CONTROL DESIGN TECHNIQUES FOR
CONSTRAINED POSITIVE COMPARTMENTAL SYSTEMS WITH
APPLICATIONS TO AIR TRAFFIC FLOW MANAGEMENT

BY

HEATHER M. ARNESON

DISSEYATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Aerospace Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2012

Urbana, Illinois

Doctoral Committee:

Assistant Professor Cédric Langbort, Chair and Director of Research
Professor Petros Voulgaris
Professor Naira Hovakimyan
Assistant Professor Hamsa Balakrishnan
Abstract

The current air traffic management system has been pushed to its limit and will not be able to keep up with the predicted increase in air traffic over the coming years. Recent research in air traffic management is concerned with increasing the capacity and throughput of the National Airspace System in order to accommodate this growing demand. Many approaches presented in the literature are systematic and performance based. In the work presented here, two particular problems are addressed: delay scheduling in the presence of uncertain flow rate constraints, and traffic flow routing under time varying airspace capacity constraints. Aggregate models are used to describe the flow of traffic in the region of airspace of interest. The solution methods presented are based on sliding mode control theory and linear programming theory. Contributions of this work are methods which: (a) react in real time to changing flow rate constraints, and (b) use routing parameters to satisfy time varying capacity constraints for linear, uncertain linear, and nonlinear models describing the flow of traffic through the region of interest. Unlike most methods described in the literature, solution method (a) does not make use of a constraint forecast. Simulation results show a reduction of flow rate constraint violation over the baseline schedule. Routing control is rarely used in the literature. The proposed approach (b) makes use of routing parameters as the primary control input. Linear constraints are derived to find time varying routing parameters which satisfy time varying capacity constraints.
Acknowledgments

I would like to thank my advisor, Prof. Cédric Langbort, for his guidance throughout my graduate studies. Meetings with him left me excited about our discoveries and reinvigorated to tackle the problems at hand. I learned a great deal from him, not only about the technical work, but also about the process of conducting research. Periodic reflection on our research progress and process would help me see the seemingly meandering path we took on the road to each discovery as a necessary part of the process, increasing our understanding of the problem and leading us towards an elegant solution.

I would also like to thank my doctoral committee members, Prof. Petros Voulgaris, Prof. Naira Hovakimyan, and Prof. Hamsa Balakrishnan, for their insightful questions and comments about my research.

I was fortunate to spend several summers conducting research at NASA Ames Research Center. I would like to thank my colleagues in the Aviation Systems Division at NASA Ames, particularly Michael Bloem, who served as my mentor during these summer internships, and Avijit Mukherjee. My experiences at NASA Ames greatly increased my knowledge and understanding of the current problems and constraints in air traffic flow management. This knowledge has been invaluable to me in directing my research and understanding and communicating the implications of my work.
# Table of Contents

List of Tables ................................................................. viii  
List of Figures ............................................................... ix  
List of Abbreviations ......................................................... xii  
Summary ................................................................. xiii  

Chapter 1 Introduction ......................................................... 1  
  1.1 Air Traffic Management Overview ............................................ 1  
    1.1.1 Current Operations ................................................ 1  
    1.1.2 Airspace Flow Program: Flight Scheduling with Capacity Constraints .......... 2  
    1.1.3 Weather and Capacity Predictions ........................................ 4  
  1.2 Literature Review ..................................................... 5  
    1.2.1 Research in Air Traffic Management Problems .............................. 5  
    1.2.2 Aggregate Models for Air Traffic Flow Management ......................... 7  
  1.3 Contributions of this Work ................................................ 9  
    1.3.1 Problems Addressed ................................................ 10  
    1.3.2 Control Techniques ................................................ 11  
  1.4 Organization of this Work ............................................... 13  
  1.5 Notation ............................................................ 14  

Chapter 2 Aggregate Model Details ......................................... 15  
  2.1 Basic Network Properties ............................................... 15  
  2.2 Continuous Time Aggregate Model ........................................ 16  
    2.2.1 Control Parameters ................................................ 19  
    2.2.2 Basic Control Design Objectives ...................................... 20
# Chapter 3  LP Based Routing Design for Systems with Integral Objective and Constraints

3.1 Motivation and Problem Description .................................................. 22
3.2 Continuous Time Aggregate Model ..................................................... 22
3.3 Control Objectives ............................................................................. 23
3.4 Control Design .................................................................................. 25
3.5 Application Example ......................................................................... 29

# Chapter 4  LP Based Routing Design for Systems with Time Varying Capacity Constraints

4.1 Motivation and Problem Description .................................................. 32
4.2 Chapter Overview .............................................................................. 32
4.3 Systems with a Fixed Linear Outflow Model ...................................... 34
    4.3.1 Motivation ................................................................................. 34
    4.3.2 Continuous Time Aggregate Model .......................................... 35
    4.3.3 Control Objectives ................................................................. 35
    4.3.4 Control Design ........................................................................ 36
        4.3.4.1 Linear Capacity Bound .................................................... 37
        4.3.4.2 Piecewise Constant Capacity Constraints ...................... 38
    4.3.5 Application Example ............................................................... 41
4.4 Systems with an Uncertain Linear Time Varying Outflow Model ............ 44
    4.4.1 Motivation ................................................................................. 44
    4.4.2 Continuous Time Aggregate Model .......................................... 44
    4.4.3 Control Objectives ................................................................. 45
    4.4.4 Control Design ...................................................................... 46
    4.4.5 Application Example ............................................................... 52
4.5 Systems with a Nonlinear Outflow Model .......................................... 57
    4.5.1 Motivation ................................................................................. 57
    4.5.2 Continuous Time Aggregate Model .......................................... 58
    4.5.3 Control Objectives ................................................................. 59
    4.5.4 Control Design ...................................................................... 59
List of Tables

1.1 Table summarizing some of the key properties of each of the three control techniques presented here (first three lines of the table) and the control techniques presented in Section 1.2.1. Note that Sun et al. [43] are able to incorporate a variety of control design objectives in the model framework they present. These objectives are not captured adequately by the list of objectives selected for this table. See Section 1.2.1 for the specific control design objectives used in this work. ................................................................. 12

3.1 Routing and simulation results. ......................................................... 31

6.1 Performance and delay results for the application of various delay scheduling algorithms and procedures. ................................................................. 123
List of Figures

2.1 Simple network example consisting of a single source and four interconnected sections. In this example, \( O_1 = \{1, 3\} \), \( O_2 = \{2, 3, 4\} \), \( O_3 = \emptyset \), \( O_4 = \emptyset \), and \( S_F = \{3, 4\} \). ........................................ 19

3.1 Network of airspace used in application example. .................................................. 29

4.1 Network of interconnected sections. Aircraft enter the network at sections 1, 2 and 3, aircraft exit the network from sections 19, 20, 21. .................................................. 41

4.2 Capacity constraint \( \hat{c}_{14} \), capacity bound \( c_{14} \) and state \( x_{14} \) of section 14. ......................... 42

4.3 Routing parameters associated with results plotted in Figure 4.2. Note that, due to symmetry of the problem, several of the routing parameters have identical profiles. Also recall that \( \beta_{1j} (t) \) must sum to 1 for each section, and thus recirculation accounts for the remainder of the flow routing (i.e. \( \beta_{11} (t) = 1 - \beta_{1,4} (t) - \beta_{1,5} (t) \), etc.). .................................................. 43

4.4 Capacity constraint \( \hat{c}_{14} \), adjusted capacity constraint \( \hat{c}_{14} \), capacity bound \( c_{14} \) and state \( x_{14} \) of section 14. Note that \( \hat{c}_{14} \) is only plotted over intervals in which it differs from \( \hat{c}_{14} \). ......................... 43

4.5 Network of airspace used in application example. .................................................. 51

4.6 Capacity constraint for sections 2 and 3 of the network depicted in Figure 4.5. ......................... 53

4.7 Simulation results for the routing solution obtained from LP problem (4.18) with \( D_1 = \emptyset \), and realized traversal times of \( \tau = [0.5, 0.4, 0.5] \). Recall that \( \beta_{1j} (t) \) must sum to 1 for section 1, and thus recirculation accounts for the remainder of the flow routing (i.e. \( \beta_{11} (t) = 1 - \beta_{1,2} (t) - \beta_{1,3} (t) \)). .................................................. 54

4.8 Simulation results for the routing solution obtained from LP problem (4.18) with \( D_1 = \{1\} \), and realized traversal times of \( \tau = [0.5, 0.4, 0.5] \). Recall that \( \beta_{1j} (t) \) must sum to 1 for section 1, and thus recirculation accounts for the remainder of the flow routing (i.e. \( \beta_{11} (t) = 1 - \beta_{1,2} (t) - \beta_{1,3} (t) \)). .................................................. 55

4.9 Simulation results for the routing solution obtained from LP problem (4.18) with \( D_1 = \{2\} \), and realized traversal times of \( \tau = [0.5, 0.4, 0.5] \). Recall that \( \beta_{1j} (t) \) must sum to 1 for section 1, and thus recirculation accounts for the remainder of the flow routing (i.e. \( \beta_{11} (t) = 1 - \beta_{1,2} (t) - \beta_{1,3} (t) \)). .................................................. 56

4.10 Simulation results for the routing solution obtained from LP problem (4.18) with \( D_1 = \{1, 2\} \), and realized traversal times of \( \tau = [0.5, 0.4, 0.5] \). Recall that \( \beta_{1j} (t) \) must sum to 1 for section 1, and thus recirculation accounts for the remainder of the flow routing (i.e. \( \beta_{11} (t) = 1 - \beta_{1,2} (t) - \beta_{1,3} (t) \)). .................................................. 57
4.11 Network of interconnected sections. Material flows into the network at sections 1, 2 and 3, material exits the network from sections 19, 20, 21. 65

4.12 Nonlinear outflow rate curve used for each section in the network. 66

4.13 Capacity constraint profile for section 14. 67

4.14 Nonlinear section outflow rate $\mu$ and the two piecewise linear under approximations $\tilde{\mu}$ of $\mu$ used in this example. 68

4.15 Capacity constraint $\bar{c}_{14}$, capacity bound $c_{14}$ and state $\tilde{x}_{14}$ of section 14. 69

4.16 Routing parameters associated with results plotted in Figure 4.2. Note that, due to symmetry of the problem, several of the routing parameters have identical profiles. Also recall that $\tilde{\beta}_{ij}(t)$ must sum to 1 for each section, and thus recirculation accounts for the remainder of the flow routing (i.e. $\tilde{\beta}_{1,1}(t) = 1 - \tilde{\beta}_{1,4}(t) - \tilde{\beta}_{1,5}(t)$, etc.). 69

4.17 Figures 4.17(a), 4.17(b), 4.17(c), and 4.17(d) depict the connectivity of the full network and sub-networks associated with destination 1, 2 and 3, respectively. The number in each box indicates the index of that section of airspace. Arrows indicate allowable flow between sections of the network or sub-network. Recirculation is allowed for all sections that are not final sections. 74

4.18 Figures 4.18(a), 4.18(b), 4.18(c), and 4.18(d) indicate inflow rates and section initial conditions for all flights in the network and flights with destination 1, 2 and 3, respectively. The inflow at sources is indicated in the ovals on the left of each diagram. The value in each box represents the initial number of aircraft in that section. 75

4.19 Capacity constraint, resulting capacity bounds and full network state for section 14 in Figure 4.19(a). Resulting capacity bounds and sub-network states for section 14 and destinations 1, 2, and 3, respectively, in Figures 4.19(b), 4.19(c), and 4.19(d). 76

5.1 Example of a network of flow between source $A$ and sink $B$. Sections 1, 2 and 3 are in level 1, sections 4, 5, 6 are in level 2 and sections 7, 8, 9 are in level 3. 78

5.2 Graphical representation of the autonomous hybrid automaton $H$. 84

5.3 Airspace between airports $A$ and $B$ divided into sections of 1-D flow, arrows indicate direction of flow. 105

5.4 Maximum allowable landing rate capacity, uncontrolled landing rate, and controlled landing rate (left) and control parameter $\alpha$ as a function of time (right). 107

5.5 Maximum allowable landing rate capacity, uncontrolled landing rate, and controlled landing rate (left) and control parameter $\alpha$ as a function of time (right). 108

6.1 FACET screen capture of the region of airspace of interest for this problem. Sectors included in the aggregate model are outlined in white, the grey lines indicate flows of air traffic between these sectors leading to the arrival fix, indicated by the white circle. 114

6.2 Actual and estimated number of selected aircraft in sector ZAU34 (a) for each time step (b) averaged over 15 minute intervals. 115
6.3 Actual and estimated number of arrivals at the arrival fix summed over 15 minute time intervals with (a) no state reset (b) the state of the model system reset to the state of the actual system every 30 minutes.

6.4 All arrivals at OXI fix on August 24, 2005 between 14:00 CDT and 17:00 CDT. Arrivals are summed over 15 minute time periods. A total of 80 flights arrive during this time window.

6.5 OXI arrival constraint profiles used in FACET simulations. The number of allowable arrivals in each 15 minute time period are plotted for the 3 hour planning horizon.

6.6 Constraint Profile 1 and actual arrivals when no control action is applied. These results correspond to the Constraint Profile 1 row under section “No Control” of Table 6.7.

6.7 Constraint Profile 1 and actual arrivals when departure delays were set using the IP Delay Scheduling method with Profile 1 as the predicted constraint profile. These results correspond to the Constraint Profile 1 row under sections “IP Departure Delay – Perfect Prediction” and “IP Departure Delay – Profile 1 Predicted” of Table 6.7.

6.8 Constraint Profile 1 and actual arrivals when departure delays were set using the IP Delay Scheduling method with Profile 1 as the predicted constraint profile and the SMC method was used to add airborne delay in order to satisfy arrival rate constraints as they were revealed in real time. These results correspond to the Constraint Profile 1 row under section “Sliding Mode Control with IP Departure Delay for Profile 1” of Table 6.7.

6.9 Control parameter generated by the SMC method for the FACET simulation results shown in Figure 6.8. The dashed curve is the continuous value of $\alpha$ generated by the SMC method using the aggregate model estimate of the state of the system. The solid curve is the piecewise constant average $\hat{\alpha}$ of the continuous value of $\alpha$. The control parameter input used for the FACET simulation is the averaged $\hat{\alpha}$. 
List of Abbreviations

AFP Airspace Flow Program
ARTCC or Center Air Route Traffic Control Center
ATCSCC Air Traffic Control System Command Center
ATFM Air Traffic Flow Management
ATM Air Traffic Management
CCFP Collaborative Convective Forecast Product
CIWS Corridor Integrated Weather System
CWAM Convective Weather Avoidance Model
FAA The Federal Aviation Administration
FACET Future ATM Concepts Evaluation Tool
FCA Flow Constrained Area
FEA Flow Evaluation Area
GDP Ground Delay Program
IP Integer Programming or Integer Program
JPDO Joint Planning and Development Office
LP Linear Programming or Linear Program
NAS National Airspace System
NCWF National Convective Weather Forecast
ORD Chicago O’Hare International airport
OXI Knox arrival fix of Chicago O’Hare International airport
RBS Ration by Schedule
SMC Sliding Mode Control
SWAP Severe Weather Avoidance Procedures
TFM Traffic Flow Management
Summary

Currently, the U.S. air transportation system manages approximately 50,000 flights each day. The current air traffic management (ATM) system relies on human air traffic controllers both to manage flow and to resolve conflicts among aircraft. Control actions determined by human air traffic controllers are based on experience dealing with the traffic and weather patterns typical to the specific region of airspace covered by the controller. Since air traffic is predicted to soon increase beyond the capabilities of the current ATM system, it has become necessary to devise a new ATM system to accommodate this increased demand [3]. The main issues involved in the design of such a system are satisfying capacity constraints (a constraint on the number of aircraft allowed to be in a certain region of airspace) or arrival rate constraints at an airport or the boundary of a region of airspace. These constraints are inherently uncertain, since they depend on uncertain weather predictions. Dealing with this uncertainty is an important aspect of any control technique designed to satisfy these types of constraints. Pushing against these constraints is the desire to minimize delay. Delays, both ground and airborne, are costly for airlines. After safety considerations, the minimization of delay is often the primary objective in ATM problems. A more systematic, performance based approach which addresses these problems must be developed in order to keep up with the predicted increase in air traffic.

Specifically, the work presented here addresses the problems of (a) routing control for integral objective and constraints, (b) routing control for time varying capacity constraints, and (c) real time reaction to changing flow rate constraints. Three control techniques, which will be referred to as LP Routing (Integral), LP Routing (Capacity) and SMC are presented which address each of these fundamental research problems. The first method, LP Routing (Integral), minimizes a measure of delay, formulated as an integral, subject to integral constraints using routing parameters as the control input. In the second method, LP Routing (Capacity), routing parameters are used to satisfy capacity constraints. Finally, the SMC method is a sliding mode control based algorithm which calculates airborne delay in order to satisfy flow rate constraints. Each of these control design techniques makes use of an aggregate model of air traffic.
flows. Aggregate models, first introduced in [35], are a popular method of describing air traffic networks. Such models describe the aggregate dynamics of groups of aircraft rather than focusing on individual flights. The use of aggregate models in air traffic flow control problems has become popular in part because these models lend themselves to traditional linear system control design. A discussion of the types of aggregate models which have been proposed for ATM applications is given in Section 1.2.2. The LP Routing (Integral) and SMC methods both make use of a fixed linear model describing the flow of traffic through the region of airspace of interest. In addition to the fixed linear model, the LP Routing (Capacity) method provides control design procedures for an uncertain linear model and a nonlinear model.
Chapter 1

Introduction

1.1 Air Traffic Management Overview

1.1.1 Current Operations

Before suggesting changes in an attempt to increase the capacity and throughput of the NAS to keep up with growing air traffic demand, we must first understand the current state of ATM. Below is a summary of the current ATM as presented in [38] and [47].

The current ATM system relies on humans to monitor and direct the movements of individual aircraft in the NAS both to manage flow and to resolve conflicts among aircraft. The control of en route aircraft (aircraft that are traveling between airports and are in regions outside the immediate vicinity of an airport) is divided into two levels. Once aircraft have left the immediate vicinity of their departure airports, they are managed by controllers at air route traffic control centers (ARTCCs). At a higher level, flow of aircraft through the nation’s airspace is managed by the Air Traffic Control System Command Center (ATCSCC) located in Herndon, Virginia.

