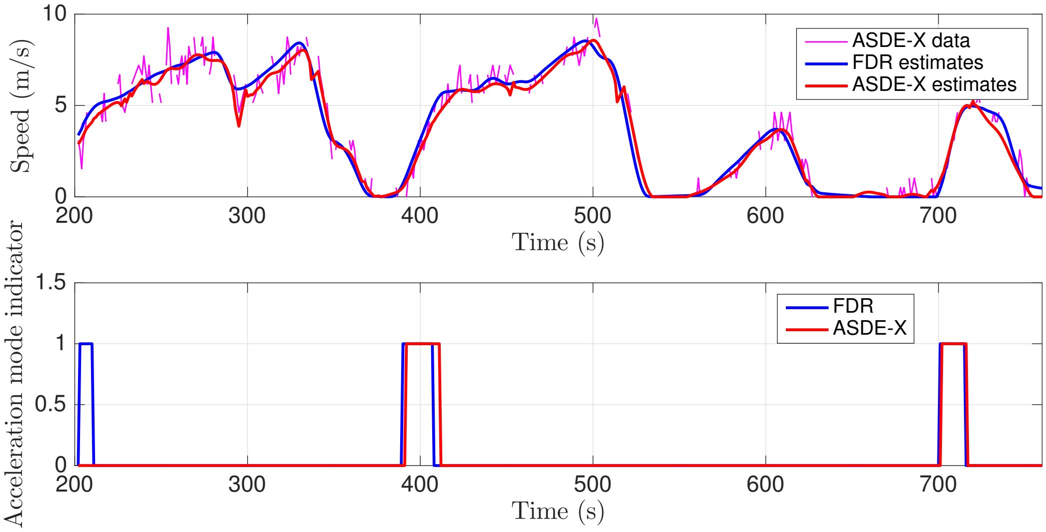


Figure B-2: Comparison between FDR and ASDE-X estimates for a sample trajectory.

the flights in our dataset.

Table B.2: Event detection statistics for different acceleration thresholds (a_t) using ASDE-X data

Thres. (ms^{-2})	Acc. events from ASDE-X	Mapped acc. events	Unmapped acc. events in FDR	Unmapped acc. events in ASDE-X	Missed detection rate	False alarm rate	Fraction of acc. duration mapped in ASDE-X	Fraction of acc. duration mapped in FDR	Acc. duration in ASDE-X/FDR
0.25	1294	194	6	1100	0.03	0.85	0.96	0.27	0.18
0.30	1029	195	5	834	0.03	0.81	0.98	0.34	0.25
0.35	757	191	9	566	0.05	0.75	0.98	0.42	0.37
0.40	538	187	13	351	0.07	0.65	0.97	0.51	0.53
0.45	376	170	30	206	0.15	0.55	0.91	0.57	0.76
0.50	289	159	41	130	0.21	0.45	0.89	0.62	1.05
0.55	216	132	68	84	0.34	0.39	0.78	0.63	1.46
0.60	161	98	102	63	0.51	0.39	0.58	0.60	1.98

Appendix C

Automatic speech recognition model

The automatic speech recognition model is based on Deep Speech [80], an end-to-end speech recognition model. We use Mozilla's implementation of Deep Speech for our analysis [132]. We briefly describe the model in this section; more detailed information can be found in the original papers [80, 132].

Model overview

Figure C-1 illustrates the model architecture of the speech recognition system. The main components are the feature extraction module, acoustic model, language model, and decoding module. The feature extraction module takes the ATC audio signal as input, and outputs coefficients associated with its frequency spectrum as follows: the entire time series of the audio signal is divided into smaller (32 ms) time-windows with a 20 ms overlap; each time-window is associated with a feature vector that corresponds to its Mel Frequency Cepstral Coefficients (MFCCs) [80]. The MFCCs of each time-window constitute the feature vector that serves as the input to the acoustic model. The acoustic model is a recurrent neural network that is trained to output a sequence of character probabilities based on the sequence of input feature vectors. The characters here correspond to letters of the English alphabet, apostrophe, space, blank, and additional identifiers corresponding to foreign words or unintelligible words. The language model outputs the probability of a sequence of characters based on the training text data, independent of the audio signal. In the decoding module, the output of the acoustic model is integrated with that of the language model to determine the transcribed text.

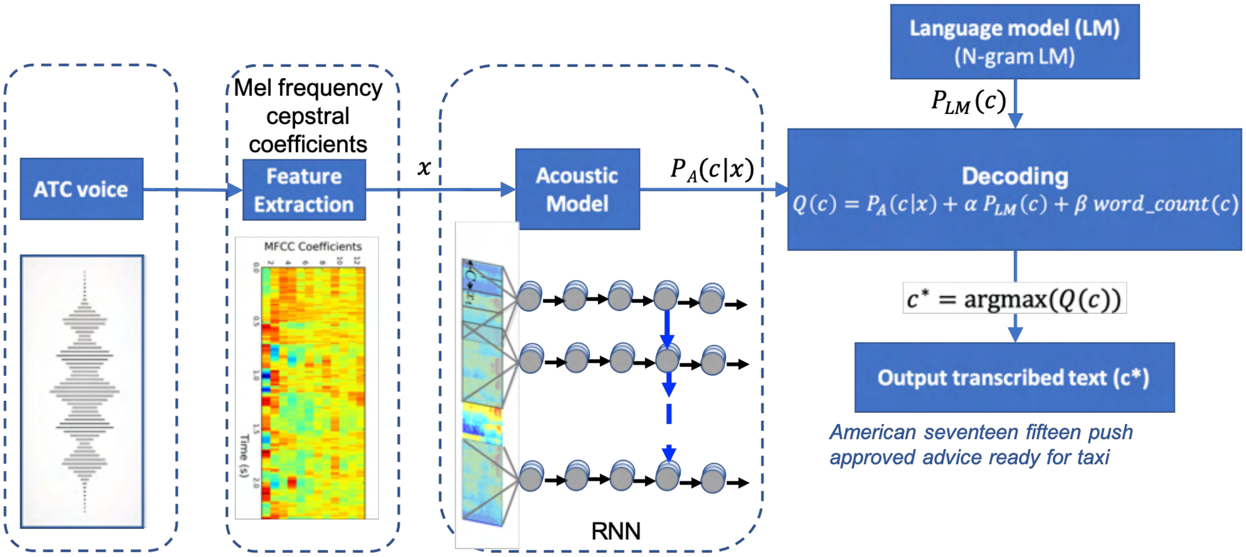


Figure C-1: Model architecture for the ASR system.