The Federal Aviation Administration (FAA) has divided the nation’s airspace into 24 areas, each under the control of a separate ARTCC. Controllers at an ARTCC are responsible for directing aircraft in their area so that collisions are avoided and a desired flow rate through the region is maintained. Since the region of airspace under the jurisdiction of each ARTCC is too large for an individual person to be able to direct all aircraft within the area, each ARTCC is further divided into smaller regions, called sectors, for which an individual controller is responsible.
A higher level of flow management is performed at the ATCSCC, where each morning a plan for air traffic flow across the nation is developed. Variables that may restrict flow through certain regions of airspace and the capacity at certain airports, such as weather, are taken into consideration at this level of flow management. Although the ATCSCC will monitor the flow throughout the nation as it evolves, this flow plan is generally formulated only once per day, even though unexpected changes occur causing bottlenecks and backups. Such real time traffic buildups are mitigated by the ARTCCs involved. If these traffic buildups get too large, e.g. involving three or more ARTCCs, the ATCSCC will get involved in solving the problem and develop a new traffic flow plan.

This current system has been pushed to its capacity, with the limiting factor being the number of aircraft that each sector controller can safely manage. With the current technology, each controller can manage a finite number of aircraft. To increase capacity, airspace within each ARTCC could be broken into smaller sectors requiring more air traffic controllers to manage these new subdivided sectors. But this too would have its limits. As stated in [38], with more sectors and controllers within each ARTCC, the communication between controllers required to mitigate congestion and hand off aircraft between sectors would become prohibitively complex.

In 2003, the U.S. Government created the Joint Planning and Development Office (JPDO) to oversee the development of a new system designed to accommodate the predicted increase in air traffic [3]. In the JPDO’s “Next Generation Air Transportation System Integrated Plan” [4], a plan is laid out for the changes needed to be made in the current ATM system in order for the volume of aircraft that can be safely routed in the NAS to match the growing demand. Specifically, this plan calls for a shift from the control of individual flights to the control of traffic flows, with an emphasis on end-to-end strategic flow management. Additionally, distributed decision making will decrease the communication complexity and burden on controllers. Further reducing communication complexity, automatic data transmissions will take the place of voice communication, allowing for faster and more accurate exchange of information.

1.1.2 Airspace Flow Program: Flight Scheduling with Capacity Constraints

When the capacity of en route airspace is limited due to convective weather or other factors, the traffic through this area must often be mitigated so as not to exceed the capacity. With weather conditions causing approximately 65% of delays in the NAS in recent years [1], minimizing delay of flights traveling through a Flow Constrained Area (FCA) can significantly reduce overall delay in the NAS. The emerging method of metering flights travelling through an FCA is the Airspace Flow Program (AFP).
Prior to 2006, the primary method of metering flights traveling through an FCA was the use of Ground Delay Programs (GDP) in support of Severe Weather Avoidance Procedures (SWAP). Typically, a GDP is used to assign ground delays to flights whose destination airport has landing rate constraints. Departures of flights heading to airports in the GDP are metered in order to satisfy the airport capacity constraints. Using GDP in support of SWAP, several destination airports are selected based on their contribution to flights traveling through the FCA. All flights traveling to these airports are given ground delays through the same procedure used to deal with airport capacity constraints.

Some of the drawbacks of using GDP in support of SWAP is that the GDP does not take into account the route that aircraft plan to follow. For this reason, the GDP captures some flights that are not scheduled to fly through the FCA and does not capture other flights that are scheduled to fly through the FCA. This leads to unnecessary delays for some flights while other flights that should be delayed are not [34]. Additionally, since a limited number of airports can be selected, larger airports that contribute more flights to the predicted traffic in the FCA are chosen over smaller airports [34]. Thus, the distribution of ground delay among airports contributing to the predicted traffic through the FCA is inequitable.

In an attempt to address some of the issues outlined above, Airspace Flow Programs (AFPs) were introduced in 2006. AFPs are designed to target only flights scheduled to fly through the FCA. Several steps are involved in issuing an AFP. First, the Federal Aviation Administration (FAA) creates a Flow Evaluation Area (FEA). Through the creation of an FEA, traffic controllers highlight a specific region of airspace that is predicted to be affected by a weather event in the next several hours. At this point, airlines can request to reroute around this area. Next, the FAA turns the FEA into an FCA. Any flight scheduled to fly through the FCA can choose to reroute around the FCA or participate in the AFP. If a flight chooses to participate in the AFP, the FAA will assign the flight a route and departure time. Departure times are allocated based on a ration by schedule (RBS) method. In this method, flights are given arrival slots at the FCA in the order in which they were originally scheduled to enter the FCA. Once the AFP is in place, the FCA is monitored and departure times and en route controls, such as miles in trail, are used to meter the arrival rate of flights at the FCA [34]. Once the weather clears and the FCA capacity returns to its nominal level, the AFP is lifted.

As mentioned earlier, airlines have the option to route flights out of the FCA so as not to be given a controlled departure time. Another degree of freedom given to airlines is the option of substitution of flights involved in the AFP. Essentially, airlines may take an FCA arrival slot assigned to a certain flight and reallocate that slot to another flight [34]. This gives airlines some control over their flight schedule and the ability to reorder flights based on their relative priority. The combination of RBS and flight substitution is the current airline
preferred form of equity. A detailed discussion of other rationing techniques and their affects on scheduling solutions is given in [26].

1.1.3 Weather and Capacity Predictions

Currently, the maximum flow through an FCA is set by air traffic controllers based on their experience controlling traffic in that region during similar weather events. A more systematic use of probabilistic forecasts of the capacity of an FCA may allow for better utilization of airspace and greater FCA throughput. In this section, current work in capacity predictions is summarized.

Some probabilistic weather forecasts exist and are a current topic of research. For instance, the National Convective Weather Forecast (NCWF-2) provides a probabilistic forecast for the NAS with a 0 to 2 hour range. A version of this forecast, NCWF-6, extends the forecast to 6 hours. The Collaborative Convective Forecast Product (CCFP) is a 2, 4, and 6 hour forecast product that is probabilistic in the sense that each predicted weather polygon is assigned a low or high confidence [2].

Weather forecasts must also be converted into predictions of flight deviations to be useful for traffic flow management (TFM) decision making. The Convective Weather Avoidance Model (CWAM) [20] uses deterministic weather forecasts from the Corridor Integrated Weather System (CIWS) [21] to predict pilot deviations around weather. More specifically, CWAM produces polygons containing weather and specifies the probability that pilots will deviate around each polygon.

Another aspect of determining airspace capacity due to weather is finding not just the probability of flight deviations, but also a deterministic or probabilistic forecast of the resulting amount of traffic flow that can pass through a particular type of airspace. This sort of computation involves the structure of the airspace that is not passable due to weather. For example, the use of a scenario-based stochastic weather model to generate expected capacity and a probability distribution of capacity for an FCA is proposed in [36]. A capacity forecast and probability are associated with the weather avoidance routes generated for each weather scenario. Thus probabilistic capacity profiles for weather avoidance routes through the FCA are generated from probabilistic weather forecasts. As suggested by the authors of [36], other route generation algorithms, such as the Flow-Based Route Planner [39] which generates routes through an FCA with deterministic constraints or the scanning method [30], could be extended to work with stochastic weather models.
1.2 Literature Review

1.2.1 Research in Air Traffic Management Problems

Much of the current ATM research is focused on the development of algorithms for scheduling ground and airborne delay for flights traveling through capacity constrained airspace or to airports with arrival rate constraints. This work relies heavily on parallel research being done in airspace capacity prediction. With the inherent uncertainty in weather predictions, and thus in capacity predictions, an important aspect of any scheduling algorithm is the way in which this uncertainty is dealt with. A selection of methods are discussed here.

Bertsimas and Stock Patterson present a method for air traffic flow management (ATFM) in the presence of en route airspace capacity constraints in [16]. A flight-level model is used and ground delay and airborne delay are used as the control input. The control design objective is to minimize the weighted sum of ground and airborne delay while satisfying capacity constraints. They assume a deterministic capacity model and note that more work must be done to account for the uncertainty in en route capacities dependent on weather. The drawbacks of this method are that, using a flight-level model, this method may be computationally intensive.

Krozel et al. also use a flight-level model in [31] to solve a capacity constrained air traffic flow problem using ground delay, airborne delay and routing as control inputs. They explicitly examine the effects of uncertainty in capacity constraint forecast on the performance of their proposed algorithm. They assume that the weather forecast will be accurate for some amount of time, beyond which the forecast is uncertain. The authors propose repeating the scheduling method presented at a fixed frequency to ensure that the realized airspace capacity is not exceeded. In testing their algorithm in simulation, they use more or less severe weather predictions, as compared to the actual weather, beyond the actual forecast horizon to obtain a rough measure of the robustness of their algorithm to uncertainty in weather predictions.

A deterministic capacity forecast is also assumed by Mukherjee et al. in [37]. This method makes use of a flight-level model with the control design problem formulated as an integer program. Pre-departure reroutes and departure delays are used as control input. Airborne delay is generally considered to be more costly than ground delay. An optimal solution would utilize ground delay over more expensive airborne delay to impose the required delay in order to satisfy deterministic capacity constraints. Thus, airborne delay is not included as a control parameter in this work. Not including airborne delay as a control input significantly
reduces the number of decision variables, and thus the resulting LP can be solved faster. Using the control input generated by this method, capacity constraints would likely be violated since the predicted constraints may not match the realized constraints. If, as proposed in [31], this algorithm were to be rerun as new constraint predictions become available, there is still the potential for constraint violation.

An aggregate flow model for use in ATFM research was introduced by Menon et al. in [35]. A control design technique is also presented in this work, with the objective of matching a network outflow rate (that is, a landing rate at a destination airport) using ground and airborne delay as the control input. Linear quadratic regulator theory was used to design the controller. However, the resulting controller does not ensure that the closed loop system remain positive. A system is a positive system if the state of the system is non-negative. Additional constraints must be applied \textit{a posteriori} to the resulting controller to ensure positivity. In addition, this control design approach uses only flow rates as a control parameter. It assumes that the network’s routing parameters, which specify how the flow out of each section is redistributed among others, are fixed and suggests that appropriate instantaneous values of these parameters can be found using a simulation tool such as the Future ATM Concepts Evaluation Tool (FACET) [17].

In [32], Le Ny and Balakrishnan present an aggregate model designed to integrate smoothly with the current ATM system. Specifically, it makes use of the current division of airspace in the NAS into sectors and emphasises distributed control techniques. The controller design method is a Max Weight policy. Two separate methods are presented, one which uses airspace flow rates as the control input with no routing and a second method which uses both airspace flow rates and routing as control inputs. The control design objective is to maximize the rate of decrease of a quadratic function of the state which can be thought of as emptying the network as fast as possible. Capacity constraints are considered, however not strictly enforced. Airspace capacity constraints are set by human air traffic controllers based on the perceived difficulty of routing aircraft through a particular region of airspace given the traffic patterns, weather conditions and other factors. Thus, sector capacity is not an exact, clearly defined number. The authors suggest that airspace sector counts can be compared to capacity constraints after a solution has been generated and then determine whether or not the solution generates an acceptable sector count time history. Sector use can be adjusted by setting a higher cost for flights traveling through certain regions of airspace.

Control techniques which make use of an aggregate model generate aggregate control parameters. These parameters are departure rates, flow rates out of and between sections of airspace and routing parameters. When implemented in the NAS (or a simulation of the NAS) these aggregate control parameters must be translated into flight-level control parameters. This process is referred to as disaggregation and is the focus of
Sun et al. [43]. A discrete time aggregate control design technique is presented. Departure delays and airborne delays are used as control input. Airspace capacity constraints and flow rate constraints between regions of airspace are enforced. A variety of control design objectives can be incorporated using this framework, including: minimization of the difference between actual section capacity profile and desired section capacity profile, minimization of total flight time, and minimization of departure delays. The disaggregation method involves rounding the aggregate control input and choosing particular flights to delay.

1.2.2 Aggregate Models for Air Traffic Flow Management

A sampling of control design techniques presented in the literature were discussed in Section 1.2.1. These methods included both flight-level methods and aggregate control techniques. The control design techniques presented here are aggregate control techniques. An overview and discussion of aggregate models developed for ATFM research is given in this section.

A current trend in ATFM research is the use of aggregate models to represent air traffic flow throughout the NAS, see [14, 15, 35, 40, 45]. Aggregate models characterize the flow of groups of aircraft, rather than keeping track of the dynamics of each individual aircraft, as is done in a Lagrangian model. Keeping track of the dynamics of each individual aircraft requires a set of differential equations for each aircraft. The complexity of a Lagrangian model will grow with the number of aircraft. A major advantage of using an aggregate model is that it provides reduced complexity. The complexity of an aggregate model is based on the interconnected network of airspace sections and not on the number of aircraft in the system. Additionally, with the push from the JPDO towards the control of air traffic flows rather than the control of individual aircraft, aggregate models are a natural choice.

Air traffic can be modeled as a positive compartmental system. A compartmental system is composed of a finite number of subsystems, or compartments, which exchange material. Conservation laws describe the flow of material between compartments. The state of the system represents the amount of material in each compartment, and thus must take only non-negative values. A system whose state takes only non-negative values is known as a positive system. The notions of positivity and conservation will be defined formally in Section 2.2.2.

Menon, et al., introduced an aggregate flow model for ATFM in [35] which is based on control volumes and conservation laws. This model is a positive conservative system in which the compartments of the system are control volumes representing regions of airspace. Flow out of a control volume, in this case, a section of airspace, is based on the density of aircraft in the section and the speed of the aircraft. The individual
identity of each aircraft is lost in this type of model, thus the state of each section is simply the number of aircraft in the section. Sections are linked together to represent routes between airports across the NAS. The dynamics of the flow within this network is represented as a linear discrete time system in which conservation of aircraft dictates the connection between sections through the constraint that all aircraft exiting a given section must enter some subsequent section(s). The aircraft in each section are presumed to travel at the same speed and along the same path. The flow rate out of a given section of airspace is proportional to the spatial density of aircraft and the average speed. The state of each section can then be calculated from the state of the section at the previous time step and the number of aircraft leaving and entering the section in one time step.

The control parameters available for this model are the aircraft departure rates from certain airports and controls within each section. The airport departure rates are the inputs to the linear system. The control parameter available in each section is recirculation. Recirculation refers to modeling some aircraft that would naturally leave the section as re-entering that same section. Recirculation effectively reduces the outflow of a given section. Physically, this control input can be realized through speed changes, varying the path length followed by each aircraft or by placing some aircraft into holding patterns. In practice, one would like to minimize the number of aircraft that are in holding patterns and the length of time of these holding sequences, so this type of control is used only after other options have been exhausted.

The aggregate model for ATFM introduced in [35] assumes fixed linear section outflow rates. That is, it is assumed that each section in the network has an average traversal time and the outflow of the section is linear in the number of aircraft in the section. The authors of [33] point out that, although the outflow of a section of airspace will increase as the density of traffic increases, there is an upper bound on the outflow rate. At low density, flights are allowed to traverse a given section of airspace at their nominal speed and thus more aircraft in a given section results in a greater outflow rate from that section. At low traffic density, it is reasonable to assume a linear relationship between the number of aircraft in the section and the outflow rate of that section. However, as traffic through a given section increases, flights must be separated for safety considerations, thus reducing the outflow rate of the section compared to the purely linear estimate. A nonlinear, saturating outflow model is proposed in [33] to more accurately capture this saturating effect in dense traffic problems.

Another class of aggregate model is the cell transmission model, presented by Sun and Bayen in [42]. This is a discrete space-discrete time aggregate model. Like the Menon model, this model is control volume based, with airspace divided into sections. In this model, section sizes and the time step are chosen such that all
aircraft in a given section will exit that section in a single time step. This is a fundamental difference from the Menon model in which flow rates out of a section are calculated based on spatial density of aircraft and average speed. Given that the size of airspace sections is coupled to the time step size, larger regions of airspace, such as sectors, are composed of some number of these smaller control volumes. Control input available is airborne holding modeled as holding some aircraft in a given section. Using this type of model, the dynamical system describing air traffic flow can be formulated as linear constraints in either a mixed integer linear program or an LP.

In [14], a third family of aggregate models is introduced. Bayen et al. use an aggregate model in which airspace is divided into airways and the density of aircraft along each airway is modeled using partial differential equations (PDEs). The density of aircraft is a function of position along the airway and time. Conservation requirements are used in the derivation of the PDEs. Airways are linked together in graphs to represent an interconnected network of airspace. When arranging these airways into networks, the PDEs used to describe each airway are coupled by boundary conditions. One notable advantage of the method presented in [14] is that density constraints can be satisfied by the controller. When dealing with congested airspace, this is an important aspect of the system that must be monitored and, ideally, controlled.

Aggregate models were chosen for this work in order to focus on computationally efficient control design techniques which are concerned with the control of air traffic flows. Flow based methods will scale well as the number of flights involved in ATM problems increases over the coming years. An aggregate model which is a continuous time analog to the Menon model [35] is used in this work. This aggregate model was chosen because it lends itself to traditional control design techniques for positive systems.

1.3 Contributions of this Work

The work presented here focuses on three issues. The first problems considered are that of using routing control to minimize a measure of delay or to satisfy time varying capacity constraints. The LP Routing (Integral) and LP Routing (Capacity) methods are proposed as solutions to these two problems, respectively. The third problem considered is that of scheduling delay to satisfy uncertain time varying flow rate constraints. The SMC method, proposed as a solution to this problem, uses no constraint forecast and reacts in real time to flow rate constraints as they are realized.

These methods are discussed based on the problems addressed, control parameters used and constraints incorporated in Section 1.3.1. The control theory and techniques used to approach each of these problems are introduced in Section 1.3.2.
1.3.1 Problems Addressed

In the following paragraphs, the three proposed control design techniques are compared based on the method used to account for capacity forecast uncertainty, mathematical model used in the controller design, control inputs used, and control design objective.

The uncertainty of weather and capacity forecasts are dealt with in a variety of different ways. SMC is a tactical method designed to satisfy time varying capacity constraints. Using this method, no prediction of capacity constraints is required. Instead, the control input is adjusted in real time to adjust to the capacity constraints as they are realized. LP Routing (Integral) and LP Routing (Capacity) make use of a deterministic capacity forecast. These methods can be run repeatedly throughout the day as updated capacity forecasts become available. We provide a method to incorporate uncertainty in average airspace traversal time as part of the LP Routing (Capacity) method. Airspace traversal time, along with airspace capacity, could decrease when weather affects a given region of airspace. Incorporating this uncertainty into the model and solution is thus a method for incorporating the inherent uncertainty of weather forecasts into the problem of interest.