Acoustic model

The acoustic model is a recurrent neural network composed of five hidden layers, the first three of which are non-recurrent. The first layer takes the MFCC input for a particular time-window as well as a few (9, in our case) context frames on either side of the window. A clipped rectified-linear unit (ReLU) activation function is used for all the layers, except the last one. The model architecture of Mozilla's implementation differs slightly from the original Deep Speech paper [132, 80]. Here, the fourth layer is a feed forward recurrent layer instead of a bidirectional recurrent layer, to reduce computational time during inference. The last layer is a non-recurrent output layer that yields the character probabilities based on a Softmax function. A Connectionist Temporal Classification (CTC) loss function is used to compute the prediction error during training [98]. We use an adaptive learning rate (Adam method) for updating the model parameters through stochastic gradient descent during training, and a dropout rate of 5% for regularization.

Language model

Training the acoustic model for the ATC domain to produce accurate character level transcription is challenging due to the limited availability of transcribed ATC audio. Furthermore, the neural network can output phonetically similar word renderings which can be incorrect (e.g., "bostin" instead of "boston"). These issues can be addressed using a language model that is tailored to the ATC domain. The language model is a probability distribution over a sequence of words, which is used in the decoding stage. We choose an N-gram language model because it can be trained using existing libraries (KenLM [83]) and yields good performance. In this language model, the probability of the k^{th} word is assumed to depend only on the $N-1$ preceding words, i.e., $P(w_k|w_{k-1}, w_{k-2}, .., w_1) = P(w_k|w_{k-1}, w_{k-2}, .., w_{k-(N-1)})$. Consequently, the probability of a sequence of words, $P(w_1, w_2, ...w_m)$, can be expressed as a product of conditional probabilities.

The conditional probabilities required for the model are determined from the ATC audio transcripts in the training data. Although a higher value of N will lead to better predictions, there may not be sufficient data to obtain consistent statistics for a larger N. An optimal value of N is determined using a parametric analysis to yield a lower word error rate on the validation set.

Decoding step

The decoding step determines the most probable sequence of characters given the output probabilities from the acoustic model and language model. Let $P_A(\mathbf{c}|\mathbf{x})$ represent the probability of a sequence of characters, $\mathbf{c} = \{c_1, c_2, c_3..\}$, obtained from the acoustic model for a given input audio ($\mathbf{x}$). Similarly, let $P_{LM}(\mathbf{c})$ represent the probability of a sequence of characters obtained from the language model. The objective of the decoding step is to obtain a sequence of characters that maximizes the confidence score ($Q(\mathbf{c})$):

$$Q(\mathbf{c}) = P_A(\mathbf{c}|\mathbf{x}) + \alpha P_{LM}(\mathbf{c}) + \beta \text{ word count}(\mathbf{c}). \tag{C.1}$$

Here, α and β are weights to balance the influence of the acoustic model, the language model, and the word count of the utterance. The last term is used because shorter sentences inherently have a higher probability in the N-gram language model. The output sequence of characters that maximizes the confidence score, $Q()$, is determined using a beam search algorithm [80]. Optimal values of α, β and beam width are determined using a parametric analysis such that the average WER over a validation set is minimized.

Bibliography

[1] `https://www.liveatc.net`, 2019. Retrieved Aug 21, 2019.

[2] Naoto Abe and Kazuo Yamanaka. Smith predictor control and internal model control-a tutorial. In *SICE 2003 Annual Conference*, volume 2, pages 1383–1387. IEEE, 2003.

[3] Air Transport Action Group. Facts and Figures. `https://www.atag.org/facts-figures.html`, 2019. Retrieved on April 15th, 2021.

[4] Airbus. Global Market Forecast - Global networks, Global citizens. `https://www.airbus.com`, 2018. Retrieved on July 27th, 2019.

[5] United Airlines. United to Work with Archer Aviation. `https://hub.united.com/`, 2021. Retrieved Feb 19, 2021.

[6] Airlines for America. U.S. Passenger Carrier Delay Costs. `http://airlines.org/dataset/per-minute-cost-of-delays-to-u-s-airlines/`, 2018. Retrieved July 22, 2019.

[7] Matthias Althoff. *Reachability analysis and its application to the safety assessment of autonomous cars*. PhD thesis, Technische Universität München, 2010.

[8] Matthias Althoff, Olaf Stursberg, and Martin Buss. Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization. In *2008 47th IEEE Conference on Decision and Control*, pages 4042–4048. IEEE, 2008.

[9] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*, pages 173–182, 2016.

[10] Anon. Reservoir Level Storage Bulletin. `http://cwc.gov.in/reservoir-level-storage-bulletin`, 2019. Retrieved on Mar 3rd, 2021.

[11] Anon. Greenfield Airports and Urban Growth: What Drives Success? `https://airporturbanism.com/articles/greenfield-airports-and-urban-growth-what-drives-success`, May 25th 2018. Retrieved on Feb 27th, 2021.

[12] Anon. Addressing Future Capacity Needs in the U.S. Aviation System. `https://www.ustravel.org/sites/default/files/media_root/USTravel_Eno_1.pdf`, Nov 2013. Retrieved on Feb 27th, 2021.

[13] Bimal Aponso, Richard Coppenbarger, Yoon Jung, Leighton Quon, Gary Lohr, and Neil O'Connor. Identifying key issues and potential solutions for integrated arrival, departure, surface operations by surveying stakeholder preferences. In *15th AIAA Aviation Technology, Integration, and Operations Conference*, 2015.

[14] Murat Arcak and John Maidens. Simulation-based reachability analysis for nonlinear systems using componentwise contraction properties. In *Principles of Modeling*, pages 61–76. Springer, 2018.