The methods presented here use continuous time aggregate flow models. The basic model is a continuous time analog to the Menon model [35]. Several variants of this model are used and described in more detail in Chapter 2. Control parameters available when using an aggregate model are ground delay, airborne delay and rerouting. Various combinations of each of these types of control input are used in the three methods presented here. LP Routing (Integral) and LP Routing (Capacity) use routing as the primary control input. Recirculation is equivalent to introducing airborne delay, thus airborne delay is also a control input for these methods. SMC makes use of delay as the control input. In the development of the SMC controller, only airborne delay is considered, although extending these results to include ground delay would be straightforward.

With control input specified at the aggregate level, i.e. as flow rates and routing parameters, the problem of disaggregation arises when applying an aggregate control scheme to a realistic simulation of the NAS such as FACET. FACET is a software tool developed at NASA Ames Research center designed to simulate flights in the NAS [17]. Disaggregation is the method of taking control specified at the aggregate level and translating this into flight-level control commands. The SMC method has been implemented in FACET. This method has an elegant disaggregation scheme in that all flights are given the same control command. Details and results of the implementation of the SMC method in FACET are given in Chapter 6.
Design objectives vary for each of these methods as well. The objective of *LP Routing (Integral)* is the minimization of delay. An infinite planning horizon is used and routing parameters are found which minimize a measure of delay and satisfy integral constraints. Satisfying time varying capacity constraints is the focus of *LP Routing (Capacity)*. In *LP Routing (Capacity)*, routing parameters are selected in order to drive the state of the system below some time varying capacity constraints. This method makes use of a deterministic capacity forecast over a finite time horizon and can be re-run throughout the day to incorporate updated capacity constraint predictions as they become available. In addition to designing a control technique for a fixed linear outflow model, routing control techniques are developed for models which incorporate uncertainty in section traversal time and allow for nonlinear outflow rates. The *SMC* method addresses the problem of satisfying a time varying flow rate constraint at the outlet of the network. The network can be thought of as emptying into an FCA or an airport, so this method can be applied to an AFP or to meter arrivals at a congested airport.

### 1.3.2 Control Techniques

In this section, the control design techniques used to approach the problems of Section 1.3.1 are discussed. Details of these proposed techniques are supplied in the following chapters.

In Chapters 3 and 4, we focus on routing as the control input. A fixed linear outflow model is used in the development of the *LP Routing (Integral)* method which minimizes a measure of delay subject to integral constraints. An LP based method is developed to solve this class of problems. The design objective of *LP Routing (Capacity)* is to satisfy time dependent piecewise constant capacity constraints on the states of the network while maximizing the throughput of the network over a finite time horizon. An LP method to design time varying routing parameters to achieve this objective is described in Section 4.3. A fixed linear model is used to describe the flow of traffic through the network. Under these routing parameters, the resulting system is positive, conservative and exhibits the desired interconnection. This method is then modified for use with an uncertain linear time varying outflow model and a nonlinear outflow model in Section 4.4 and Section 4.5, respectively. In Sections 4.3 through 4.5, control design techniques are developed for single destination problems, that is, all aircraft modeled have the same destination. Each of these control design techniques can be extended to multiple destination problems. As an example, the fixed linear model control design technique is extended for application to a multiple destination problem in Section 4.6.

The problem of satisfying flow rate constraints is addressed by the method *SMC* and discussed in Chapter 5. Using a sliding mode control approach, controllers are designed which, using only knowledge of the number
Table 1.1: Table summarizing some of the key properties of each of the three control techniques presented here (first three lines of the table) and the control techniques presented in Section 1.2.1. Note that Sun et al. [43] are able to incorporate a variety of control design objectives in the model framework they present. These objectives are not captured adequately by the list of objectives selected for this table. See Section 1.2.1 for the specific control design objectives used in this work.

<table>
<thead>
<tr>
<th>Model</th>
<th>Model</th>
<th>Forecast</th>
<th>Constraint</th>
<th>Objective</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Input</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Forecast</td>
<td>Constraint</td>
<td>Input</td>
</tr>
<tr>
<td>Aggregate Model</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Aircraft-level</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Continuous Time</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Discrete Time</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deterministic</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Not Required</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Capacity</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Flow Rate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Min Weighted Delay</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Max Throughput</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Match Flow Rate</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Airborne Delay</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ground Delay</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Routing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

List of symbols:

✓ Indicates that the corresponding model, constraint forecast, constraint, control design objective or control input is incorporated by a particular control design method.

□ Indicates that, although not explicitly incorporated in the published work, the method could easily be extended to incorporate the indicated item.

■ Indicates that a particular constraint is not explicitly enforced, however it can be incorporated through the cost.
of aircraft in their own section and the flow out of the network, match a maximum allowable throughput
profile between the source and sink, when possible. One of the advantages of this control strategy is that,
since it is not required that each section have knowledge of the states of all other sections, it requires only
a limited amount of communication between sections. Exploiting the particular compartmental structure of
the system, proofs of asymptotic stability of the control scheme are given, along with an upper bound on
the time needed for the tracking error to fall below a prescribed level.

One of the notable properties of the SMC technique is that it does not require a prediction of future flow rate
constraints. Instead, the control algorithm reacts in real time to satisfy changing flow rate constraints. In
an ATM application, this control technique can be used to react in real time to changing weather conditions
which change the allowable flow rate into affected regions of airspace. Given that the primary control
parameter available using this technique is airborne delay, it would only be advantageous to use this method
on top of a more strategic control method which uses an inherently inaccurate flow rate constraint forecast.
This algorithm can then be used in real time to make small adjustments to flow rates through airborne
holding so that the realized flow rate constraints are satisfied. This control technique has been applied to
a realistic simulation of the NAS in FACET with details of the problem and simulation results given in
Chapter 6.

Each of the proposed control design techniques are discussed in greater detail in the following chapters.
Since all of the methods presented here use some form of continuous time aggregate model, some details
and general discussion of this type of model is provided in Chapter 2. The specific variant of the aggregate
model used for each method is given in more detail in their respective chapters.

Properties of each of these control techniques, along with control techniques discussed in Section 1.2.1, are
summarized in Table 1.3.1.

1.4 Organization of this Work

The work presented here could be organized in a variety of ways. Focusing on air traffic management
applications, this work could be organized based on the problem solved. Alternatively, the methods could
be grouped based on the specific aggregate model used. In order to bring out the similarities between many
of the control design techniques used and the derivation of these techniques, we chose to organize the work
based on control design technique. Each method uses an aggregate model to describe the flow of air traffic
through a network of airspace. In Chapter 2, we describe the general properties and the basic mathematical
details of continuous time aggregate models which are relevant to all the control design techniques. After
this chapter, we present the first of the LP based methods for routing control, the *LP Routing (Integral)* method, in Chapter 3. In Chapter 4, we present the *LP Routing (Capacity)* methods used to design routing methods to satisfy airspace capacity constraints. The methods of those chapters each make use of a different type of aggregate model. Although these models are quite different, the control design methods are very closely related. The SMC method is presented in Chapter 5 and the details and results of the application of this method in a realistic simulation of the national airspace system is given in Chapter 6. Concluding remarks are given in Chapter 7.

### 1.5 Notation

We are often concerned with vectors, scalars and elements of matrices which are indexed over some range. Therefore, for every positive integer $n$, we define $[n] = \{1, 2, \ldots, n\}$. We denote the cone of entry-wise non-negative vectors of dimension $n$ by $\mathbb{R}_n^+$ and write “$x \geq 0$” to mean that vector $x$ belongs to that set, and “$x > 0$” to mean that it belongs to its interior, i.e., that every entry of vector $x$ is strictly positive. Likewise, $\mathbb{R}_{+}^{n \times m}$ will denote the set of all $n \times m$ matrices with non-negative entries. A real matrix $M$ is called a *Metzler* matrix if its off-diagonal elements are non-negative, i.e., $M_{ij} \geq 0$, $\forall i, j \in [n]$, such that $i \neq j$. For all $i \in [n]$, $e_i$ represents the $i^{th}$ canonical basis vector of $\mathbb{R}^n$. 
Chapter 2

Aggregate Model Details

In this work, we make use of several variants of a continuous time aggregate model. The basic properties and mathematical details of the general form of these models is given in this chapter. Each problem and solution method will include various constraints or assumptions on the model used. Each chapter or section introducing a new problem or control design technique has a section titled Continuous Time Aggregate Model. In these sections, the reader is referred back to this chapter for the general model details. The specific constraints or assumptions on the model for the current problem is stated and the refined version of the general model relevant in each specific chapter or section is presented.

2.1 Basic Network Properties

In this work, we focus on control design for positive compartmental systems. Such systems represent the dynamics of the flow of material through an interconnected network of reservoirs. The dynamics are derived from conservation laws and the underlying interconnection of the network [13]. We are particularly interested in the use of these models to describe air traffic flows. However, these models have been used to describe a variety of different systems including automobile or aircraft traffic flow, job-balancing in computer clusters [23], or any system of connected reservoirs with natural constraints, such as irrigation networks [19].

These aggregate models are compartmental. They are control volume based and are used to describe the flow of aircraft through a network of these volumes, which we refer to as “sections.” The state of the system is the number of aircraft in each section. Since the state represents a physical quantity, the state must only take on non-negative values.

Each model describes the flow of aircraft between sections. Aircraft enter the network at one or more sources,
and exit the network through sinks. In the air traffic management application, sources can be thought of as departure airports or regions of airspace not captured by the model. Sinks can represent arrival airports or other regions of airspace not captured by the model. The systems must be conservative, that is, aircraft flows modeled as leaving one section of the network, must either enter another section of the network or exit the network at a sink. In addition to conservation, the networks must be set up to be physically meaningful. Since we are dealing with networks that describe the flow of aircraft through airspace, the interconnection of sections must be defined in such a way that flow from one section to a subsequent section is physically possible. That is, adjacent sections in the network must represent regions of airspace that share boundaries, allowing flow between these regions of airspace. In order to ensure stability of the system, that is, that all aircraft eventually exit the network, the network must be outflow connected. That is, there must exist at least one path from each section in the network to a sink.

In this work, we focus on the development of control design techniques for positive compartmental networks. We assume that the given structure of the network satisfies the interconnection requirements of the network of airspace modeled and that the networks are outflow connected. Care must be taken in the derivation of the control design to ensure positivity, conservation and, when appropriate, stability. The specific mathematical constraints required to ensure these properties will be discussed in more detail after the presentation of the general model.

2.2 Continuous Time Aggregate Model

The work presented in Chapters 3, 4, and 5 make use of the continuous time aggregate model of air traffic flows described in this section. We consider positive systems which can be described as a network of sections through which aircraft can travel. In the development of this model we make the following assumptions. It is assumed that the number of aircraft in a section can take on any nonnegative real value. Aircraft are evenly distributed throughout each section and move at a constant speed.

The systems of interest are described by a directed graph with \( n \) vertices. Each vertex \( i \) of the graph corresponds to a different subsystem or section, whose state \( x_i(t) \) takes values in \( \mathbb{R}_+ \) and represents the number of aircraft present in the section at time \( t \geq 0 \). The presence of oriented edge \((i, j)\) in the edge set of the graph means that aircraft can travel out of section \( i \) into section \( j \) at time \( t \). This conforms with typical notation in air traffic management literature, for example [32, 43]. The flow rate of aircraft out of each section \( i \) may depend on the state of that section and other parameters that vary with time and is denoted by \( f_i(x_i(t), t) \). In general, the outflow of each section \( i \) have the following properties:
1. \( f_i(0, t) = 0 \),

2. \( f_i(a, t) \geq 0 \) for all \( a \geq 0 \).

The first property ensures that empty sections have no outflow. The second property ensures that aircraft flow through the network in the prescribed direction, since a negative outflow would indicate flow in the reverse direction, which is not permitted. The function \( f \) will take on various forms throughout this work including, linear time invariant, linear parameter varying, and nonlinear.

The fraction of the outflow of section \( i \) routed to section \( j \) is indicated by the parameter \( \beta_{ij}(t) \). To simplify notation, we will sometimes refer to these routing parameters collectively as \( \beta(t) \) or as \( \beta \) when the values of the routing parameters are constant.

We denote by \( O_i \) the set of sections into which the flights in section \( i \) can flow. When this set is empty, (i.e., when vertex \( i \) is a sink of the graph), we say that \( i \) is a “final section.” The set of all final sections is denoted by \( S_F \). These sections can be thought of as leading to landing airports or other regions of airspace not captured by the model. We assume that the network has at least one sink and thus \( S_F \neq \emptyset \) and that there exists at least one path from each section \( i \) to a section in \( S_F \). That is, there exist vertices \( i = i_1, i_2, \ldots, i_p \) such that \( i_{l+1} \in O_{i_l} \) for \( l = 1, 2, \ldots, p - 1 \), \( i_l \in [n]\setminus S_F \) for all \( l < p \), and \( i_p \in S_F \). When \( i \notin S_F \), we assume that \( i \in O_i \), that is, aircraft can be recirculated back into the section that they have just exited. Such recirculation represents holding or the slowing down of flights moving through that section.

Flights can enter any section in the network from sources outside of the network. These sources can be either airports or regions of airspace not included in the model. Let \( S \) be the number of sources supplying the network. The output of source \( s \) is represented by \( d_s(t) \) for all \( s \in [S] \). The fraction of \( d_s(t) \) routed into section \( i \) is denoted by \( b_{si} \), with \( 0 \leq b_{si} \leq 1 \) for all \( i \in [n] \), and \( \sum_{i=1}^{n} b_{si} = 1 \) for all \( s \in [S] \).

Overall, the state \( x_i(t) \) of section \( i \) thus satisfies the differential equation

\[
\dot{x}_i(t) = -f_i(x_i(t), t) + \sum_{j: i \in O_j} \beta_{ji}(t)f_j(x_j(t), t) + \sum_{s=1}^{S} b_{si} d_s(t). \tag{2.1}
\]

Equation (2.1) expresses the conservation of material in section \( i \) by equating its time derivative to the sum of incoming flows minus the sum of outgoing flows. The dynamics (2.1) of all \( n \) sections can be summarized...
by the linear system

\[ \dot{x}(t) = F(x(t), t) + B^T d(t) \]
\[ x(0) = x_0 \]

(2.2)

where we have introduced

\[ x(t) = \left( x_1(t) \ldots x_n(t) \right)^T, \]
\[ d(t) = \left( d_1(t) \ldots d_S(t) \right)^T, \]
\[ B = \begin{bmatrix} b_{11} & \ldots & b_{1n} \\ \vdots & \ddots & \vdots \\ b_{S1} & \ldots & b_{Sn} \end{bmatrix}. \]

A simple example network is given in Figure 2.1.

We often assume that the outflow of each section depends linearly on the state of the section. In such instances, each section has an associated traversal time \( \tau_i(t) \) and the outflow of section \( i \) is

\[ f_i(x_i(t), t) = \frac{x_i(t)}{\tau_i(t)} \]

and we write the dynamics of the \( n \) section network as

\[ \dot{x}(t) = A(\beta(t), t)x(t) + B^T d(t) \]
\[ x(0) = x_0 \]

(2.3)

where

\[ A(\beta(t), t) = A_0 + \sum_{i=1}^n \sum_{j \in O_i} \frac{\beta_{ij}(t)}{\tau_i(t)} e_j e_i^T, \]

(2.4)

and \( A_0 = \text{diag} \left( -\frac{1}{\tau_1(t)}, \ldots, -\frac{1}{\tau_n(t)} \right) \).

In an ATFM application, the assumption that aircraft are evenly distributed throughout each section is reasonable when considering dense air traffic. The assumption that aircraft move at a constant speed may seem restrictive considering that speed changes or increasing the length of the path which aircraft fly through a given region of airspace are commonly used control actions in ATFM. However, if the constant speed used to generate model parameters is the fastest speed that the aircraft can fly, this speed can be reduced by
recirculation. That is, if the routing parameter $\beta_{ii}(t)$ for some section $i$ is positive, this effectively reduces the outflow rate of section $i$, which can be achieved through speed reduction or increasing the path length.

### 2.2.1 Control Parameters

Using this basic aggregate model, we make use of two different types of control input. One class of control input is airborne delay modeled as recirculation. The second type of control input used is routing. In this case, the routing parameters, $\beta(t)$, in system (2.2) are the control variables.

In Chapter 5, airborne delay is used as the control input. Section outflow is linear with fixed traversal times $\tau_i$ and routing parameters are assumed to be fixed and given as part of the network model. The connectivity of the network is defined by the fixed routing structure, that is, $j \in O_i$ if and only if $\beta_{ij} > 0$. In this case, the control input is the recirculation rate, that is, the control input is modeled as taking part of the natural outflow of the section and recirculating it back into that same section, thus effectively reducing the outflow of the section. As mentioned earlier, airborne delay can be implemented through speed reduction, increasing the length of the path flown through a given section of airspace, or, as a last resort, airborne holding. We define the recirculation control input as $u_i(t)$. With parameters defined in this way, we can define the controlled outflow rate of section $i$ as $\frac{x_i(t)}{\tau_i} - u_i(t)$. The dynamics of section $i$ can then be given by

$$
\dot{x}_i(t) = -\frac{x_i(t)}{\tau_i} + u_i(t) + \sum_{j: i \in O_j} \beta_{ji} \left( \frac{x_j(t)}{\tau_j} - u_j(t) \right) + \sum_{s=1}^{S} b_{si} d_s(t).
$$

In Chapters 3 and 4, routing parameters $\beta(t)$ are used as the control input. As discussed in the development
of the continuous time aggregate model in Section 2.2, it is assumed that \( i \in O_i \) for all \( i \notin S_F \), that is, recirculation is allowed in all sections of the network that are not final sections. A value of \( \beta_{ii} > 0 \) indicates that some fraction of the outflow of section \( i \) is routed back into section \( i \), which is the same as the airborne delay control described above. Thus, as formulated here, routing control can be used to both dictate the way in which flows of aircraft are routed through the network and add airborne delay.

### 2.2.2 Basic Control Design Objectives

In Chapters 3, 4 and 5 we address a variety of different control problems using continuous time aggregate models to describe the flow of air traffic. Although these problems have different constraints and performance objectives, it is important that any solution to an air traffic flow management problem using an aggregate model satisfy some basic control design objectives. We must ensure that the closed loop system is positive, conservative and, when the control design problem has a finite time horizon, internally stable. These objectives are described in more detail below.

In Chapters 3 and 5 we address control design problems with infinite time horizons. The problems addressed in Chapter 4 have finite time horizons. For the infinite horizon problem, we require the closed loop system to satisfy the following constraint

**Stability:** System (2.2) is internally stable.

In addition, the following two constraints must hold for both the infinite horizon and finite horizon problems.

**Positivity:** System (2.2) is internally positive, that is,

\[
x_0 \geq 0 \text{ and } d(t) \geq 0 \quad \forall t \geq 0 \Rightarrow x(t) \geq 0, \quad \forall t \geq 0.
\]  

\[(2.6)\]

**Conservation:** For all \( t \geq 0 \),

\[
\beta_{ij}(t) \geq 0, \quad \forall \ i, j \in [n],
\]  

\[
\beta_{ij}(t) = 0, \quad \forall \ j \in [n] \setminus O_i, \forall \ i \in [n],
\]  

\[
\sum_{j \in O_i} \beta_{ij}(t) = 1, \quad \forall \ i \in [n] \setminus S_F.
\]  

\[(2.8)\]  

\[(2.9)\]
Subsequently, the group of constraints Stability, Positivity, and Conservation will be referred to by the acronym SPC and Positivity, and Conservation will be referred to by the acronym PC.

Physically, the Conservation requirement expresses that material leaving every non-final section must be conserved. Note that the Conservation constraint (2.9) is enforced only for $i \in [n]\backslash S_F$, which physically means that material is allowed to flow out of the network through final sections. The Stability requirement guarantees that the network is eventually emptied of all material, while the Positivity requirement ensures that each coordinate of state $x$, which represents the quantity of material present in a section, is non-negative at all times.

The following sufficient and necessary conditions, which follow directly from Theorem 2 in [27], will be useful to ensure that the Positivity requirement is satisfied.

**Theorem 1** Let $A(\cdot), B(\cdot)$ be continuous matrix-valued maps with $A(t) \in \mathbb{R}^{n \times n}$, $B(t) \in \mathbb{R}^{S \times n}$ for all $t \geq 0$. Then, the linear time-varying system

$$\dot{x}(t) = A(t)x(t) + B^T(t)d(t)$$

satisfies the internal positivity condition (2.6) if and only if

(i) $B(t) \in \mathbb{R}_{+}^{S \times n}$ for all $t \geq 0$,

(ii) matrix $\int_0^t A(s)ds$ is Metzler for all $t \geq 0$.

Given our assumptions on $B$, it readily follows from Theorem 1 that constraint (2.7) is a sufficient condition for system (2.2) to be internally positive.

When a network is given with fixed routing parameters, as in Chapter 5, it is assumed that the routing parameters have been chosen such that the uncontrolled system is conservative. That is, it is assumed that when given, routing parameters satisfy (2.7) through (2.9). When routing parameters are used as the control input in Chapters 3 and 4, care is taken to ensure that the designed values of the routing parameters satisfy (2.7) through (2.9).
Chapter 3

LP Based Routing Design for Systems with Integral Objective and Constraints

3.1 Motivation and Problem Description

In this chapter, we focus on linear techniques for the design of static routing parameters for single destination networks with the goal of minimizing total delay while satisfying additional delay constraints. Delay in an air traffic network is costly, with costs arising, e.g., from missed connections and extra fuel consumption associated with airborne delays. For this reason, many algorithms for air traffic flow management have as the objective, at least in part, the minimization of delay costs (see for example, [16] and [31]).

We first present an LP based method to design routing parameters to minimize a measure of total delay. We prove that this method minimizes delay over all choices of routing parameters which ensure that the closed loop system is stable, positive, conservative and exhibits user specified interconnection. We add constraints that ensure that additional delay constraints are satisfied while minimizing total delay. The control design techniques developed in this chapter are also presented in [10] and [12].

3.2 Continuous Time Aggregate Model

The model used to describe the flow of aircraft through the network of airspace of interest is the continuous time aggregate model presented in generality in Section 2.2. In this chapter, we use a linear outflow model for each section $i$ of the network with time invariant traversal times $\tau_i$, that is

$$f_i(x_i(t), t) = \frac{x_i(t)}{\tau_i}.$$
The choice of routing strategy is the control input. The class of control problem of interest in this chapter is concerned with the design of constant routing strategies \( \beta(t) = \beta \) for all \( t \geq 0 \) over an infinite time horizon.

With these assumptions, the dynamics of each section of the network can be written as

\[
\dot{x}_i(t) = -\frac{x_i(t)}{\tau_i} + \sum_{j \in \mathcal{O}_i} \beta_{ij} \frac{x_j(t)}{\tau_j} + \sum_{s=1}^{S} b_{si} d_s(t),
\]

for all \( i \in [n] \). The dynamics of the \( n \) section network is then described by the following linear time invariant system

\[
\dot{x}(t) = A(\beta)x(t) + B^T d(t)
\]

\[
x(0) = x_0
\]

where

\[
A(\beta) = A_0 + \sum_{i=1}^{n} \sum_{j \in \mathcal{O}_i} \frac{\beta_{ij}}{\tau_i} e_j e_i^T,
\]

and \( A_0 = \text{diag} \left(-\frac{1}{\tau_1}, \ldots, -\frac{1}{\tau_n} \right) \).

We will designate the unique solution of (3.2) for a given initial condition \( x_0 > 0 \), inflow profile \( t \mapsto d(t) \), with \( d(t) \geq 0 \) for all \( t \) and routing strategy \( \beta \) as \( t \mapsto x^\beta(t;x_0,d) \). When the choice of \( x_0 \) and \( d \) is clear from context, we will abbreviate this to \( x^\beta \).

### 3.3 Control Objectives

We now rigorously present the class of control problem that will be solved in the remainder of this chapter.

**Problem 1** Let vectors \( w \) and \( \{w_m\}_{m=1}^{M} \) and scalars \( \{\gamma_m\}_{m=1}^{M} \) be given, such that \( w > 0 \), \( w_m \geq 0 \) and \( \gamma_m > 0 \) for all \( m \in [M] \). Let an initial condition \( x_0 \) and inflow profile \( d \) be given. Find a constant routing strategy \( \beta(t) = \beta \) for all \( t \geq 0 \) that minimizes

\[
\int_0^{\infty} w^T x^\beta(t) dt,
\]

subject to the Stability, Positivity and Conservation constraints, referred to collectively as SPC, and

\[
\int_0^{\infty} w_m^T x^\beta(t) dt \leq \gamma_m, \forall \ m \in [M].
\]
Problem 1 is a weighted $\ell_1$ optimal control problem with weighted $\ell_1$ constraints, and can be given different interpretations depending on the actual plant modeled by system (2.2). For example, [41], [24] and references therein proposed the integral objective (3.4) with $w = 1$ (a vector of ones) as a measure of total delay in single destination networks. Alternatively, in load balancing applications, objective (3.4) can be interpreted as the total load experienced by the servers [22]. This kind of performance objective was also considered for abstract linear positive systems in [5], although routing strategies were not used as control inputs.

Before beginning the derivation of the control design technique to solve Problem 1, it is important to note that the Stability and Conservation constraints are not in conflict. Recall the structure of networks considered described in Section 2.2. We assume that the network has at least one sink and thus $S_F \neq \emptyset$ and that there exists at least one path from each section $i$ to a section in $S_F$. That is, there exist vertices $i = i_1, i_2, \ldots, i_p$ such that $i_{t+1} \in O_{i_t}$ for $l = 1, 2, \ldots, p - 1$, $i_t \in [n] \setminus S_F$ for all $l < p$, and $i_p \in S_F$. Given this assumption, the Stability and Conservation constraints are not in conflict. This statement is formalized in the following claim.

**Claim 1** Assume that for each $i \in [n] \setminus S_F$ there exists a path from section $i$ to at least one section $j \in S_F$. Then, there exists a set of routing parameters $\beta$ such that $A(\beta)$ given by (2.4) satisfies SPC.

**Proof 1** By assumption, for each section $i \in [n] \setminus S_F$, there exists a shortest path from $i$ to some $j \in S_F$. Renumber the sections of the network starting with those furthest from a final section, i.e. those with the longest shortest path, arbitrarily ordering sections with the same shortest path length. The new numbering system will be distinguished from the old through the use of primes, for example, $i'$ denotes the index of a section and $S'_F$ indicates the set of final sections under the new numbering scheme. A shortest path from $i'$ to some $j' \in S'_F$ can be described as a series of sections, $i', i'_1, i'_2, \ldots, i'_r, j'$, where $i'_1 \in O_{i'}, i'_2 \in O_{i'_1}, \ldots, j' \in O_{i'_r}$, and $i' < i'_1 < i'_2 < \ldots < i'_r < j'$, for each $i' \in [n] \setminus S'_F$. Indeed, by the optimality principle, every section in the shortest path from $i'$ must have a shorter shortest path, and hence larger index than $i'$. A routing strategy $\beta$ can be designed which satisfies constraints (2.7) - (2.9) by setting $\beta_{i'i'_1} = 1$ and $\beta_{i'j'} = 0$ for all $k' \in [n] \setminus \{i'_1\}$ for each $i' \in [n] \setminus S'_F$. With constraints (2.7) - (2.9) satisfied, the resulting system will be positive and conservative.

Given the numbering scheme and routing strategy described above, for each $i' \in [n] \setminus S'_F$ the corresponding column of $A(\beta)$ given by (2.4) is $\frac{1}{v_i'} e_{i'} + \frac{1}{v_{i'j'}} e_{i'}$. For each $i' \in S'_F$ the corresponding column of $A(\beta)$ is $\frac{1}{v_{i'i'}} e_{i'}$, since the outflow of final sections is not routed into other sections in the network, i.e. $O_{i'} = \emptyset$ for all $i' \in S'_F$. It can be seen that $A(\beta)$ is lower triangular, since every section along the shortest path from $i'$ has an index
larger than $i'$, with values $\frac{1}{\tau_i} < 0$ for each $i' \in [n]$ on the diagonal. These values are also the eigenvalues of $A(\beta)$, thus it can be concluded that $A(\beta)$ is Hurwitz. Under the routing strategy constructed here, $A(\beta)$ satisfies SPC.

3.4 Control Design

In this section, we propose an LP-based design method to solve Problem 1. We start by recalling some useful facts about Metzler matrices.

**Theorem 2** Assume that matrix $A$ is Metzler, then the following statements are equivalent.

(i) $A$ is Hurwitz.

(ii) There exists $\lambda \in \mathbb{R}^n$ such that $A\lambda < 0$, $\lambda > 0$.

In addition, for every vector $v > 0$, there exists a unique vector $\mu > 0$ such that $A\mu + v = 0$.

The equivalence of (i) and (ii) is shown in [6]. It is straightforward to verify the additional remark.

With these results in hand, we are in a position to state our first main result.

**Theorem 3** Let vectors $w > 0$ and $x_0 > 0$ and inflow profile $t \mapsto d(t)$ be given, with $d(t) \geq 0$ for all $t$. Let $v = x_0 + \int_0^\infty B^T d(t) dt > 0$, and $(\mu^*, z^*)$ be an optimal point of the following LP problem:

\[
\begin{align*}
\min_{\mu, z} & \quad w^T \mu \\
\text{subject to} & \quad A_0 \mu + \sum_{i=1}^n \sum_{j \in O_i} \frac{z_{ij}}{\tau_i} e_j + v \leq 0, \\
& \quad \mu > 0, \\
& \quad z_{ij} \geq 0, \quad \forall j \in O_i, \quad \forall i \in [n], \\
& \quad z_{ij} = 0, \quad \forall j \in [n] \setminus O_i, \quad \forall i \in [n], \\
& \quad \sum_{j \in O_i} z_{ij} = \mu_i, \quad \forall i \in [n] \setminus S_F.
\end{align*}
\]

Let $\beta_{ij}^*$ be defined as $\beta_{ij}^* = \frac{z_{ij}^*}{\mu_i^*}$, $\forall i,j \in [n]$. Then,
(i) Matrix $A(\beta^*)$ given by (3.3) is such that constraints SPC are satisfied.

(ii) $w^T \mu^* = \int_0^\infty w^T x^\beta^*(t) dt$.

(iii) For every routing strategy $\beta$ such that constraints SPC hold,

$$\int_0^\infty w^T x^\beta(t) dt \geq w^T \mu^*.$$ 

Note that properties (i), (ii), and (iii) combined imply that

$$w^T \mu^* = \min \left\{ \int_0^\infty w^T x^\beta(t) dt : A(\beta) \text{ satisfies SPC} \right\}.$$ 

**Proof 2** We start by establishing that LP problem (3.6) is feasible and that its value is always attained, which justifies the use of “min” in (3.6a) and the definition of $(\mu^*, z^*)$ as an optimal point of LP problem (3.6). First note that if $\beta$ is such that $A(\beta)$ satisfies SPC and $(\mu, z)$ is defined by

$$\mu := \int_0^\infty x^\beta(t) dt \quad (3.7a)$$

$$z_{ij} := \mu_i \beta_{ij}, \forall \ i, j \in [n], \quad (3.7b)$$

then $(\mu, z)$ is a feasible point for LP problem (3.6). Indeed, $\mu$ is positive and the only solution to $A(\beta) \mu + v = 0$, thus (3.6b) and (3.6c) hold. Also, (3.6d) - (3.6f) hold because of the conservation condition. Since there always exists $\beta$ such that $A(\beta)$ satisfies SPC by Claim 1, we deduce that LP problem (3.6) is always feasible.

Now, let $(\tilde{\mu}, \tilde{z})$ be a feasible point for (3.6). Then (3.6) has the same optimal value as the following LP problem:

$$\min_{\mu, z} \ w^T \mu \quad (3.8a)$$

subject to (3.6b), (3.6d), (3.6e), (3.6f)  

$$w^T \mu \leq w^T \tilde{\mu}, \quad (3.8b)$$

$$\mu_i \geq \tau_i v_i, \forall i \in [n]. \quad (3.8d)$$

This is because $(\mu, z)$ satisfies [(3.6b), (3.6d), (3.6e), (3.6f) and (3.8d)] if and only if it satisfies (3.6b) - (3.6f), and adding condition (3.8c) does not modify the value since $(\tilde{\mu}, \tilde{z})$ is feasible for LP problem (3.6). LP problem (3.8) is feasible, specifically $(\tilde{\mu}, \tilde{z})$ is a feasible point, and has a compact feasible set. Closedness of the feasible set is clear, and boundedness of $\mu$ follows from (3.8c) and (3.8d) since $w > 0$. Boundedness
of $z$ follows from (3.6d) - (3.6f), since $0 \leq z_{ij} \leq \mu_i$ for all $i, j \in [n]$. As a result, the value of LP problem (3.8) is attained for some $(\mu^*, z^*)$ which is also feasible for LP problem (3.6) and, hence, the value of LP problem (3.6) is also attained at $(\mu^*, z^*)$.

Now we can prove (i) - (iii). Clearly, $A(\beta^*)$ is Metzler since $\beta^*_{ij} \geq 0$ for all $i, j \in [n]$. This proves positivity. In addition, because of (3.6b) and the definition of $v$, $A(\beta^*)\mu^* < 0$ and $\mu^* > 0$. Hence, according to Theorem 2, $A(\beta^*)$ is Hurwitz, which proves stability. Finally, the conservation property follows directly from constraints (3.6d)-(3.6f) on $\mu^*$ and $z^*$. This proves (i).

For (iii), recall that if $\beta$ is such that $A(\beta)$ is SPC, $(\mu, z)$ defined by (3.7) is feasible for LP problem (3.6). Furthermore, by definition of $(\mu^*, z^*)$,

$$w^T \mu^* \leq w^T \mu = \int_0^\infty w^T x^\beta(t) dt.$$  

Since this inequality holds for all $\beta$ such that SPC is satisfied, (iii) is proved.

Finally, we prove (ii). To this end let us define $\bar{\mu} = \int_0^\infty x^\beta^*(t) dt$ and, for all $i, j \in [n]$, $\bar{z}_{ij} = \frac{z_{ij} \bar{\mu}_i}{\mu_i} = \beta^*_ij \bar{\mu}_i$.

Since $A(\beta^*)$ satisfies SPC, we can proceed as before to deduce that $(\bar{\mu}, \bar{z})$ is feasible for LP problem (3.6). As a result, $w^T \mu^* \leq \int_0^\infty w^T x^\beta^* (t) dt$. To prove the converse inequality, note that because $(\mu^*, z^*)$ is feasible by definition, constraint (3.6b) is satisfied and, hence,

$$A(\beta^*)\mu^* + v \leq 0. \quad (3.9)$$

Let $\xi(t)$ be the solution to the system

$$\dot{\xi} = A(\beta^*)^T \xi; \xi(0) = w.$$  

Note that this is a positive system (because $A(\beta^*)^T$ is Metzler whenever $A(\beta^*)$ is) and that, because $\xi(0) \geq 0$, $\xi(t) \geq 0$ for all $t$. In addition $A(\beta^*)^T$ is Hurwitz. Now, left multiply (3.9) by $\xi^T$ and integrate to get

$$- w^T \mu^* + \int_0^\infty \xi(t)^T v dt \leq 0. \quad (3.10)$$

Using the fact that $A(\beta^*)$ is Hurwitz, vector $x_0$ can be expressed as the negative of the integral of $\dot{x}^\beta^*$, i.e.,

$$x_0 = - \int_0^\infty \left( A(\beta^*) x^\beta^*(t) + B^T d(t) \right) dt.$$
and, thus, \( v \) can be expressed as \( v = -\int_0^\infty A(\beta^*)x(\beta^*)(t)dt \). Using this expression for \( v \) we find

\[
\int_0^\infty \xi(t)^Tvd(t) = -\int_0^\infty \xi(t)^T \left( \int_0^\infty A(\beta^*)x(\beta^*)(s)ds \right) dt \\
= -\int_0^\infty \xi(t)^TA(\beta^*) \left( \int_0^\infty x(\beta^*)(s)ds \right) dt \\
= \int_0^\infty w^T x(\beta^*)(s)ds,
\]

where, in the last equality, we have used the fact that \( A(\beta^*)^T \) is Hurwitz. Substituting this into (3.10) we find

\[
\int_0^\infty w^T x(\beta^*)(t)dt \leq w^T \mu^*,
\]

which concludes the proof of (ii) and of the theorem.

Note that property (ii) of Theorem 3 implies that the same optimal value is attained in LP problem (3.6) when inequality (3.6b) is replaced by an equality. Although using such an equality would allow us to simplify the formulation by eliminating the vector variable \( \mu \), we prefer to work with LP problem (3.6) as presented in the theorem because this form naturally lends itself to the introduction of additional integral constraints, such as those considered in Problem 1.