[15] Louise Butcher Elena Ares, Henry Midgley, and Louise Smith. Heathrow expansion. In *Briefing paper*. House of Commons Library, 2018.

[16] Akshay Ashok. *Reducing the air quality impacts of aircraft activity at airports*. PhD thesis, Massachusetts Institute of Technology, 2016.

[17] Stephen C Atkins. Estimating departure queues to study runway efficiency. *Journal of guidance, control, and dynamics*, 25(4):651–657, 2002.

[18] Sandeep Badrinath and Hamsa Balakrishnan. Control of a non-stationary tandem queue model of the airport surface. In *American Control Conference (ACC), 2017*, pages 655–661. IEEE, 2017.

[19] Sandeep Badrinath and Hamsa Balakrishnan. Automatic speech recognition for air traffic control communications. *Transportation Research Record (under review)*, 2021.

[20] Sandeep Badrinath, Hamsa Balakrishnan, Emily Clemons, and Tom Reynolds. Evaluating the impact of uncertainty on airport surface operations. In *2018 Aviation Technology, Integration, and Operations Conference*, page 4242, 2018.

[21] Sandeep Badrinath, Hamsa Balakrishnan, Emily Joback, and Tom G Reynolds. Impact of off-block time uncertainty on the control of airport surface operations. *Transportation Science*, 54(4):920–943, 2020.

[22] Sandeep Badrinath, Hamsa Balakrishnan, Ji Ma, and Daniel Delahaye. A Comparative Analysis of Departure Metering at Paris (CDG) and Charlotte (CLT) Airports. In *13th USA/Europe Air Traffic Management Research and Development Seminar, Vienna*, 2019.

[23] Sandeep Badrinath, Emily Joback, James Abel, Hamsa Balakrishnan, and Tom Reynolds. Spatial modeling of airport surface fuel burn for environmental impact analyses. In *Submitted to the 14th USA/Europe Air Traffic Management Research and Development Seminar*, 2021.

[24] Sandeep Badrinath, Max Z Li, and Hamsa Balakrishnan. Integrated surface–airspace model of airport departures. *Journal of Guidance, Control, and Dynamics*, 42(5):1049–1063, 2018.

[25] Hamsa Balakrishnan and Bala G Chandran. Algorithms for scheduling runway operations under constrained position shifting. *Operations Research*, 58(6):1650–1665, 2010.

[26] Michael O Ball, Robert L Hoffman, Dave Knorr, James Wetherly, and Mike Wambsganss. Assessing the benefits of collaborative decision making in air traffic management. *Progress in Astronautics and Aeronautics*, 193:239–252, 2001.

[27] MO Ball, Thomas Vossen, and Robert Hoffman. Analysis of demand uncertainty effects in ground delay programs. In *4th USA/Europe air traffic management R&D seminar*, pages 51–60, 2001.

[28] Chaithanya Bandi, Dimitris Bertsimas, and Nataly Youssef. Robust queueing theory. *Operations Research*, 63(3):676–700, 2015.

[29] Dimitris Bertsimas, Michael Frankovich, and Amedeo Odoni. Optimal selection of airport runway configurations. *Operations Research*, 59(6):1407–1419, 2011.

[30] Henk AP Blom and Yaakov Bar-Shalom. The interacting multiple model algorithm for systems with markovian switching coefficients. *IEEE transactions on Automatic Control*, 33(8):780–783, 1988.

[31] Christabelle S Bosson and Dengfeng Sun. Optimization of airport surface operations under uncertainty. *Journal of Air Transportation*, pages 84–92, 2016.

[32] K Bouyoucef and K Khorasani. Robust feedback linearization-based congestion control using a fluid flow model. In *American Control Conference, 2006*. IEEE, 2006.

[33] EG Bowen and T Pearcey. Delays in the flow of air traffic. *The Aeronautical Journal*, 52(448):251–258, 1948.

[34] Mark Briers, Arnaud Doucet, and Simon Maskell. Smoothing algorithms for state–space models. *Annals of the Institute of Statistical Mathematics*, 62(1):61–89, 2010.

[35] Pierrick Burgain, Olivia J Pinon, Eric Feron, John-Paul Clarke, and Dimitri N Mavris. Optimizing pushback decisions to valuate airport surface surveillance information. *IEEE Transactions on Intelligent Transportation Systems*, 13(1):180–192, 2012.

[36] Viggo Butler and Robert W Poole. *Increasing airport capacity without increasing airport size*. Reason Foundation, 2009.

[37] A Castelletti, Stefano Galelli, Marcello Restelli, and Rodolfo Soncini-Sessa. Tree-based reinforcement learning for optimal water reservoir operation. *Water Resources Research*, 46(9), 2010.

[38] Cristiano Cervellera, Victoria CP Chen, and Aihong Wen. Optimization of a large-scale water reservoir network by stochastic dynamic programming with efficient state space discretization. *European journal of operational research*, 171(3):1139–1151, 2006.

[39] Changi Airport. Changi Airport A-CDM Handbook. `http://changiairport-cdm.sg/files/Changi\%20A-CDM\%20Handbook\%20v2.pdf`, 2018. Retrieved Dec 12, 2018.

[40] Yashovardhan Sushil Chati. *Statistical modeling of aircraft engine fuel burn*. PhD thesis, Massachusetts Institute of Technology, 2018.

[41] Qiang Chen and Oliver WW Yang. Robust controller design for aqm router. *IEEE Transactions on Automatic Control*, 52(5):938–943, 2007.

[42] Shuo Chen, HD Kopald, R Chong, Y Wei, and Zachary Levonian. Read back error detection using automatic speech recognition. In *12th USA/Europe Air Traffic Management Research and Development Seminar (ATM2017), Seattle, WA, USA*, 2017.

[43] Francesco Chiti, Romano Fantacci, and Andrea Rizzo. An integrated software platform for airport queues prediction with application to resources management. *Journal of Air Transport Management*, 67:11–18, 2018.

[44] Darren M Chitty, Mario Gongora, and Shengxiang Yang. Evolutionary dynamic optimisation of airport security lane schedules. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8. IEEE, 2016.

[45] Linsen Chong and Carolina Osorio. A simulation-based optimization algorithm for dynamic large-scale urban transportation problems. *Transportation Science*, 52(3):637–656, 2017.

[46] Civil Aviation Administration of China. `http://www.caac.gov.cn/`. Retrieved Sept 8, 2019.

[47] Samuel Coogan. Mixed monotonicity for reachability and safety in dynamical systems. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 5074–5085. IEEE, 2020.

[48] Samuel Coogan and Murat Arcak. Efficient finite abstraction of mixed monotone systems. In *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, pages 58–67. ACM, 2015.

[49] Rich Coppenbarger, Yoon Jung, Tom Kozon, Amir Farrahi, Waqar Malik, Hanbong Lee, Eric Chevalley, and Matt Kistler. Benefit opportunities for integrated surface and airspace departure scheduling: a study of operations at charlotte-douglas international airport. In *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, pages 1–10. IEEE, 2016.

[50] José Manuel Cordero, Natalia Rodríguez, José Miguel de Pablo, and Manuel Dorado. Automated speech recognition in controller communications applied to workload measurement. *3rd SESAR Innovation Days, Stockholm, Sweden*, 2013.

[51] Martin Corless and George Leitmann. Continuous state feedback guaranteeing uniform ultimate boundedness for uncertain dynamic systems. *IEEE Transactions on Automatic Control*, 26(5):1139–1144, 1981.

[52] W. J. Coupe, L. Bagasol, L. Chen, H. Lee, and Y. C. Jung. A data-driven analysis of a tactical surface scheduler. In *2018 Aviation Technology, Integration, and Operations Conference*. AIAA, 2018.

[53] John L Crassidis and John L Junkins. *Optimal estimation of dynamic systems*. CRC press, 2011.

[54] Christophe Cros and Carsten Frings. Alternative taxiing means–engines stopped. In *Airbus workshop on Alternative taxiing means–Engines stopped*, 2008.

[55] JG Dai and Mark Gluzman. Queueing network controls via deep reinforcement learning. *arXiv preprint arXiv:2008.01644*, 2020.

[56] Estelle Delpech, Marion Laignelet, Christophe Pimm, Céline Raynal, Michal Trzos, Alexandre Arnold, and Dominique Pronto. A real-life, french-accented corpus of air traffic control communications. In *Language Resources and Evaluation Conference (LREC)*, 2018.

[57] Tony Diana. An evaluation of the impact of wake vortex re-categorization: The case of Charlotte Douglas International airport (CLT). *Transportation Research Part A: Policy and Practice*, 109:41–49, 2018.

[58] Agner Krarup Erlang. The theory of probabilities and telephone conversations. *Nyt. Tidsskr. Mat. Ser. B*, 20:33–39, 1909.

[59] Eurocontrol. Airport CDM implementation manual, 2017.

[60] EUROCONTROL Airport CDM Team. Airport CDM Implementation–The Manual, 2017. https://www.eurocontrol.int/sites/default/files/publication/files/airport-cdm-manual-2017.PDF, Retrieved Dec 10, 2018.

[61] C Evertse and HG Visser. Real-time airport surface movement planning: Minimizing aircraft emissions. *Transportation Research Part C: Emerging Technologies*, 79:224–241, 2017.

[62] FAA. TFDM Implementation Map and Timeline. `https://www.faa.gov/air_traffic/technology/tfdm/implementation/`, 2020. Retrieved Dec 30, 2020.

[63] FAA Surface CDM Team. US Airport Surface Collaborative Decision Making (CDM) Concept of Operations (ConOps) in the Near-Term: Application of Surface CDM at United States Airports, June 2012.

[64] Chuchu Fan, James Kapinski, Xiaoqing Jin, and Sayan Mitra. Simulation-driven reachability using matrix measures. *ACM Transactions on Embedded Computing Systems (TECS)*, 17(1):21, 2018.

[65] Chuchu Fan, Bolun Qi, Sayan Mitra, Mahesh Viswanathan, and Parasara Sridhar Duggirala. Automatic reachability analysis for nonlinear hybrid models with c2e2. In *International Conference on Computer Aided Verification*, pages 531–538. Springer, 2016.

[66] Federal Aviation Administration. Aviation Environmental Design Tool (AEDT), 2017. https://aedt.faa.gov.

[67] Federal Aviation Administration. Airport Surface Detection Equipment, Model X (ASDE-X). https://www.faa.gov, 2018. Retrieved Jan 1, 2019.

[68] Federal Aviation Administration. Aviation System Performance Metrics (ASPM). `http://aspm.faa.gov/`, 2018. Retrieved Oct 18, 2018.

[69] Federal Aviation Administration. TFDM Overview, 2018. https://www.faa.gov/.

[70] Wu-chang Feng, Kang G Shin, Dilip D Kandlur, and Debanjan Saha. The blue active queue management algorithms. *IEEE/ACM Transactions on Networking (ToN)*, 10(4):513–528, 2002.

[71] Eric R Feron, R John Hansman, Amadeo R Odoni, Ruben B Cots, Bertrand Delcaire, William D Hall, Husni R Idris, Alp Muharremoglu, and N Pujet. The departure planner: A conceptual discussion. 1997.

[72] Sally Floyd and Van Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on networking*, (4):397–413, 1993.

[73] Emilia Fridman. Tutorial on lyapunov-based methods for time-delay systems. *European Journal of Control*, 20(6):271–283, 2014.

[74] Giulio Garbi, Emilio Incerto, and Mirco Tribastone. Learning queuing networks by recurrent neural networks. In *Proceedings of the ACM/SPEC International Conference on Performance Engineering*, pages 56–66, 2020.

[75] Jean Gregoire, Xiangjun Qian, Emilio Frazzoli, Arnaud De La Fortelle, and Tichakorn Wongpiromsarn. Capacity-aware backpressure traffic signal control. *IEEE Transactions on Control of Network Systems*, 2(2):164–173, 2014.

[76] Jean Gregoire, Xiangjun Qian, Emilio Frazzoli, Arnaud De La Fortelle, and Tichakorn Wongpiromsarn. Capacity-aware backpressure traffic signal control. *IEEE Transactions on Control of Network Systems*, 2(2):164–173, 2015.