Notice that it is straightforward to add constraints of the form \( \beta_{ij} \leq c_{ij} \) for some \( c_{ij} \geq 0 \) by adding inequalities of the form \( z_{ij} \leq c_{ij}\mu_i \) to LP (3.6). Such constraints correspond to limiting the fraction of the outflow of section \( i \) which is routed to section \( j \). This type of constraint can be used to decrease traffic through one or more particular sections of the network.

The results of Theorem 3 can be extended in a straightforward manner to accommodate integral constraints of the form (3.5) and to provide a solution to Problem 1.

**Corollary 1** Let vectors \( w \) and \( \{w_m\}_{m=1}^M \) and scalars \( \{\gamma_m\}_{m=1}^M \) be given, such that \( w > 0 \), \( w_m \geq 0 \) and \( \gamma_m > 0 \) for all \( m \in [M] \). Let \( v = x_0 + \int_0^\infty B^{T}d(t)dt > 0 \). Let \( (\mu^*, z^*) \) be an optimal point of the following
Let routing strategy $\beta^*$ be defined as $\beta^*_{ij} = \frac{z^*_{ij}}{\mu^*}$, $\forall i, j \in [n]$. Then $A(\beta^*)$ satisfies SPC and $\int_0^\infty w^*_m T_x^* (t) dt \leq \gamma_m$ for all $m \in [M]$. In addition,

$$\int_0^\infty w^T x^* (t) dt = w^T \mu^*.$$ 

The proof is similar to the proof of Theorem 3 and is thus omitted for brevity.

### 3.5 Application Example

We applied the routing design techniques of Section 3.4 to a network consisting of four sections. The traversal times for each section are

$$\tau_1 = 0.6250, \quad \tau_2 = 0.9375, \quad \tau_3 = 0.2083, \quad \tau_4 = 0.2500,$$
in units of hours and were chosen to be comparable to the values used in [35]. The connectivity of the network is defined by

\[
O_1 = \{1, 3, 4\}, \quad O_2 = \{2, 3, 4\}, \quad O_3 = \emptyset, \quad O_4 = \emptyset.
\]

The initial conditions used in all simulations are \(x_0 = [200, 200, 50, 50]^T\). This network is depicted graphically in Fig. 3.1 with arrows representing allowable routing from each section to subsequent sections.

Routing parameters were designed and simulations performed for the following three problems.

(a) Minimize total delay with no additional constraints.

(b) Minimize total delay, with the additional constraints that \(\beta_{13} \leq 0.2\) and \(\beta_{23} \leq 0.2\).

(c) Minimize total delay and ensure that \(\int_0^\infty x_3(t)dt \leq 70\).

Note that minimizing total delay corresponds to a cost of \(\int_0^\infty w^T x(t)dt\) where \(w = 1\) (that is, \(w\) is a vector with each element equal to 1). Routing and performance results are presented in Table 3.1.

Several remarks are in order regarding Table 3.1. First, note that \(\int_0^\infty x_i(t)dt\) and the maximum values of \(x_i(t)\) for \(i = 1\) and \(2\), do not vary with the problem formulation. This is due to the fact that the solutions for each of these problems did not include recirculation, thus the dynamics of sections 1 and 2 were the same in all cases.

Appropriate constraints for a given network are application dependent. As discussed earlier, both capacity and delay constraints are applicable to networks of air traffic. Arguments for the use of these constraints for other positive compartmental systems can be made. With the tools presented here, the user can impose capacity and delay constraints as they pertain to specific applications.
Table 3.1: Routing and simulation results.

<table>
<thead>
<tr>
<th>Problem</th>
<th>(a)</th>
<th>(b)</th>
<th>(c)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost</td>
<td>418.75</td>
<td>432.08</td>
<td>423.50</td>
</tr>
<tr>
<td>( \beta_{13} )</td>
<td>1.00</td>
<td>0.20</td>
<td>0.68</td>
</tr>
<tr>
<td>( \beta_{14} )</td>
<td>0.00</td>
<td>0.80</td>
<td>0.32</td>
</tr>
<tr>
<td>( \beta_{23} )</td>
<td>1.00</td>
<td>0.20</td>
<td>0.75</td>
</tr>
<tr>
<td>( \beta_{24} )</td>
<td>0.00</td>
<td>0.80</td>
<td>0.25</td>
</tr>
<tr>
<td>( \int_0^\infty x_1(t)dt )</td>
<td>125.00</td>
<td>125.00</td>
<td>125.00</td>
</tr>
<tr>
<td>( \int_0^\infty x_2(t)dt )</td>
<td>187.50</td>
<td>187.50</td>
<td>187.50</td>
</tr>
<tr>
<td>( \int_0^\infty x_3(t)dt )</td>
<td>93.75</td>
<td>27.08</td>
<td>70.00</td>
</tr>
<tr>
<td>( \int_0^\infty x_4(t)dt )</td>
<td>12.50</td>
<td>92.50</td>
<td>41.00</td>
</tr>
<tr>
<td>max(( x_1(t) ))</td>
<td>200.00</td>
<td>200.00</td>
<td>200.00</td>
</tr>
<tr>
<td>max(( x_2(t) ))</td>
<td>200.00</td>
<td>200.00</td>
<td>200.00</td>
</tr>
<tr>
<td>max(( x_3(t) ))</td>
<td>78.53</td>
<td>50.00</td>
<td>60.76</td>
</tr>
<tr>
<td>max(( x_4(t) ))</td>
<td>50.00</td>
<td>73.87</td>
<td>50.00</td>
</tr>
</tbody>
</table>
Chapter 4

LP Based Routing Design for Systems with Time Varying Capacity Constraints

4.1 Motivation and Problem Description

In this chapter, we focus on the problem of designing time varying routing strategies to satisfy piecewise constant capacity constraints on each section of the network. Such constraints arise naturally in air traffic problems. The capacity constraint can be identified with the weather-dependent sector capacity, which specifies the maximum number of aircraft that a trained human air traffic controller can safely route in given weather conditions. Because weather predictions are often coarse and the definition of sector capacity also involves additional factors which are difficult to predict, a precise profile is typically not available for all times. Instead, sector capacity updates are typically issued at regular intervals, or as needed. This justifies our use of a piecewise constant capacity constraint, as proposed, for example, in [31].

4.2 Chapter Overview

The high level control design objective for all sections in this chapter is that of designing routing parameters to satisfy piecewise constant, time varying capacity constraints over a finite time horizon. With routing used as the control input, care must be taken to ensure that each flight is able to reach its intended destination. Using an aggregate model, the identity and intent of each aircraft involved in the problem is lost, thus it has been suggested, for example, in [32, 33], that flights be aggregated based on destination. The control design problem can then be solved for each specified destination and the solutions implemented simultaneously to address the full multiple destination problem.
In a multiple destination problem, the number of aircraft in each section is represented by the states of several different aggregate models, one for each destination. In this case, care must be taken to ensure that the total traffic (summed over each destination) satisfies the specified capacity constraints.

We use three different section outflow models. We first focus on single destination problems using each of these outflow models. In Section 4.3 we use the fixed traversal time linear outflow model used in Chapter 3. The control design technique developed in Section 4.3 is also presented in [11] and [12]. With a method of designing routing strategies to satisfy capacity constraints for the fixed linear outflow model, we then alter the outflow model and derive control design procedures which parallel that of Section 4.3 in the following two sections. In Section 4.4, we also use a linear outflow model, but allow the traversal times of each section to depend on uncertain parameters which vary over time. In Section 4.5, a nonlinear outflow model is used and a time varying state feedback routing strategy is developed. All of these solution methods can be extended for application to multiple destination problems. As an example, in Section 4.6 we modify the control design technique for the fixed linear outflow model problem of Section 4.3 to incorporate multiple destinations.

The control design techniques for each outflow model are very similar. This technique is presented in Section 4.3 for the simplest outflow model, the fixed traversal time linear outflow model. The mathematical details of the specific dynamic model used is presented in Section 4.3.2. The problem of designing routing parameters to satisfy time varying capacity constraints using this particular dynamic model is formally stated in Section 4.3.3. The control design technique for this model is developed in detail in Section 4.3.4. This section begins with the introduction of constraints on the dynamics of the system which ensure that capacity constraints are satisfied. A detailed description of designing routing parameters over a finite time interval such that the closed loop system satisfies the Positivity and Conservation constraints and linear capacity constraints on each section of the network are satisfied is given in Section 4.3.4.1. To indicate that a system satisfies the Positivity and Conservation constraints, we will say that the system satisfies $PC$ or that the system is $PC$. Finally, in Section 4.3.4.2, a procedure to design time varying routing parameters to satisfy piecewise constant capacity constraints and ensure that the closed loop system is $PC$ is detailed. This procedure involves developing linear constraints on a finite number of control design variables and a method of recovering routing parameters at each time instant which ensure that the capacity constraints are satisfied and that the closed loop system is $PC$. This control design technique is illustrated on a simulated network of air space in Section 4.3.5.

In Section 4.4 a linear time varying outflow model is used with traversal times of each section depending on a set of uncertain, time varying parameters. A nonlinear, saturating outflow model is used in Section
4.5. The mathematical details of each of these models and motivation for the use of these models are given in Sections 4.4.2 and 4.5.2 respectively. The problem of routing design under capacity constraints is formally stated, with constraints and assumptions specific to the respective models, in Sections 4.4.3 and 4.5.3. The control design techniques presented in Sections 4.4.4 and 4.5.4 have a very similar structure to the technique of Section 4.3.4. The major differences between these three techniques are the constraints on the network dynamics used to ensure that the capacity constraints are satisfied. In Sections 4.4.4 and 4.5.4, these constraints are given for the specific dynamic models of interest. The details of the control design development and procedures used to generate routing parameters are presented briefly, focusing mainly on the specific additions or changes needed compared to the method presented for the fixed linear outflow model in Section 4.3.4.

Application examples which illustrate the unique properties of the outflow models used and the control design problems are given in Sections 4.4.5 and 4.5.5.

The problem of routing design to satisfy section capacity constraints for multiple destination networks is addressed in Section 4.6. All of the solution methods presented for the single destination problem can be extended to solve multiple destination problems by aggregating flights based on destination and incorporating additional constraints in the control design procedure to ensure that the total number of aircraft in each section satisfies the desired capacity constraints. As an illustrative example, the fixed linear outflow model and solution method of Section 4.3 are extended for application to the multiple destination problem. The multiple destination aggregate model is presented in Section 4.6.2. The routing problem is formally stated in Section 4.6.3. The control design technique of Section 4.3 is modified for this problem in Section 4.6.4. And finally, an application example is presented in Section 4.6.5.

4.3 Systems with a Fixed Linear Outflow Model

4.3.1 Motivation

In this section, we focus on control design for fixed linear outflow systems. The assumption that the outflow of each section is linear in the state of the section is reasonable when operating in the neighborhood of a steady state where the section traversal times are constant. Physically, this corresponds to low density traffic in which aircraft are free to fly at their nominal speeds.
4.3.2 Continuous Time Aggregate Model

In this section, we use the same linear outflow model with fixed section traversal times as was used in Chapter 3. Recall the dynamics of each section $i$ in the network given in Section 2.2 by equation (2.1)

$$\dot{x}_i(t) = -f_i(x_i(t), t) + \sum_{j: i \in O_j} \beta_{ji}(t) f_j(x_j(t), t) + \sum_{s=1}^S b_{si} d_s(t),$$

with a linear outflow model for each section $i$ of the network with time invariant traversal times $\tau_i$, that is

$$f_i(x_i(t), t) = \frac{x_i(t)}{\tau_i}.$$ 

The dynamics of each section of the network can be written as

$$\dot{x}_i(t) = -\frac{x_i(t)}{\tau_i} + \sum_{j: i \in O_j} \beta_{ji}(t) \frac{x_j(t)}{\tau_j} + \sum_{s=1}^S b_{si} d_s(t), \quad (4.1)$$

for all $i \in [n]$. The dynamics of the $n$ section network is then described by the following linear system

$$\dot{x}(t) = A(\beta(t))x(t) + B^T d(t)$$

$$x(0) = x_0 \quad (4.2)$$

where

$$A(\beta(t)) = A_0 + \sum_{i=1}^n \sum_{j \in O_i} \frac{\beta_{ji}(t)}{\tau_i} e_j e_i^T , \quad (4.3)$$

and $A_0 = \text{diag} \left(-\frac{1}{\tau_1}, \ldots, -\frac{1}{\tau_n} \right)$.

4.3.3 Control Objectives

We now formally present the problem of routing design with section capacity constraints for the fixed linear outflow model. The problem of interest in this section can now formally be written as the following.

**Problem 2** Let a piecewise constant vector-valued function $t \mapsto \bar{c}(t)$ be given such that $\bar{c}(t) > 0$ for all $0 \leq t \leq T$. Find a (possibly time-varying) routing strategy $\beta(t)$ such that constraints PC are satisfied and

$$x^{\beta}(t) \leq \bar{c}(t) \ \forall \ t \in [0, T).$$

35
where $x^\beta(t)$ denotes the solution of (4.2) under routing strategy $\beta(t)$.

### 4.3.4 Control Design

In this section, we give sufficient conditions for the design of time-varying routing strategies satisfying piecewise constant constraints on the state, as specified in Problem 2. We start with a single linear capacity constraint defined over a finite interval, then generalize the results to piecewise constant time-varying capacity constraints using piecewise linear under-approximations.

We start with the following simple but central proposition, which generalizes results from [6] and [10] to time-varying capacity constraints.

**Proposition 1** Let $t \mapsto A(t)$ be a matrix-valued map such that $A(t)$ is Metzler for all $0 \leq t \leq T$ and let $t \mapsto c(t)$ be a differentiable vector-valued map such that $c(t) > 0$ for all $0 \leq t \leq T$. Then, if $x_0 \leq c(0)$ and

$$A(t)c(t) + B^Td(t) \leq \dot{c}(t) \text{ for all } 0 \leq t \leq T,$$

the solution of system

$$\dot{x}(t) = A(t)x(t) + B^Td(t) \quad x(0) = x_0$$

satisfies $x(t) \leq c(t)$ for all $0 \leq t \leq T$.

**Proof 3** Let $\xi = c - x$. We show that, for all $i \in [n],$

$$\begin{bmatrix}
\xi_i(t) = 0 \\
\xi(t) \geq 0
\end{bmatrix} \Rightarrow \dot{\xi}_i(t) \geq 0,$$

which, because $\xi(0) \geq 0$, implies that $\xi(t) \geq 0$ for all $t \geq 0$. By definition of $\xi,$

$$\dot{\xi}(t) = \dot{c}(t) - (A(t)x(t) + B^Td(t))$$

$$\geq A(t)(c(t) - x(t))$$

$$= A(t)\xi(t).$$

Since $A(t)$ is Metzler for all $t$, we know that if $\xi_i(t) = 0$ and $\xi(t) \geq 0$ then $[A(t)\xi(t)]_i \geq 0$. In particular,

$$\dot{\xi}_i(t) \geq [A(t)\xi(t)]_i \geq 0.$$
Thus, $\xi(t) \geq 0$ for all $0 \leq t \leq T$, i.e., $x(t) \leq c(t)$ for all $0 \leq t \leq T$.

4.3.4.1 Linear Capacity Bound

Let us now consider the special case of a constant scalar inflow, i.e.,

$$d(t) = d \geq 0 \text{ for all } 0 \leq t \leq T,$$
$$B \in \mathbb{R}^{1 \times n}.$$

Let us also assume that the capacity constraint varies linearly according to

$$c(t) = b + tm \text{ for all } 0 \leq t \leq T, \quad (4.4)$$

where $b$ and $m$ are constant vectors in $\mathbb{R}^n$.

The following results show how to design routing strategies $\beta(t)$ that satisfy the PC conditions and such that $x^\beta(t) \leq c(t)$ for all $t$.

**Theorem 4** Let the constraint vector $c$ be given as in (4.4) and $x_0 \leq c(0)$. If there exist $\beta(0)$ and $\beta(T)$ such that constraints (2.7) - (2.9) are satisfied and

$$A(\beta(0))c(0) + B^T d \leq m,$$
$$A(\beta(T))c(T) + B^T d \leq m,$$

(4.5)

then the parameters $\beta(t)$ defined by

$$\beta_{ij}(t) = \frac{\left(1 - \frac{1}{T}\right) \beta_{ij}(0)c_i(0) + \frac{t}{T} \beta_{ij}(T)c_i(T)}{\left(1 - \frac{1}{T}\right) c_i(0) + \frac{1}{T} c_i(T)},$$

(4.6)

for all $i,j \in [n]$ and $0 \leq t \leq T$ are such that $A(\beta(t))$ is PC for all $0 \leq t \leq T$. In addition, $x^\beta(t) \leq c(t)$ for all $0 \leq t \leq T$.

**Proof 4** First, note that since each $\beta_{ij}(t)$ is a convex combination of $\beta_{ij}(0)$ and $\beta_{ij}(T)$, it satisfies (2.7) - (2.9) whenever $\beta_{ij}(0)$ and $\beta_{ij}(T)$ do, thus Conservation holds. This also implies that $A(\beta(t))$ is Metzler for all $0 \leq t \leq T$, thus Positivity holds. To simplify notation in the remainder of the proof, let us define $G(t)$ as

$$G(t) = A(\beta(t))c(t) + B^T d,$$
where $\beta(t)$ is given as per (4.6). From (4.6) and the fact that constraint $c(t)$ is linear, we find that

$$
\frac{\beta_{ij}(t)c_i(t)}{\tau_i} e_j = \frac{\beta_{ij}(t) \left[ (1 - \frac{t}{T}) c_i(0) + \frac{t}{T} c_i(T) \right]}{\tau_i} e_j
$$

$$
= \frac{1}{T} \beta_{ij}(0) c_i(0) + \frac{1}{T} \beta_{ij}(T) c_i(T) e_j,
$$

Summing both sides over $i, j \in [n]$ and adding $A_0 c(t) + B^T d$ to both sides results in

$$
G(t) = \left( 1 - \frac{t}{T} \right) G(0) + \frac{t}{T} G(T).
$$

In turn, $G(t) \leq (1 - \frac{t}{T}) m + \frac{t}{T} m = m$, which, according to Proposition 1, implies that $x^\beta(t) \leq c(t)$ for all $0 \leq t \leq T$.