[77] K. Gu, V. Kharitonov, and J. Chen. *Stability of Time-delay Systems*. Birkhauser, Boston, 2003.

[78] SV Gudmundsson, M Cattaneo, and R Redondi. Forecasting temporal world recovery in air transport markets in the presence of large economic shocks: The case of covid-19. *Journal of Air Transport Management*, 91:102007, 2021.

[79] Yves Günther, Anthony Inard, Bernd Werther, Marc Bonnier, Gunnar Spies, Alan Marsden, Marco Temme, Dietmar Böhme, Roger Lane, and Helmut Niederstraßer. Total Airport Management (operational concept and logical architecture), 2006.

[80] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.

[81] Mark Hansen. Micro-level analysis of airport delay externalities using deterministic queuing models: a case study. *Journal of Air Transport Management*, 8(2):73–87, 2002.

[82] Mark Hansen, Tasos Nikoleris, David Lovell, Kleoniki Vlachou, and Amedeo Odoni. Use of queuing models to estimate delay savings from 4d trajectory precision. In *Eighth USA/Europe Air Traffic Management Research and Development Seminar*, 2009.

[83] Kenneth Heafield. Kenlm: Faster and smaller language model queries. In *Proceedings of the sixth workshop on statistical machine translation*, pages 187–197. Association for Computational Linguistics, 2011.

[84] Andreas Heidt, Hartmut Helmke, Manu Kapolke, Frauke Liers, and Alexander Martin. Robust runway scheduling under uncertain conditions. *Journal of Air Transport Management*, 56:28–37, 2016.

[85] Hartmut Helmke, Jürgen Rataj, Thorsten Mühlhausen, Oliver Ohneiser, Heiko Ehr, Matthias Kleinert, Y Oualil, Marc Schulder, and D Klakow. Assistant-based speech recognition for atm applications. In *11th USA/Europe Air Traffic Management Research and Development Seminar (ATM2015), Lisbon, Portugal*, 2015.

[86] Christopher V Hollot, Vishal Misra, Donald Towsley, and Weibo Gong. Analysis and design of controllers for aqm routers supporting tcp flows. *IEEE Transactions on automatic control*, 47(6):945–959, 2002.

[87] IATA. Jet Fuel Price Monitor. `https://www.iata.org/publications/economics/fuel-monitor/Pages/index.aspx`, 2018. Retrieved Dec 14, 2018.

[88] ICAO. ICAO Engine Emissions Databank. `https://www.easa.europa.eu/easa-and-you/environment/icao-aircraft-engine-emissions-databank`, 2018. Retrieved Dec 14, 2018.

[89] ICAO. The World of Air Transport in 2019. `https://www.icao.int/annual-report-2019/Pages/the-world-of-air-transport-in-2019.aspx`, 2019. Retrieved Dec 30, 2020.

[90] Husni Idris, Ni Shen, Aditya Saraf, Jason Bertino, and Shannon Zelinski. Comparison of different control schemes for strategic departure metering. In *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, pages 1–13. IEEE, 2016.

[91] Airports Council International. 2016 aircraft movements. `https://aci.aero/data-centre/annual-traffic-data/aircraft-movements/2016-final-summary/`, 2016.

[92] Eri Itoh and Mihaela Mitici. Analyzing tactical control strategies for aircraft arrivals at an airport using a queuing model. *Journal of Air Transport Management*, 89:101938, 2020.

[93] Alexandre Jacquillat and Amedeo R Odoni. Endogenous control of service rates in stochastic and dynamic queuing models of airport congestion. *Transportation Research Part E: Logistics and Transportation Review*, 73:133–151, 2015.

[94] Frédéric M Jordan, Jean-Louis Boillat, and Anton J Schleiss. Optimization of the flood protection effect of a hydropower multi-reservoir system. *International journal of river basin management*, 10(1):65–72, 2012.

[95] Yoon Jung, Shawn Engelland, Richard Capps, Rich Coppenbarger, Becky Hooey, Shivanjli Sharma, Lindsay Stevens, Savita Verma, Gary Lohr, Eric Chevalley, et al. Airspace Technology Demonstration 2 (ATD-2) Phase 1 Concept of Use (ConUse), 2018.

[96] Ghasem Kahe, Amir Hossein Jahangir, and Behrouz Ebrahimi. Aqm controller design for tcp networks based on a new control strategy. *Telecommunication Systems*, 57(4):295–311, 2014.

[97] Iasson Karafyllis and Miroslav Krstic. *Predictor feedback for delay systems: Implementations and approximations*. Springer, 2017.

[98] Kazuya Kawakami. Supervised sequence labelling with recurrent neural networks. *Ph. D. dissertation, PhD thesis. Ph. D. thesis*, 2008.

[99] Harshad Khadilkar and Hamsa Balakrishnan. A multi-modal unscented Kalman filter for inference of aircraft position and taxi mode from surface surveillance data. In *Aviation Technology, Integration and Operations, AIAA Conference on, Virginia Beach, VA*, 2011.

[100] Harshad Khadilkar and Hamsa Balakrishnan. Estimation of aircraft taxi fuel burn using flight data recorder archives. *Transportation Research Part D: Transport and Environment*, 17(7):532–537, 2012.

[101] Harshad Khadilkar and Hamsa Balakrishnan. Network congestion control of airport surface operations. *Journal of Guidance, Control, and Dynamics*, 37(3):933–940, 2014.

[102] Harshad Dilip Khadilkar. *Networked control of aircraft operations at airports and in terminal areas*. PhD thesis, Massachusetts Institute of Technology, 2013.

[103] Sang Hyun Kim and Eric Feron. Impact of gate assignment on departure metering. *IEEE Transactions on Intelligent Transportation Systems*, 15(2):699–709, 2013.

[104] Toshikazu Kimura. Diffusion approximation for an m/g/m queue. *Operations Research*, 31(2):304–321, 1983.

[105] JFC Kingman. The first erlang century—and the next. *Queueing Systems*, 63(1-4):3, 2009.

[106] Peeter Andrus Kivestu. *Alternative methods of investigating the time dependent M/G/k queue*. PhD thesis, Massachusetts Institute of Technology, 1976.

[107] Dag Kjenstad, Carlo Mannino, Tomas Eric Nordlander, Patrick Schittekat, and Morten Smedsrud. Optimizing AMAN-SMAN-DMAN at Hamburg and Arlanda airport. *Proceedings of the SID, Stockholm*, 2013.

[108] Hunter D Kopald, Ari Chanen, Shuo Chen, Elida C Smith, and Robert M Tarakan. Applying automatic speech recognition technology to air traffic management. In *2013 IEEE/AIAA 32nd Digital Avionics Systems Conference (DASC)*, pages 6C3–1. IEEE, 2013.

[109] Hanbong Lee. *Airport surface traffic optimization and simulation in the presence of uncertainties*. PhD thesis, Massachusetts Institute of Technology, 2014.

[110] Hanbong Lee, Waqar Malik, Bo Zhang, Balaji Nagarajan, and Yoon C Jung. Taxi time prediction at charlotte airport using fast-time simulation and machine learning techniques. In *15th AIAA Aviation Technology, Integration, and Operation (ATIO) Conference, Dallas, TX*, 2015.

[111] Li Li, Victor Okoth, and Saif Eddin Jabari. Backpressure control with estimated queue lengths for urban network traffic. *IET Intelligent Transport Systems*, 2020.

[112] Max Z Li and Megan S Ryerson. Reviewing the datas of aviation research data: Diversity, availability, tractability, applicability, and sources. *Journal of Air Transport Management*, 75:111–130, 2019.

[113] Yi Lin, Linjie Deng, Zhengmao Chen, Xiping Wu, Jianwei Zhang, and Bo Yang. A real-time atc safety monitoring framework using a deep learning approach. *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[114] Robin Lineberger, Aijaz Hussain, and David Silver. Advanced air mobility: Can the united states afford to lose the race?

[115] Yi Liu, Mark Hansen, Gautam Gupta, Waqar Malik, and Yoon Jung. Predictability impacts of airport surface automation. *Transportation Research Part C: Emerging Technologies*, 44:128–145, 2014.

[116] Remy Lopez and Patrick Danes. Low-complexity IMM smoothing for jump markov nonlinear systems. *IEEE Transactions on Aerospace and Electronic Systems*, 2017.

[117] David J Lovell, Kleoniki Vlachou, Tarek Rabbani, and Alexandre Bayen. A diffusion approximation to a single airport queue. *Transportation Research Part C: Emerging Technologies*, 33:227–237, 2013.

[118] Steven H Low, Fernando Paganini, and John C Doyle. Internet congestion control. *IEEE control systems magazine*, 22(1):28–43, 2002.

[119] Michael Lowry, Thomas Pressburger, Deborah A Dahl, and Michael Dalal. Towards autonomous piloting: Communicating with air traffic control. In *AIAA Scitech 2019 Forum*, page 2207, 2019.

[120] Hui-Ling Lu, Victor H Cheng, Deborah Ballinger, Andrew Fong, Jimmy Nguyen, Stephen Jones, and Steven E Cowart. A speech-enabled simulation interface agent for airspace system assessments. In *AIAA Modeling and Simulation Technologies Conference*, page 0148, 2015.

[121] Guglielmo Lulli, Amedeo Odoni, and Bruno F Santos. Introduction to the special section: Air transportation systems planning and operations under uncertainty. *Transportation Science*, 54(4):855–857, 2020.

[122] Ji Ma, Daniel Delahaye, Mohammed Sbihi, and Paolo Scala. Integrated Optimization of Arrival, Departure, and Surface Operations. In *ICRAT 2018, 8th International Conference for Research in Air Transportation*, 2018.

[123] Moussa Maiga, Nacim Ramdani, Louise Travé-Massuyès, and Christophe Combastel. A comprehensive method for reachability analysis of uncertain nonlinear hybrid systems. *IEEE Transactions on Automatic Control*, 61(9):2341–2356, 2015.

[124] Waqar Malik, Gautam Gupta, and Yoon Jung. Managing departure aircraft release for efficient airport surface operations. In *AIAA Guidance, Navigation, and Control Conference*, page 7696, 2010.

[125] Mauro Masiol and Roy M Harrison. Aircraft engine exhaust emissions and other airport-related contributions to ambient air pollution: A review. *Atmospheric Environment*, 95:409–455, 2014.

[126] Patrick McFarlane and Hamsa Balakrishnan. Optimal control of airport pushbacks in the presence of uncertainties. In *American Control Conference (ACC), 2016*, pages 233–239. IEEE, 2016.

[127] Vineet Mehta, Mary Ellen Miller, Tom Reynolds, Mariya Ishutkina, Richard Jordan, Robert Seater, and William Moser. Decision support tools for the tower flight data manager system. In *2011 Integrated Communications, Navigation, and Surveillance Conference Proceedings*, pages I4–1. IEEE, 2011.

[128] Pierre-Jean Meyer, Alex Devonport, and Murat Arcak. TIRA: Toolbox for Interval Reachability Analysis. *arXiv preprint arXiv:1902.05204*, 2019.

[129] Pierre-Jean Meyer and Dimos V Dimarogonas. Hierarchical decomposition of LTL synthesis problem for nonlinear control systems. *IEEE Transactions on Automatic Control*, 2019.

[130] Nicolò Michelusi, James Boedicker, Mohamed Y El-Naggar, and Urbashi Mitra. Queuing models for abstracting interactions in bacterial communities. *IEEE Journal on Selected Areas in Communications*, 34(3):584–599, 2016.

[131] Vishal Misra, Wei-Bo Gong, and Don Towsley. Fluid-based analysis of a network of aqm routers supporting tcp flows with an application to red. In *ACM SIGCOMM Computer Communication Review*, volume 30, pages 151–160. ACM, 2000.

[132] Mozilla Corporation. DeepSpeech Model. `https://deepspeech.readthedocs.io/en/v0.7.3/DeepSpeech.html`, 2020. Retrieved on June 4th, 2020.

[133] Mayara Condé Rocha Murça. A robust optimization approach for airport departure metering under uncertain taxi-out time predictions. *Aerospace Science and Technology*, 68:269–277, 2017.