4.3.4.2 Piecewise Constant Capacity Constraints

With Theorem 4 in hand, we are now ready to tackle Problem 2. However, some new notation must be introduced before we can proceed.

Let $T$ be the length of the time interval of interest. As discussed in Section 4.3.3, we assume that the capacity constraint $\bar{c}$ is piecewise constant over intervals of some length. Let $\Delta T$ be the length of these intervals, where $T$ is an integer multiple of $\Delta T$. We would like to build on the method developed in Section 4.3.4.1 to design routing parameters which drive the state to satisfy these constraints. This involves finding a piecewise linear bound $c$ which lies below $\bar{c}$. This bound is piecewise linear over a time interval of length $\Delta t \leq \Delta T$ such that both $T$ and $\Delta T$ are integer multiples of $\Delta t$. The specific value of $\Delta t$ can be selected based on the given application. Decreasing the value of $\Delta t$ leads to greater flexibility in fitting the piecewise linear capacity bound below the piecewise constant capacity constraint.

Define $K = \frac{T}{\Delta t}$, $t_k = k \Delta t$ for $k = 0, \ldots, K$. Using time steps of length $\Delta t$, $T$ can be divided into intervals of the form $i_k = [t_k, t_{k+1})$ with $\bigcup_{k=0}^{K-1} i_k = [0, T)$. Similarly, $L = \frac{T}{\Delta T}$, $T_l = l \Delta T$ for $l = 0, \ldots, L$. Using time steps of length $\Delta T$, $T$ can be divided into intervals of the form $I_l = [T_l, T_{l+1})$ with $\bigcup_{l=0}^{L-1} I_l = [0, T)$.

We will be dealing with discontinuous functions, therefore for any function $g$ we define

$$
g(t_k^+) = \lim_{t \to t_k^+ \atop t > t_k} g(t) \quad \text{and} \quad g(t_k^-) = \lim_{t \to t_k^- \atop t < t_k} g(t).
$$
Recall that our goal is to find, when possible, a time-varying routing strategy $\beta(t)$ such that $A(\beta(t))$ satisfies $PC$ for all $t$ and

$$x^\beta(t) \leq \bar{c}(t) \text{ for all } 0 \leq t \leq T. \quad (4.7)$$

The given constraint $\bar{c}$ is assumed to be constant over intervals $I_l$ for $l = 0, \ldots, L$.

Note that neither Proposition 1 nor Theorem 4 can be used directly in this case, because function $\bar{c}$ is discontinuous from the left at $T_l$ for every $l$. In particular, it is possible that

$$x^\beta(t) \leq \bar{c}(t) \text{ and } A(\beta(t))\bar{c}(t) + B^T d \leq 0 \text{ for all } t \in I_l$$

but that $x^\beta(T_{l+1}) > \bar{c}(T_{l+1})$.

In order to design routing strategies $\beta(t)$ such that constraint (4.7) is satisfied, and guarantee that the inequality is enforced at points of discontinuity of $\bar{c}$, we proceed in two steps.

- First, we introduce a continuous, positive, piecewise linear function $c$ such that

$$c(t) \leq \bar{c}(t) \text{ for all } 0 \leq t \leq T. \quad (4.8)$$

Over each time interval $i_k$ for $k = 0, \ldots, K - 1$, we parametrize this function as

$$c(t) = c(t_k) + (t - t_k)m(t_k^+)$$

for all $t_k \leq t < t_{k+1}$ where $m$ is constant over intervals $i_k$. We define $c$ at the end time of each interval $i_k$ as

$$c(t_{k+1}) = c(t_k) + \Delta t \ m(t_k^+)$$

to ensure continuity. In turn, condition (4.8) and positivity can be equivalently formulated as

$$0 \leq c(t_k) \leq \min\{\bar{c}(t_k^+), \bar{c}(t_k^-)\}.$$

Note that $\bar{c}(t_k^+) = \bar{c}(t_k^-)$ for all $k \in \{0, \ldots, K\}$, unless $t_k = T_l$ for some $l$.

- Second, treating this piecewise linear under-approximation $c$ as a free variable, we apply Theorem 4 over every interval $[t_k, t_{k+1}]$ ($k = 0, \ldots, K - 1$) to design routing strategies such that $x^\beta(t) \leq c(t)$ (and
hence \( x^\beta(t) \leq \bar{c}(t) \) for all \( t_k \leq t \leq t_{k+1} \).

These parameters are computed by

1. Finding a feasible point for the following set of linear constraints, denoted by \( \phi(\bar{c}) \):

\[
\begin{align*}
    c(t_0) & \geq x_0 \\
    c(t_k) & \leq \min\{\bar{c}(t_k^-), \bar{c}(t_k^+)\}, \\
    c(t_k) & \geq 0, \\
    z_{ij}(t_k) & \geq 0, \forall i, j \in [n], \\
    z_{ij}(t_k) & \leq c_i(t_k), \forall i, j \in [n], \\
    z_{ij}(t_k) & = 0, \forall j \in [n] \setminus O_i, \forall i \in [n], \\
    \sum_{j=1}^{n} z_{ij}(t_k) & = c_i(t_k), \forall i \in [n] \setminus S_F, \\
    B^T d(t_k^+) + A_0 c(t_k) + \sum_{i=1}^{n} \sum_{j=1}^{n} z_{ij}(t_k) \tau_{ij} e_j & \leq m(t_k^+), \\
    & \text{for all } k \in \{0, \ldots, K\} \\
    c(t_{k+1}) & = c(t_k) + \Delta t m(t_k^+), \\
    B^T d(t_{k+1}^-) + A_0 c(t_{k+1}) + \sum_{i=1}^{n} \sum_{j=1}^{n} z_{ij}(t_{k+1}) \tau_{ij} e_j & \leq m(t_{k+1}^+), \\
    & \text{for all } k \in \{0, \ldots, K-1\} \\
\end{align*}
\]

2. Recovering \( \beta(t_k) \) according to \( \beta_{ij}(t_k) = \frac{z_{ij}(t_k)}{c_i(t_k)} \) for all \( i, j \in [n] \), \( k = 0, \ldots, K \) and

3. Interpolating non-linearly between \( \beta_{ij}(t_k) \) and \( \beta_{ij}(t_{k+1}) \) over each interval \( i_k \) according to

\[
\beta_{ij}(t) = \frac{(1 - \frac{t-t_k}{\Delta t}) \beta_{ij}(t_k) c_i(t_k) + \frac{t-t_k}{\Delta t} \beta_{ij}(t_{k+1}) c_i(t_{k+1})}{(1 - \frac{t-t_k}{\Delta t}) c_i(t_k) + \frac{t-t_k}{\Delta t} c_i(t_{k+1})}
\]

for all \( t_k \leq t < t_{k+1} \).

Note that the resulting routing strategy is continuous over \([0, T]\) since function \( c \) is continuous by assumption.

It is assumed that \( d \) is constant over intervals \( i_k \) for \( k = 0, \ldots, K \). Also note that the set of variables \( z \) plays the same role as in Theorem 3 in allowing us to linearize constraints (4.5) when both \( \beta \) and \( c \) are free variables.

If linear constraints \( \phi(\bar{c}) \) are infeasible, it is natural to try to alter the desired capacity \( \bar{c} \) so as to find a feasible solution, while ensuring that the resulting bounds are close to \( \bar{c} \) in some sense. In such a case, we
allow $\hat{c}$ to be increased to $\hat{c}$ over intervals of length $\Delta t$ in order to achieve feasible capacity bounds below $\hat{c}$. The problem of finding capacity constraints $\hat{c}$ and capacity bound $c$ can then be written as the following LP problem:

$$
\begin{align*}
\min & \quad \sum_{k=0}^{K-1} \sum_{i=1}^{N} (\hat{c}_i(t_k^+) - \bar{c}_i(t_k^+)) \Delta t \\
\text{subject to} & \quad \phi(\hat{c}) \\
& \quad \hat{c}(t_k^+) \geq \bar{c}(t_k^+), \quad k \in \{0, \ldots, K\}.
\end{align*}
$$

(4.9)

### 4.3.5 Application Example

We applied the routing design strategy of Section 4.3.4.2 to the compartmental system depicted in Figure 4.1. We chose a traversal time $\tau_i = 0.4$ hours for every $i = 1, \ldots, 21$, to agree with typical orders of magnitude encountered in the air traffic management literature [35]. The connectivity of the network, i.e. the definition of $O_i$ for all $i \in [n]$, can be inferred from the diagram, for example $O_1 = \{1, 4, 5\}$, $O_2 = \{2, 4, 5, 6\}$, and the set of final sections is $S_F = \{19, 20, 21\}$.

The inflow rate of sections 1 and 3 is 25 aircraft per hour, while the inflow rate of section 2 is 30 aircraft per hour. Initial conditions were set to 10 aircraft for all sections in the top and bottom rows and 12 for all sections in the middle row. With this inflow and initial conditions, the state of every section remains constant when flows are routed along the rows of the network (i.e., when $\beta_{1,4} = \beta_{4,7} = \ldots = \beta_{16,19} = 1$ and similar equalities hold for the second and third rows).

Each section except 14 has a constant capacity of 15, i.e., $\bar{c}_i(t) = 15$ aircraft for all $i \in [n]$ such that $i \neq 14$ and all $t \geq 0$. Section 14, on the other hand, has the piecewise constant capacity profile pictured in Figure 4.2, where each base interval has length $\Delta T = 30$ minutes.

Based on this profile, we formulated linear constraints $\phi(\bar{c})$ using a base interval for the piecewise linear capacity bound $c$ of $\Delta t = 15$ minutes. Constraints $\phi(\bar{c})$ were found to be feasible, which, in turn, implies that Problem 2 has a solution. The corresponding routing parameters for select sections are plotted in Figure 4.1.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{network_diagram.png}
\caption{Network of interconnected sections. Aircraft enter the network at sections 1, 2 and 3, aircraft exit the network from sections 19, 20, 21.}
\end{figure}
Figure 4.2: Capacity constraint $\bar{c}_{14}$, capacity bound $c_{14}$ and state $x_{14}$ of section 14.

4.3. Notice that in sections 1, 2 and 3, well upstream of the capacity constrained section, the majority of the section outflow is routed to the upper and lower sections of the graph. Closer to the capacity constrained section, in sections 10, 11 and 12, a larger portion of the section outflow is routed to the upper and lower sections of the graph.

In a second example, we imposed the capacity profile pictured in Figure 4.4 on section 14. The capacity was decreased from 12 to 2 over the first half hour of the simulation. We found that constraints $\phi(\bar{c})$ were not feasible, i.e., that it is not possible to enforce such a steep decrease in the state of section 14 using the proposed method. Thus, the capacity constraints must be increased to $\hat{c}$ in order to find a feasible solution. The integral of the difference between $\hat{c}$ and $\bar{c}$, calculated as the cost of LP problem (4.43), is 0.84 aircraft × hour. That is, using this solution, the actual section count will be above the constraint $\bar{c}$ by no more than an average of 0.84 aircraft over a one hour time period.

Notice that in both Figures 4.2 and 4.4 the state $x_{14}(t)$ does not match the capacity bound $c_{14}(t)$ for most of the simulation. This is because the use of Proposition 1 to ensure that $x(t) \leq c(t)$ implicitly assumes that the state is at capacity at all times, which is conservative, resulting in the gap between $x_{14}(t)$ and $c_{14}(t)$. 
Figure 4.3: Routing parameters associated with results plotted in Figure 4.2. Note that, due to symmetry of the problem, several of the routing parameters have identical profiles. Also recall that $\beta_{ij}(t)$ must sum to 1 for each section, and thus recirculation accounts for the remainder of the flow routing (i.e. $\beta_{1,1}(t) = 1 - \beta_{1,4}(t) - \beta_{1,5}(t)$, etc.).

Figure 4.4: Capacity constraint $\hat{c}_{14}$, adjusted capacity constraint $\tilde{c}_{14}$, capacity bound $c_{14}$ and state $x_{14}$ of section 14. Note that $\tilde{c}_{14}$ is only plotted over intervals in which it differs from $\hat{c}_{14}$. 
4.4 Systems with an Uncertain Linear Time Varying Outflow Model

4.4.1 Motivation

In this section, we keep the general form of the section outflow rate that was used in Section 4.3, however, we allow the traversal time of each section to depend on time-varying uncertain parameters. These uncertain parameters can be used to capture a number of effects in air traffic. For example, weather conditions may increase the average traversal times of flights traveling through a region of airspace affected by weather. Flights may be subject to air traffic control actions which reduce their flight speed or increase the length of the path traveled through the section, to either avoid bad weather or to increase spacing between flights. All of these air traffic control actions act to reduce the flow rate of flights traveling through the section and effectively increase the average traversal time, \( \tau_i \).

Due to uncertainty in weather predictions, the resulting changes in the average flow rates through affected regions of airspace is uncertain. The model used in this section allows for a linear dependence of the inverse of the average traversal time on these parameters. These parameters are uncertain, but are given some bounds and represent the set of possible weather scenarios that may occur given an uncertain forecast.

4.4.2 Continuous Time Aggregate Model

The basic linear outflow model of Chapter 3 and Section 4.3 is used here, however, the traversal times of each section are allowed to vary over time, thus the outflow rate of section \( i \) can be written as

\[
f_i(x_i(t), t) = \frac{x_i(t)}{\tau_i(t)}
\]

for all \( i \in [n] \). The dynamics of each section of the network can be written as the following linear parameter varying system

\[
\dot{x}_i(t) = -\frac{x_i(t)}{\tau_i(t)} + \sum_{j : i \in \mathcal{O}_j} \beta_{ji}(t) \frac{x_j(t)}{\tau_j(t)} + \sum_{s=1}^{S} b_{si} d_s(t),
\]

for all \( i \in [n] \). The dynamics of the \( n \) section network is then described by the system

\[
\dot{x}(t) = A(\beta(t), t)x(t) + B^T d(t)
\]

\[
x(0) = x_0
\]  

(4.10)
where

\[ A(\beta(t), t) = A_0(t) + \sum_{i=1}^{n} \sum_{j \in O_i} \frac{\beta_{ij}(t)}{\tau_i(t)} c_{je_i}^{T}, \]

and \( A_0(t) = \text{diag}\left(-\frac{1}{\tau_1(t)}, \ldots, -\frac{1}{\tau_n(t)}\right). \)

Here we consider uncertain, time varying, piecewise constant traversal times, \( \tau_i(t) \) for all \( i \in [n] \). We allow the traversal times of each section to depend on a set of uncertain parameters, \( \delta(t) = (\delta_1(t), \delta_2(t), \ldots, \delta_p(t))^{T} \in \mathbb{R}^p \). While \( \delta(t) \) is unknown, it is assumed that bounds for each element of \( \delta(t) \) are known, that is

\[ \underline{\delta}_w \leq \delta_w(t) \leq \bar{\delta}_w \]

for all \( w \in [p] \), and that \( \delta(t) \) is piecewise constant over each interval \( i_k \) for \( k = 0, \ldots, K - 1 \). We define

\[ \Delta = \left\{ \delta \mid \underline{\delta}_w \leq \delta_w \leq \bar{\delta}_w, \ \forall \ w \in [p] \right\}. \]

For each section \( i \) we define \( \frac{1}{\tau_i(t)} \) as a function of \( \delta(t) \) by

\[ \frac{1}{\tau_i(t)} = \gamma_0^i + \sum_{w=1}^{p} \gamma_w^i \delta_w(t), \] (4.11)

where \( \gamma_0^i \) and \( \gamma_w^i \) for all \( w \in [p] \) are fixed constant coefficients. Note that each of \( \underline{\delta}_w \) and \( \bar{\delta}_w \) for \( w \in [p] \) and the coefficients \( \gamma_0^i \) and \( \gamma_w^i \) for all \( i \in [n] \) and \( w \in [p] \) may take on any real value, however the values of these parameters must be chosen to ensure that \( \frac{1}{\tau_i(t)} \) is positive for all \( \delta(t) \in \Delta \), since \( \tau_i(t) \) represents the traversal time of section \( i \) and must be a nonnegative value.

4.4.3 Control Objectives

We now formally present the problem of routing design with section capacity constraints for the linear model with piecewise constant, uncertain traversal times.

**Problem 3** Let \( \Delta \), fixed constant scalar values \( \gamma_0^i \) and \( \gamma_w^i \) for all \( w \in [p] \) and \( i \in [n] \), and a piecewise constant vector-valued function \( t \mapsto \bar{c}(t) \) such that \( \bar{c}(t) > 0 \) for all \( 0 \leq t \leq T \) be given. Find a (possibly time-varying) routing strategy \( \beta(t) \) that may depend on piecewise constant \( \delta(t) \in \Delta \), which is revealed in real time, such that constraints PC are satisfied and

\[ x^{\beta}(t) \leq \bar{c}(t) \ \forall \ t \in [0, T). \]
where \(x^\beta(t)\) denotes the solution of (4.10) under routing strategy \(\beta(t)\) with \(\tau_i(t)\) defined by (4.11).

### 4.4.4 Control Design

In order to design an adaptive routing strategy which depends on the realized values of \(\delta(t)\), we allow the routing parameters to depend linearly on some user defined subset of the values of delta \(\delta(t)\). That is, for each section \(i\) there is an associated user defined set \(D_i \subseteq [p]\) such that for each \(j \in O_i\), \(\beta_{ij}(t)\) may depend on \(\delta_w(t)\) for all \(w \in D_i\). In practice, this set \(D_i\) could include the uncertain parameters that effect the immediate neighbors of section \(i\). Allowing \(\beta(t)\) to depend on some subset of the parameters \(\delta(t)\), we can develop a procedure for generating routing parameters before the actual values of the uncertain parameters are revealed. During real time application, a simple calculation can be done given the actual values \(\delta(t)\) to generate the specific values of \(\beta(t)\) to be implemented. We allow \(\beta_{ij}(t)\) to be discontinuous since \(\tau_i(t)\) is discontinuous.