[134] A Nakahara and Tom G Reynolds. Estimating current & future system-wide benefits of airport surface congestion management. In *10th USA/Europe Air Traffic Management Research and Development Seminar, Chicago*, 2013.

[135] Kai Ling Neo. Analysis of airside operations at singapore changi airport. Master's thesis, Massachusetts Institute of Technology, 2019.

[136] Van Nhan Nguyen. Using linguistic knowledge for improving automatic speech recognition accuracy in air traffic control. Master's thesis, 2016.

[137] Van Nhan Nguyen and Harald Holone. Possibilities, challenges and the state of the art of automatic speech recognition in air traffic control. *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 9(8):1940–1949, 2015.

[138] OAG. Airport operations data. `https://www.oag.com`, 2016. Retrieved Dec 14, 2018.

[139] Nikolai Okuniek and Lukas Sparenberg. Opportunities and challenges when implementing trajectory-based taxi operations at European and US CDM airports. In *Digital Avionics Systems Conference (DASC), 2017 IEEE/AIAA 36th*, pages 1–10. IEEE, 2017.

[140] Carolina Osorio and Michel Bierlaire. An analytic finite capacity queueing network model capturing the propagation of congestion and blocking. *European Journal of Operational Research*, 196(3):996–1007, 2009.

[141] Limeng Pan, Mashor Housh, Pan Liu, Ximing Cai, and Xin Chen. Robust stochastic optimization for reservoir operation. *Water Resources Research*, 51(1):409–429, 2015.

[142] Markos Papageorgiou, Christina Diakaki, Vaya Dinopoulou, Apostolos Kotsialos, and Yibing Wang. Review of road traffic control strategies. *Proceedings of the IEEE*, 91(12):2043–2067, 2003.

[143] Thomas Pellegrini, Jérôme Farinas, Estelle Delpech, and François Lancelot. The airbus air traffic control speech recognition 2018 challenge: Towards atc automatic transcription and call sign detection. *arXiv preprint arXiv:1810.12614*, 2018.

[144] Andreas Pitsillides, Petros Ioannou, Marios Lestas, and Loukas Rossides. Adaptive nonlinear congestion controller for a differentiated-services framework. *IEEE/ACM Transactions on Networking (TON)*, 13(1):94–107, 2005.

[145] Michele Polese, Federico Chiariotti, Elia Bonetto, Filippo Rigotto, Andrea Zanella, and Michele Zorzi. A survey on recent advances in transport layer protocols. *arXiv preprint arXiv:1810.03884*, 2018.

[146] Nicolas Pujet, Bertrand Delcaire, and Eric Feron. Input-output modeling and control of the departure process of busy airports. *Air Traffic Control Quarterly*, 8(1):1–32, 2000.

[147] Nikolas Pyrgiotis, Kerry M Malone, and Amedeo Odoni. Modelling delay propagation within an airport network. *Transportation Research Part C: Emerging Technologies*, 27:60–75, 2013.

[148] David Rappaport, Peter Yu, Katy Griffin, and Chris Daviau. Quantitative analysis of uncertainty in airport surface operations. In *9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO) and Aircraft Noise and Emissions Reduction Symposium (ANERS)*, page 6987, 2009.

[149] Isaac Robeson, William J Coupe, Hanbong Lee, Yoon Jung, Liang Chen, Leonard Bagasol, Bob Staudenmeier, and Pete Slattery. Strategic surface metering at charlotte douglas international airport. In *2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC)*, pages 1–10. IEEE, 2020.

[150] Álvaro Rodríguez-Sanz, Alberto Fernández de Marcos, Javier A Pérez-Castán, Fernando Gómez Comendador, Rosa Arnaldo Valdés, and Ángel París Loreiro. Queue behavioural patterns for passengers at airport terminals: A machine learning approach. *Journal of Air Transport Management*, 90:101940, 2021.

[151] Ramon Sanchez-Iborra and Maria-Dolores Cano. On the similarities between urban traffic management and communication networks: Application of the random early detection algorithm for self-regulating intersections. *IEEE Intelligent Transportation Systems Magazine*, 9(4):48–61, 2017.

[152] Shankar Sastry. *Nonlinear systems: analysis, stability, and control*, volume 10. Springer Science & Business Media, 2013.

[153] David A Senzig, Gregg G Fleming, and Ralph J Iovinelli. Modeling of terminal-area airplane fuel consumption. *Journal of Aircraft*, 46(4):1089–1093, 2009.

[154] Shivanjli Sharma, Al Capps, Shawn Engelland, Yoon Jung, and Richard A Coppenbarger. Operational Impact of the Baseline Integrated Arrival, Departure, and Surface System Field Demonstration. In *Digital Avionics Systems Conference*, 2018.

[155] Ni Shen, Husni Idris, and Vincent Orlando. Estimation of departure metering benefits at major airports using queuing analysis. In *2012 IEEE/AIAA 31st Digital Avionics Systems Conference (DASC)*, pages 4E5–1. IEEE, 2012.

[156] Ioannis Simaiakis and Hamsa Balakrishnan. Probabilistic modeling of runway inter-departure times. *Journal of Guidance, Control, and Dynamics*, 37(6):2044–2048, November 2014.

[157] Ioannis Simaiakis and Hamsa Balakrishnan. A queuing model of the airport departure process. *Transportation Science*, 50(1):94–109, 2015.

[158] Ioannis Simaiakis, Harshad Khadilkar, Hamsa Balakrishnan, Tom G Reynolds, and R John Hansman. Demonstration of reduced airport congestion through pushback rate control. *Transportation Research Part A: Policy and Practice*, 66:251–267, 2014.

[159] Ioannis Simaiakis and Nikolas Pyrgiotis. An analytical queuing model of airport departure processes for taxi out time prediction. In *AIAA Aviation Technology, Integration and Operations (ATIO) Conference*, 2010.

[160] Ioannis Simaiakis, Melanie Sandberg, and Hamsa Balakrishnan. Dynamic control of airport departures: Algorithm development and field evaluation. *IEEE Transactions on Intelligent Transportation Systems*, 15(1):285–295, 2013.