Given the definition of time varying traversal times for each section which depend on uncertain parameters, we would like to derive a control design procedure similar to that presented in Section 4.3.4. We begin by constructing constraints similar to those given in \(\phi(\bar{c})\). With discontinuous routing parameters \(\beta(t)\) which depend linearly on the uncertain parameters \(\delta(t)\), we redefine the parameters \(z\) as follows

\[
z_{ij}(t^+_k) = \beta_{ij}(t^+_k)c_i(t_k) \\
= z^0_{ij}(t^+_k) + \sum_{w \in D_i} \delta_w(t^+_k)z^w_{ij}(t^+_k)
\]

\[
z_{ij}(t^-_k) = \beta_{ij}(t^-_k)c_i(t_k) \\
= z^0_{ij}(t^-_k) + \sum_{w \in D_i} \delta_w(t^-_k)z^w_{ij}(t^-_k).
\]

The \(i^{th}\) row of \(A(\beta(t^+_k), t^+_k)\) becomes

\[
-\left(\gamma^i_{j0} + \sum_{w=1}^p \gamma^i_w \delta_w(t^+_k)\right) c_i(t_k) + \sum_{j : i \in O_j} \left(\gamma^i_{j0} + \sum_{w=1}^p \gamma^j_w \delta_w(t^+_k)\right) \left(z^0_{ji}(t^-_k) + \sum_{w \in D_j} \delta_w(t^-_k)z^w_{ji}(t^-_k)\right) \leq m_i(t^+_k) \quad (4.12)
\]

Making a direct translation of the constraints of \(\phi(\bar{c})\) using these new definitions of \(\tau\) and \(z\), we define
\( \phi_\delta(c, \Delta) : \)

\[
c(t_0) \geq x_0
\]

for all \( k \in \{0, \ldots, K\} \)

\[
c(t_k) \leq \min\{\bar{c}(t^-_k), \bar{c}(t^+_k)\},
\]

\[
c(t_k) \geq 0,
\]

for all \( k \in \{0, \ldots, K\} \), \( \delta \in \Delta \)

\[
z^0_0(t^+_k) + \sum_{w \in D_i} \delta_w z^0_{ij}(t^+_k) \geq 0, \quad \forall \, i, j \in [n],
\]

\[
z^0_0(t^+_k) + \sum_{w \in D_i} \delta_w z^0_{ij}(t^+_k) \leq c_i(t_k), \quad \forall \, i, j \in [n],
\]

\[
z^0_0(t^+_k) + \sum_{w \in D_i} \delta_w z^0_{ij}(t^+_k) = 0, \quad \forall \, j \in [n] \setminus O_i, \forall \, i \in [n],
\]

\[
\sum_{j=1}^{n} z^0_0(t^+_k) + \sum_{w \in D_i} \delta_w z^0_{ij}(t^+_k) = c_i(t_k), \quad \forall \, i \in [n] \setminus \mathcal{F},
\]

\[
z^0_0(t^-_k) + \sum_{w \in D_i} \delta_w z^0_{ij}(t^-_k) \geq 0, \quad \forall \, i, j \in [n],
\]

\[
z^0_0(t^-_k) + \sum_{w \in D_i} \delta_w z^0_{ij}(t^-_k) \leq c_i(t_k), \quad \forall \, i, j \in [n],
\]

\[
z^0_0(t^-_k) + \sum_{w \in D_i} \delta_w z^0_{ij}(t^-_k) = 0, \quad \forall \, j \in [n] \setminus O_i, \forall \, i \in [n],
\]

\[
\sum_{j=1}^{n} z^0_0(t^-_k) + \sum_{w \in D_i} \delta_w z^0_{ij}(t^-_k) = c_i(t_k), \quad \forall \, i \in [n] \setminus \mathcal{F},
\]

for all \( k \in \{0, \ldots, K - 1\} \)

\[
c(t_{k+1}) = c(t_k) + \Delta t m(t^+_k),
\]

for all \( k \in \{0, \ldots, K\} \), \( \delta \in \Delta, i \in [n] \)

\[
- \left( \gamma^i_0 + \sum_{w=1}^{p} \gamma^i_w \delta_w \right) c_i(t_k) + \sum_{j \in O_i} \left\{ \left( \gamma^j_0 + \sum_{w=1}^{p} \gamma^j_w \delta_w \right) \left( z^0_{ji}(t^+_k) + \sum_{w \in D_j} \delta_w z^w_{ji}(t^+_k) \right) \right\}
\]

\[
\quad + (B^T d(t^+_k)) \leq m_i(t^+_k),
\]

for all \( k \in \{0, \ldots, K - 1\} \), \( \delta \in \Delta, i \in [n] \)

\[
- \left( \gamma^i_0 + \sum_{w=1}^{p} \gamma^i_w \delta_w \right) c_i(t_k) + \sum_{j \in O_i} \left\{ \left( \gamma^j_0 + \sum_{w=1}^{p} \gamma^j_w \delta_w \right) \left( z^0_{ji}(t^-_{k+1}) + \sum_{w \in D_j} \delta_w z^w_{ji}(t^-_{k+1}) \right) \right\}
\]

\[
\quad + (B^T d(t^-_{k+1})) \leq m_i(t^+_k).
\]
Assuming that a feasible point of $\phi_\delta(\bar{c}, \Delta)$ can be found, $\beta_{ij}(t)$ can be recovered over each interval $i_k$ for $k = \{0, ..., K - 1\}$ such that the resulting system is positive, conservative and $x^\delta(t) \leq \bar{c}(t)$ for all $0 \leq t \leq T$. As $\delta(t)$ is revealed in real time, the values of $\beta_{ij}(t)$ can be calculated in a two step process. First, at the beginning of each interval $i_k$, recover $\beta_{ij}(t)$ at the end points of the interval according to

$$\beta_{ij}(t^+_k) = z^0_{ij}(t^+_k) + \sum_{w \in D_i} \delta_w(t^+_k) z^w_{ij}(t^+_k),$$

$$\beta_{ij}(t^-_{k+1}) = z^0_{ij}(t^-_{k+1}) + \sum_{w \in D_i} \delta_w(t^-_{k+1}) z^w_{ij}(t^-_{k+1})$$

for all $\delta(t)$ is assumed to be constant between $t^+_k$ and $t^-_{k+1}$. Thus $\delta(t^+_k)$ and $\delta(t^-_{k+1})$ are available at the beginning of interval $i_k$ in order to calculate (4.13) and (4.14). Second, interpolate non-linearly between $\beta_{ij}(t^+_k)$ and $\beta_{ij}(t^-_{k+1})$ by

$$\beta_{ij}(t) = \frac{(1 - \frac{t-t_k}{\Delta t}) \beta_{ij}(t^+_k) \delta(t_k) + \frac{t-t_k}{\Delta t} \beta_{ij}(t^-_{k+1}) \delta(t_{k+1})}{(1 - \frac{t-t_k}{\Delta t}) \delta(t_k) + \frac{t-t_k}{\Delta t} \delta(t_{k+1})}$$

for all $t_k \leq t < t_{k+1}$. Given the constraints on $z$ in $\phi_\delta(\bar{c}, \Delta)$ which hold for all $\delta \in \Delta$, the values of $\beta_{ij}(t)$ at the end points of the intervals $i_k$ for $k = \{0, ..., K\}$ given by (4.13) satisfy (2.7) through (2.9). Additionally, the final two constraints of $\phi_\delta(\bar{c}, \Delta)$ imply that inequalities (4.5) hold at the end points of each interval $i_k$ for $k = \{0, ..., K\}$ and for all $\delta \in \Delta$. Thus Theorem 4 can be applied over each interval to conclude that under this routing strategy, system (4.10) is positive, conservative and $x^\delta(t) \leq \bar{c}(t)$ for all $0 \leq t \leq T$. And finally, the third constraint of $\phi_\delta(\bar{c}, \Delta)$ ensures that $c(t) \leq \bar{c}(t)$ for $0 \leq t \leq T$, and so we can conclude that $x^\delta(t) \leq \bar{c}(t)$ for all $0 \leq t \leq T$.

With these definitions of $\tau$ and $z$ and their dependencies on the uncertain parameter $\delta$, the constraints given in $\phi_\delta(\bar{c}, \Delta)$ are polynomial in the uncertain parameter $\delta$. These constraints can be translated into linear constraints through the application of Handelman’s Theorem [25]. The formulation of this theorem which is presented in [18] is included below for completeness.

**Theorem 5 (Handelman’s Theorem)** Let $S$ be a compact polytope defined by linear inequalities $g_i(\cdot) \geq 0$, that is $S = \{ x \in \mathbb{R}^N \mid g_i(x) \geq 0, \forall i \}$ Then, every polynomial $P$ that is positive over $S$ can be expressed as a linear combination with nonnegative (nonpositive) coefficients of products of member of $\{g_i(\cdot)\}$.

A simple example, also given in [18], will illustrate how Theorem 5 can be applied for the current application.

**Example 1** Assume we are given an uncertain scalar parameter $y$ such that $y$ is constrained to take on
values between $-1$ and $1$, that is $y \in S$ where $S = \{y : -1 \leq y \leq 1\}$. Suppose we would like to characterize the polynomials of the form

$$p(y) = \alpha_2 y^2 + \alpha_1 y + \alpha_0$$

such that $p(y) \geq 0$ for all $-1 \leq y \leq 1$. The polytope $S$ is defined by

$$g_1(y) = y + 1 \geq 0$$
$$g_2(y) = 1 - y \geq 0.$$ 

The polynomial $p(y) \geq 0$ if it can be written as

$$p(y) = \tau_1 g_1(y) + \tau_2 g_2(y) + \tau_3 g_1(y) g_2(y) + \tau_4 g_1(y)^2 + \tau_5 g_2(y)^2$$

$$= (\tau_4 + \tau_5 - \tau_3) y^2 + (\tau_1 - \tau_2 + 2\tau_4 - 2\tau_5) y + (\tau_1 + \tau_2 + \tau_3 + \tau_4 + \tau_5)$$

for some $\tau_i \geq 0$, $i = 1, \ldots, 5$. Thus, finding $\alpha_i, i = 0, 1, 2$ such that $p(y) \geq 0$ can be achieved by matching the coefficients of the $y^0, y$ and $y^2$ in these two expressions for $p(y)$, respectively. That is, the problem of finding $\alpha_0, \alpha_1$ and $\alpha_2$ such that $p(y) \geq 0$ for all $-1 \leq y \leq 1$ can be transformed into the problem of finding $\alpha_0, \alpha_1, \alpha_2$ and $\tau_1, \tau_2, \tau_3, \tau_4, \tau_5$ such that

$$\alpha_2 = \tau_4 + \tau_5 - \tau_3,$$
$$\alpha_1 = \tau_1 - \tau_2 + 2\tau_4 - 2\tau_5,$$
$$\alpha_0 = \tau_1 + \tau_2 + \tau_3 + \tau_4 + \tau_5,$$
$$\tau_i \geq 0, \forall i = 1, 2, \ldots, 5.$$ 

As in Example 1, we will use Theorem 5 to transform the inequality constraints in $\phi_3(\bar{c}, \Delta)$ involving the uncertain parameter $\delta$ into the problem of matching the coefficients between two polynomials.

Here, the inequalities defining polytope $\Delta$ are

$$g_w(\delta) = \delta_w - \bar{\delta}_w \geq 0, \forall w \in [p],$$
$$g_{p+w}(\delta) = -\delta_w + \bar{\delta}_w \geq 0, \forall w \in [p].$$

Each of the inequalities of $\phi_3(\bar{c}, \Delta)$ can be rearranged into the form $p(\delta) \geq 0$ where $p(\cdot)$ is a polynomial of degree two. According to Theorem 5, $p(\delta) \geq 0$ is satisfied for all $\delta \in \Delta$ if and only if there exists a set of
parameters $\psi \in \mathbb{R}^2_+$ and $\Psi \in \mathbb{R}^{2p \times 2p}_+$ such that

$$p(\delta) = \sum_{w=1}^{2p} \psi_w g_w(\delta) + \sum_{r=1}^{2p} \sum_{s=1}^{2p} \Psi_{rs} g_r(\delta) g_s(\delta).$$

Given $g_w(\delta)$ for all $w \in [2p]$, and assuming, without loss of generality, that $\Psi$ is symmetric, the right hand side of (4.16) can be written as

\[
\psi^T \begin{bmatrix}
-\delta & \delta \\
-\delta & \delta
\end{bmatrix} + \begin{bmatrix}
-\delta & \delta \\
-\delta & \delta
\end{bmatrix}^T \Psi \begin{bmatrix}
-\delta & \delta \\
-\delta & \delta
\end{bmatrix} = \psi^T \begin{bmatrix}
0 & 0 \\
0 & 0
\end{bmatrix} + \begin{bmatrix}
0 & 0 \\
0 & 0
\end{bmatrix}^T \Psi \begin{bmatrix}
-\delta & \delta \\
-\delta & \delta
\end{bmatrix}. \tag{4.17a}
\]

The problem then becomes one of matching the coefficients of the constant, linear and quadratic terms of (4.17b) with those in the particular polynomial constraint of interest $p(\delta)$. Let $E_r$ be the $r^{th}$ canonical basis vector of $\mathbb{R}^p$. The constant term of (4.17b) is

\[
\psi^T \begin{bmatrix}
0 & 0 \\
0 & 0
\end{bmatrix} = \psi^T \begin{bmatrix}
0 & 0 \\
0 & 0
\end{bmatrix}.
\]

For each $r \in [p]$, the coefficient of $\delta_r$ is

\[
\begin{bmatrix}
E_r & -E_r
\end{bmatrix} \psi^T \begin{bmatrix}
-\delta_r \\
\delta_r
\end{bmatrix}.
\]

For each $r, s \in [p]$, the coefficient of $\delta_r \delta_s$ is

\[
2 \begin{bmatrix}
E_r \\
-E_r
\end{bmatrix}^T \psi \begin{bmatrix}
E_s \\
-E_s
\end{bmatrix}.
\]

Most of the constraints in $\phi_\delta(\bar{c}, \Delta)$ are order zero or one in $\delta$ and it is straightforward to pick out the coefficients of the components of $\delta$. The final two constraints are order two in $\delta$. Focusing on the second to
last constraint of $\phi_\delta(\bar{c}, \Delta)$ the constant term is

$$m_i(t_k^+ - (B^T d(t_k^+))_i) + \gamma_0^j c_i(t_k) - \sum_{j: i \in O_j} \gamma_0^j z_{ji}^0 (t_k^+),$$

the coefficient of $\delta_w$ is

$$c_i(t_k^+) \gamma_w^j - \sum_{j: i \in O_j, w \in D_j} \gamma_0^j z_{wj}^0 (t_k^+) - \sum_{j: i \in O_j} z_{ji}^0 (t_k^+) \gamma_w^j,$$

and finally the coefficient of $\delta_r \delta_s$ is

$$- \sum_{j: i \in O_j, s \in D_j} \gamma_r^j z_{js}^0 (t_k^+) - \sum_{j: i \in O_j, r \in D_j} \gamma_s^j z_{ri}^0 (t_k^+).$$

Thus, each constraint of $\phi_\delta(\bar{c}, \Delta)$ which is polynomial in $\delta$ can be translated into a coefficient matching problem, resulting in linear constraints. A different set of parameters, $\psi$ and $\Psi$ must be used for each constraint at each time interval.

Using Theorem 5 to transform constraints in $\phi_\delta(\bar{c}, \Delta)$ which are polynomial in $\delta$ and recovering $\beta(t)$ as per (4.15) the routing problem of interest can be written as the following LP problem.

$$\begin{align*}
\text{min} & \quad \sum_{k=0}^{K-1} \sum_{i=1}^{N} (\bar{c}_i(t_k^+) - \hat{c}_i(t_k^+)) \Delta t \\
\text{subject to} & \quad \phi_\delta(\hat{c}, \Delta) \\
& \quad \hat{c}(t_k^+) \geq \bar{c}(t_k^+), \; k \in \{0, \ldots, K}\end{align*}$$

As in LP problem (4.43) given in Section 4.3.4.2, we introduce $\hat{c}$ in order to adjust constraints $\bar{c}$ when they are found to be infeasible.
4.4.5 Application Example

The control design technique for networks with uncertain parameters developed in Section 4.4.4 is applied to the simple network depicted in Figure 4.5. Two uncertain parameters are used to describe the traversal times of each section. The upper and lower bounds of these parameters are

\[
1.25 \text{ hr}^{-1} \leq \delta_1(t) \leq 2.00 \text{ hr}^{-1} \\
-0.25 \text{ hr}^{-1} \leq \delta_2(t) \leq 0.25 \text{ hr}^{-1}.
\]

The dependency of the inverse of the traversal times of each section are defined by

\[
\frac{1}{\tau_1(t)} = \delta_1(t), \\
\frac{1}{\tau_2(t)} = 2.25 \text{ hr}^{-1} + \delta_2(t), \\
\frac{1}{\tau_3(t)} = 2.25 \text{ hr}^{-1} - \delta_2(t).
\]

The corresponding ranges of the traversal times are

\[
0.5 \text{ hr} \leq \tau_1(t) \leq 0.8 \text{ hr}, \\
0.4 \text{ hr} \leq \tau_2(t) \leq 0.5 \text{ hr}, \\
0.4 \text{ hr} \leq \tau_3(t) \leq 0.5 \text{ hr}.
\]

Each section in the network initially contains 10 aircraft. The inflow rate into section 1 is 20 aircraft per hour for the duration of the simulation. The capacity constraint for section 1 is set to a constant value of 15 aircraft. The capacity constraints for both section 2 and section 3 are time varying and shown graphically in Figure 4.6.

Since both sections 2 and 3 are final sections of the network (leading to the sink) the problem is to find routing parameters out of section 1 to sections 2 and 3 to satisfy the specified capacity constraints, or adjust these constraints when they are found to be infeasible. In order to study the effect of allowing $\beta(t)$ to depend
on various combinations of the uncertain parameters $\delta$, we let $D_1$ take on the following values

\[
D_1 = \emptyset, \\
D_1 = \{1\}, \\
D_1 = \{2\}, \\
D_1 = \{1, 2\}.
\]

For each of these values of $D_1$, LP program (4.18) is solved to find design parameters $c$ and $z$. Then, given the realized value of $\delta(t)$, routing parameters are found according to (4.15). For the purposes of this example, the realized value of $\delta(t)$ was chosen such that

\[
\tau_1 = 0.5 \text{ hr}, \\
\tau_2 = 0.4 \text{ hr}, \\
\tau_3 = 0.5 \text{ hr}.
\]

Section 2 is faster than section 3 (i.e. it has a shorter traversal time and aircraft move through that section faster than section 3). Given that sections 2 and 3 have the same capacity constraints, it is expected that a solution which routes more traffic through section 2 than section 3 will be a lower cost solution.