[161] Jacek Skorupski and Hubert Wierzbicki. Airport capacity increase via the use of braking profiles. *Transportation Research Part C: Emerging Technologies*, 80:467–484, 2017.

[162] Jean-Jacques E Slotine, Weiping Li, et al. *Applied nonlinear control*, volume 199. Prentice hall Englewood Cliffs, NJ, 1991.

[163] Luboš Šmídl, Jan Švec, Aleš Pražák, and Jan Trmal. Semi-supervised training of dnn-based acoustic model for atc speech recognition. In *International Conference on Speech and Computer*, pages 646–655. Springer, 2018.

[164] Spacy. `https://spacy.io`, 2020. Retrieved Jun 19, 2020.

[165] Ajay Srinivasamurthy, Petr Motlicek, Ivan Himawan, Gyorgy Szaszak, Youssef Oualil, and Hartmut Helmke. Semi-supervised learning with semantic knowledge extraction for improved speech recognition in air traffic control. Technical report, 2017.

[166] Shaler Stidham and Richard Weber. A survey of markov decision models for control of networks of queues. *Queueing systems*, 13(1-3):291–314, 1993.

[167] Shaler Stidham Jr. Analysis, design, and control of queueing systems. *Operations Research*, 50(1):197–216, 2002.

[168] Steven Stroiney and Benjamin Levy. Departure queue management benefits across many airports. In *2011 Integrated Communications, Navigation, and Surveillance Conference Proceedings*, pages M5–1. IEEE, 2011.

[169] Steven Stroiney, Benjamin Levy, Harshad Khadilkar, and Hamsa Balakrishnan. Assessing the impacts of the jfk ground management program. In *2013 IEEE/AIAA 32nd Digital Avionics Systems Conference (DASC)*, pages 1B2–1. IEEE, 2013.

[170] Shreyas Vathul Subramanian, Peter F Kostiuk, and Graham Katz. Custom ibm watson speech-to-text model for anomaly detection using atc-pilot voice communication. In *2018 Aviation Technology, Integration, and Operations Conference*, page 3979, 2018.

[171] Robert Tarakan, Keith Baldwin, and Nicholas Rozen. An automated simulation pilot capability to support advanced air traffic controller training. In *The 26th Congress of ICAS and 8th AIAA ATIO*, page 8897, 2008.

[172] Leandros Tassiulas. Adaptive back-pressure congestion control based on local information. *IEEE Transactions on Automatic Control*, 40(2):236–250, 1995.

[173] The Times of India. `https://timesofindia.indiatimes.com/india/over-160-dead-in-flood-rain-related-incidents-in-south-and-west-india/articleshow/70631801.cms`. Retrieved Sept 8, 2019.

[174] ARILD THOWSEN. Uniform ultimate boundedness of the solutions of uncertain dynamic delay systems with state-dependent and memoryless feedback control. *International Journal of control*, 37(5):1135–1143, 1983.

[175] David Tipper and Malur K. Sundareshan. Numerical methods for modeling computer networks under nonstationary conditions. *IEEE Journal on Selected Areas in Communications*, 8(9):1682–1695, 1990.

[176] Transportation Research Board and National Academies of Sciences, Engineering, and Medicine. *Enhanced Modeling of Aircraft Taxiway Noise, Volume 1: Scoping*. The National Academies Press, Washington, DC, 2009.

[177] Parker D Vascik. *Systems analysis of urban air mobility operational scaling, Chapter 8*. PhD thesis, Massachusetts Institute of Technology, 2020.

[178] Michael H Veatch and Lawrence M Wein. Optimal control of a two-station tandem production/inventory system. *Operations Research*, 42(2):337–350, 1994.

[179] Savita Verma, William J Coupe, Hanbong Lee, Isaac Robeson, Yoon Jung, Shivanjli Sharma, Victoria L Dulchinos, and Lindsay Stevens. Tactical surface metering procedures and information needs for charlotte douglas international airport. In *International Conference on Applied Human Factors and Ergonomics*, pages 157–169. Springer, 2018.

[180] Eric A Wan and Rudolph Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, pages 153–158. IEEE, 2000.

[181] Carter Wang. *Urban transportation networks: analytical modeling of spatial dependencies and calibration techniques for stochastic traffic simulators*. PhD thesis, Massachusetts Institute of Technology, 2013.

[182] Jinwen Wang. A new stochastic control approach to multireservoir operation problems with uncertain forecasts. *Water resources research*, 46(2), 2010.

[183] Wei-Ping Wang, David Tipper, and Sujata Banerjee. A simple approximation for modeling nonstationary queues. In *INFOCOM'96. Fifteenth Annual Joint Conference of the IEEE Computer Societies*, volume 1, pages 255–262. IEEE, 1996.

[184] Y. Wang, T. Liu, M. Hu, S. Alam, and V. N. Duong. Temporal patterns underlying domestic departure passengers behavior in the airport. *IEEE Access*, 8:127969–127980, 2020.

[185] Robert D Windhorst, Justin V Montoya, Zhifan Zhu, Sergei Gridnev, Katy Griffin, Aditya Saraf, and Steven Stroiney. Validation of simulations of airport surface traffic with the surface operations simulator and scheduler. In *Los Angeles, CA, 13th AIAA Aviation Technology, Integration, and Operations Conference*, 2013.

[186] Qinggang Wu, Minghua Hu, Xiaozhen Ma, Yanjun Wang, Wei Cong, and Daniel Delahaye. Modeling flight delay propagation in airport and airspace network. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3556–3561. IEEE, 2018.

[187] Lei Yang, Suwan Yin, Ke Han, Jack Haddad, and Minghua Hu. Fundamental diagrams of airport surface traffic: Models and applications. *Transportation research part B: Methodological*, 106:29–51, 2017.

[188] Liren Yang, Oscar Mickelin, and Necmiye Ozay. On sufficient conditions for mixed monotonicity. *IEEE Transactions on Automatic Control*, 2019.

[189] Chuhang Yu, Dong Zhang, and HYK Henry Lau. A heuristic approach for solving an integrated gate reassignment and taxi scheduling problem. *Journal of Air Transport Management*, 62:189–196, 2017.