Results are shown for each of the values of $D_1$ in Figures 4.7, 4.8, 4.9, 4.10.
Simulation results for $D_1 = \emptyset$ are shown in Figure 4.7. Here $\beta$ is not allowed to depend on $\delta$, that is, the same routing strategy is implemented regardless of the realized value of $\delta(t)$. Since the constraints on $\delta(t)$ result in the same range of values for $\tau_2$ and $\tau_3$, the solution is symmetric and routes the same fraction of traffic from section 1 to section 2 as it does from section 1 to section 3. The problem of routing to satisfy constraints $\bar{c}$ are found to be infeasible and are thus adjusted to $\hat{c}$. The cost of this solution is 8.2 aircraft × hour.

We then allow $\beta$ to depend on $\delta_1$, with results plotted in Figure 4.8. Although this provides no information on the realized traversal times of sections 1 and 2, it does affect the routing parameters of section 1. Given the realized value of $\delta_1$, the traversal time of section 1 is at its lower bound. The routing solution is symmetric (that is $\beta_{1,2}(t) = \beta_{1,3}(t)$) however, these values are lower than those for the solution found for $D_1 = \emptyset$, with
Figure 4.8: Simulation results for the routing solution obtained from LP problem (4.18) with $D_1 = \{1\}$, and realized traversal times of $\tau = [0.5, 0.4, 0.5]$. Recall that $\beta_1(t)$ must sum to 1 for section 1, and thus recirculation accounts for the remainder of the flow routing (i.e. $\beta_{1,1}(t) = 1 - \beta_{1,2}(t) - \beta_{1,3}(t)$).

more flow recirculated back into section 1. This effectively slows the outflow of section 1 in order to decrease capacity constraint violations downstream. The cost of this solution is 2.9 aircraft × hour.

Next we allow $\beta$ to depend on $\delta_2$ which is used to generate the realized values of $\tau_2$ and $\tau_3$. Here we see an asymmetric routing solution, shown in Figure 4.9. The fact that section 2 is the faster section is now incorporated in the routing solution since $\beta$ is allowed to depend on $\delta_2$. A greater portion of the outflow of section 1 is routed to section 2 compared to the portion routed to section 3. The cost of this solution is 5.8 aircraft × hour.

Finally, we allow $\beta$ to depend on both $\delta_1$ and $\delta_2$ with results shown in Figure 4.10. This solution combines the cost reducing benefits of the scenarios in which $\beta$ is allowed to depend only on $\delta_1$ and $\delta_2$, respectively. Here we see an asymmetric routing solution similar to that shown in 4.9 with greater recirculation into
Figure 4.9: Simulation results for the routing solution obtained from LP problem (4.18) with $D_1 = \{2\}$, and realized traversal times of $\tau = [0.5, 0.4, 0.5]$. Recall that $\beta_{1,j}(t)$ must sum to 1 for section 1, and thus recirculation accounts for the remainder of the flow routing (i.e. $\beta_{1,1}(t) = 1 - \beta_{1,2}(t) - \beta_{1,3}(t)$).
Recall that $\beta_{1,j}(t)$ must sum to 1 for section 1, and thus recirculation accounts for the remainder of the flow routing (i.e. $\beta_{1,1}(t) = 1 - \beta_{1,2}(t) - \beta_{1,3}(t)$).

The cost of this solution is lower than the previous three solutions at 2.4 aircraft $\times$ hour.

### 4.5 Systems with a Nonlinear Outflow Model

#### 4.5.1 Motivation

In this section, we again address the problem of designing a routing strategy to satisfy piecewise constant capacity constraints. The modification to the problem formulation introduced in this section is that it is assumed that the outflow rate of each section is a nonlinear function of the state of the section. As discussed in [33], although the outflow of a section of airspace will increase as the density of traffic increases, there is a limit on the maximum outflow rate due to minimum separation requirements between flights. Thus, we
assume that the outflow rate functions of each section are increasing, concave, saturating functions.

### 4.5.2 Continuous Time Aggregate Model

In this section, the outflow rate of each section is a nonlinear function of the state of that section, that is

\[ f_i(x_i(t), t) = \mu_i(x_i(t)). \]

With this outflow rate, the section dynamics given in equation (2.1) becomes

\[
\dot{x}_i(t) = -\mu_i(x_i(t)) + \sum_{j : i \in O_j} \beta_{ji}(t) \mu_j(x_j(t)) + \sum_{s=1}^{S} b_{si} d_s(t). \tag{4.19}
\]

We restrict ourselves to outflow functions \( \mu_i : \mathbb{R}_+ \to \mathbb{R}_+ \) which satisfy the following assumptions

1. \( \mu_i(0) = 0 \),
2. \( \mu_i \) is monotonically increasing,
3. \( \mu_i \) is concave,

for all \( i \in [n] \). The dynamics (4.19) of all \( n \) sections can be summarized by the system

\[
\dot{\tilde{x}}(t) = F\left(\tilde{x}(t), \tilde{\beta}(t)\right) + B^T d(t) \\
\tilde{x}(0) = \tilde{x}_0. \tag{4.20}
\]

We introduced the state variable \( \tilde{x} \) because we will make use of the network model which uses fixed linear outflow rates for each section in the derivation of the routing design strategy in this section. The fixed linear outflow model is recalled here. The dynamics of each section of the network can be written as

\[
\dot{x}_i(t) = -\frac{x_i(t)}{\tau_i} + \sum_{j : i \in O_j} \beta_{ji}(t) \frac{x_j(t)}{\tau_j} + \sum_{s=1}^{S} b_{si} d_s(t), \tag{4.21}
\]

for all \( i \in [n] \). The dynamics of the \( n \) section network is then described by the following linear time invariant system

\[
\dot{x}(t) = A(\beta(t)) x(t) + B^T d(t) \\
x(0) = x_0. \tag{4.22}
\]
where
\[
A(\beta(t)) = A_0 + \sum_{i=1}^{n} \sum_{j \in O_i} \frac{\beta_{ij}(t)}{\tau_i} e_j e_i^T,
\]
and \(A_0 = \text{diag}\left(-\frac{1}{\tau_1}, \ldots, -\frac{1}{\tau_n}\right)\).

### 4.5.3 Control Objectives

We now formally present the problem of routing design with section capacity constraints for the model with nonlinear, saturating outflow rates. The problem of interest in this section is the same as Problem 2 used in Section 4.3, and is reproduced here for completeness.

**Problem 4** Let a piecewise constant vector-valued function \(t \mapsto \bar{c}(t)\) be given such that \(\bar{c}(t) > 0\) for all \(0 \leq t \leq T\). Find a (possibly time-varying) routing strategy \(\beta(t)\) such that constraints PC are satisfied and
\[
x^{\beta}(t) \leq \bar{c}(t) \quad \forall \ t \in [0, T).
\]
where \(x^{\beta}(t)\) denotes the solution of (4.20) under routing strategy \(\beta(t)\).

### 4.5.4 Control Design

Here we develop a control design technique to solve Problem 4 which parallels the technique presented in Section 4.3.4. We begin by stating a theorem that gives constraints on the dynamics of system (4.20) which ensure that linear capacity constraints on the state are satisfied.

**Theorem 6** Modify the connectivity of the network by setting \(O_i = \{i\}\) for all \(i \in S_F\), that is, allow recirculation in the final sections of the network. Let the constraint vector \(c\) be given as in (4.4), an outflow function \(\mu\) satisfying assumptions 1 through 3, and \(\bar{x}_0 \leq c(0)\) be given. Define \(\tau_i = \frac{d}{c_i} \mu_i(0)\) for all \(i \in [n]\).
If there exist \(\beta(0)\) and \(\beta(T)\) such that constraints (2.7) - (2.9) are satisfied and
\[
A(\beta(0))c(0) + B^T d \leq m,
A(\beta(T))c(T) + B^T d \leq m,
\]
\[
\beta_{ii}(0) \geq 1 - \frac{\tau_i \mu_i(c_i(0))}{c_i(0)}, \quad \forall \ i \in [n],
\]
\[
\beta_{ii}(T) \geq 1 - \frac{\tau_i \mu_i(c_i(T))}{c_i(T)}, \quad \forall \ i \in [n],
\]
(4.24)

59
\[
\beta_{ii}(0) \leq 1, \; \forall \; i \in \mathcal{S}_F, \\
\beta_{ii}(T) \leq 1, \; \forall \; i \in \mathcal{S}_F,
\]

where \(A(\beta(0))\) and \(A(\beta(T))\) are defined as in (4.23), and \(\beta(t)\) is given as in (4.6) by

\[
\beta_{ij}(t) = \frac{1 - \frac{1}{T}}{1 - \frac{1}{T}} c_i(0) + \frac{1}{T} \beta_{ij}(T) c_i(T), \; \forall \; i, j \in [n], \; 0 \leq t \leq T,
\]

then the closed loop system (4.20) under the decentralized time-varying state feedback control policy

\[
\begin{align*}
\tilde{\beta}_{ii}(t) &= 1 - (1 - \beta_{ii}(t)) \frac{x_i(t)}{\tau_{i}\mu_i(x_i(t))}, \; \forall \; i \in [n] \\
\tilde{\beta}_{ij}(t) &= \beta_{ij}(t) - \frac{x_i(t)}{\tau_{i}\mu_i(x_i(t))}, \; \forall \; i, j \in [n], \; i \neq j
\end{align*}
\]

has the following properties

(i) The solution \(\tilde{x}^\beta(t)\) of system (4.20) is identical to the solution \(\tilde{x}^\beta(t)\) of system (4.22) with routing parameters \(\beta(t)\),

(ii) Positivity holds for system (4.20),

(iii) Conservation holds for system (4.20),

(iv) The solution \(\tilde{x}^\beta(t)\) of system (4.20) satisfies \(\tilde{x}^\beta(t) \leq c(t)\) for \(0 \leq t \leq T\).

**Proof 5** We first note that the feedback control policy defined by (4.27) and (4.28) was chosen so that the nonlinear system (4.20) in closed loop is identical to the linear system (4.22) with routing parameters \(\beta(t)\). To see this, we substitute the expressions for \(\tilde{\beta}(t)\) given in (4.27) and (4.28) into (4.19), which describes the dynamics of each section \(i \in [n]\) of the network,

\[
\begin{align*}
\dot{x}_i &= -\mu_i(x_i(t)) + \sum_{j: i \in \mathcal{O}_j, \; j \neq i} \beta_{ji}(t) \frac{x_j(t)}{\tau_j} \mu_j(x_j(t)) + \left\{ 1 - (1 - \beta_{ii}(t)) \frac{x_i(t)}{\tau_i\mu_i(x_i(t))} \right\} \mu_i(x_i(t)) \\
&\quad + \sum_{s=1}^{S} b_{si} \overline{d}_s \\
&= -\mu_i(x_i(t)) + \sum_{j: i \in \mathcal{O}_j, \; j \neq i} \beta_{ji}(t) \frac{x_j(t)}{\tau_j} + \mu_i(x_i(t)) - (1 - \beta_{ii}(t)) \frac{x_i(t)}{\tau_i} + \sum_{s=1}^{S} b_{si} \overline{d}_s \\
&= -\frac{x_i(t)}{\tau_i} + \sum_{j: i \in \mathcal{O}_j} \beta_{ji}(t) \frac{x_j(t)}{\tau_j} + \sum_{s=1}^{S} b_{si} \overline{d}_s,
\end{align*}
\]
which is the same as the description of the dynamics of each section \(i \in [n]\) of linear system (4.22) given in (4.21). Thus, if \(x(0) = \hat{x}(0) = \hat{x}_0\) then \(\hat{x}^\beta(t) = x^\beta(t)\) and (i) holds. Additionally, since inequalities (4.24) are the same as (4.5), and \(\beta(t)\) is recovered by (4.6), Theorem 4 can be applied to conclude that Positivity holds for system (4.22), that is \(\hat{x}(t) \geq 0\), and \(x^\beta(t) \leq c(t)\), for \(0 \leq t \leq T\). Given (i), these properties also hold for system (4.20), thus (ii) and (iv) hold.

Finally, we will show that Conservation holds by verifying that (2.7) through (2.9) hold for routing parameters \(\hat{\beta}(t)\), that is

\[
\hat{\beta}_{ij}(t) \geq 0, \forall i, j \in [n], \quad (4.29)
\]

\[
\hat{\beta}_{ij}(t) = 0, \forall j \in [n]\setminus O_i, \forall i \in [n], \quad (4.30)
\]

\[
\sum_{j \in O_i} \hat{\beta}_{ij}(t) = 1, \forall i \in [n]\setminus S_F. \quad (4.31)
\]

From Theorem 4, we know that (2.7) through (2.9) hold for \(\beta(t)\). It is easily seen that (4.30) holds whenever (2.8) holds for \(\beta(t)\). To check (4.31) we compute

\[
\sum_{j \in O_i} \hat{\beta}_{ij}(t) = \sum_{\substack{j \in O_i, \ j \neq i}} \hat{\beta}_{ij}(t) + \hat{\beta}_{ii}(t)
\]

\[
= \sum_{\substack{j \in O_i, \ j \neq i}} \beta_{ij}(t) \frac{\hat{x}_i(t)}{\tau_i \mu_i(\hat{x}_i(t))} + 1 - (1 - \beta_{ii}(t)) \frac{\hat{x}_i(t)}{\tau_i \mu_i(\hat{x}_i(t))}
\]

\[
= \sum_{j \in O_i} \beta_{ij}(t) \frac{\hat{x}_i(t)}{\tau_i \mu_i(\hat{x}_i(t))} + 1 - \frac{\hat{x}_i(t)}{\tau_i \mu_i(\hat{x}_i(t))}
\]

\[
= \frac{\hat{x}_i(t)}{\tau_i \mu_i(\hat{x}_i(t))} \left( \sum_{j \in O_i} \beta_{ij}(t) - 1 \right) + 1
\]

\[
= 1
\]

for all \(i \in [n]\setminus S_F\), making use of the fact that \(\hat{x}(t) \geq 0\).

Then clearly (4.29) holds when \(i \neq j\). In order to show that (4.29) holds when \(i = j\), we must first show that

\[
\beta_{ii}(t) \geq 1 - \frac{\tau_i \mu_i(c_i(t))}{c_i(t)}, \forall i \in [n], 0 \leq t \leq T, \quad (4.32)
\]

and

\[
\frac{\mu_i(c_i(t))}{c_i(t)} \leq \frac{\mu_i(\hat{x}_i(t))}{\hat{x}_i(t)} \quad (4.33)
\]
whenever \( \hat{x}_i(t) \leq c_i(t) \). To see that (4.32) holds, we look at the product of \( \beta_{ii}(t) \) and \( c_i(t) \). Given the definition of \( \beta(t) \) in (4.6) and the fact that \( c(t) \) is linear, we have

\[
\beta_{ii}(t)c_i(t) = \left(1 - \frac{t}{T}\right) \beta_{ii}(0)c_i(0) + \frac{t}{T} \beta_{ii}(T)c_i(T)
\]

\[
\geq \left(1 - \frac{t}{T}\right)\left(c_i(0) - \tau_i \mu_i(c_i(0))\right) + \frac{t}{T} \left(c_i(T) - \tau_i \mu_i(c_i(T))\right)
\]

\[
= c_i(t) - \left(1 - \frac{t}{T}\right) \tau_i \mu_i(c_i(0)) - \frac{t}{T} \tau_i \mu_i(c_i(T))
\]

\[
\geq c_i(t) - \tau_i \mu_i(c_i(t)).
\]

The first inequality follows from (4.25) and the final inequality comes from the fact that \( \mu_i \) is concave and thus

\[
\left(1 - \frac{t}{T}\right) \mu_i(c_i(0)) + \frac{t}{T} \mu_i(c_i(T)) \leq \mu_i(c_i(t)).
\]

And we see that (4.32) is satisfied.

In order to show (4.33) we use the facts that \( \mu_i \) is concave, \( \mu_i(0) = 0 \), \( 0 \leq \hat{x}_i(t) \leq c_i(t) \) which follows from (ii) and (iv), and \( \hat{x}_i(t) = \frac{\hat{x}_i(t)}{c_i(t)} c_i(t) + \left(1 - \frac{\hat{x}_i(t)}{c_i(t)}\right) \times 0 \), therefore

\[
\mu_i(\hat{x}_i(t)) \geq \frac{\hat{x}_i(t)}{c_i(t)} \mu_i(c_i(t)) + \left(1 - \frac{\hat{x}_i(t)}{c_i(t)}\right) \mu_i(0)
\]

\[
\mu_i(\hat{x}_i(t)) \geq \frac{\hat{x}_i(t)}{c_i(t)} \mu_i(c_i(t))
\]

\[
\frac{\mu_i(\hat{x}_i(t))}{\hat{x}_i(t)} \geq \frac{\mu_i(c_i(t))}{c_i(t)}
\]

Starting from (4.32) and the fact that \( \beta_{ii}(t) \leq 1 \), which follows from the combination of constraints (2.7) and (2.9) when \( i \in [n] \setminus S_T \) and from constraint (4.26) and the definition of \( \beta_{ii}(t) \) given in (4.6) when \( i \in S_T \),

\[
1 \geq \beta_{ii}(t) \geq 1 - \frac{\tau_i \mu_i(c_i(t))}{\hat{x}_i(t)}
\]

\[
1 \geq \beta_{ii}(t) \geq 1 - \frac{\tau_i \mu_i(\hat{x}_i(t))}{\hat{x}_i(t)}
\]

\[
-1 \leq -\beta_{ii}(t) \leq -1 + \frac{\tau_i \mu_i(\hat{x}_i(t))}{\hat{x}_i(t)}
\]

\[
0 \leq 1 - \beta_{ii}(t) \leq \frac{\tau_i \mu_i(\hat{x}_i(t))}{\hat{x}_i(t)}
\]

\[
0 \leq (1 - \beta_{ii}(t)) \frac{\hat{x}_i(t)}{\tau_i \mu_i(\hat{x}_i(t))} \leq 1
\]

\[
0 \leq \hat{\beta}_{ii}(t) \leq 1
\]

and (4.29) holds when \( i = j \).
Since (4.29) through (4.31) hold for \( \tilde{\beta}(t) \) we can conclude that Conservation holds for system (4.20) in closed loop, thus (iii) holds.

Remark 1  The idea motivating the development of Theorem 6 is that the outflow rates of each section of linear system (4.22) with \( \tau_i = \frac{d}{dx_i} \mu_i(0) \) for all \( i \in [n] \) can be restricted through constraints on recirculation so that the linear system behaves like system (4.20) with nonlinear outflow rates \( \mu_i(\tilde{x}_i(t)) \).

Now, as in Section 4.3.4.2, we formulate a procedure based on Theorem 6 to design routing parameters \( \tilde{\beta}(t) \) to solve Problem 4 for the nonlinear system (4.20). First, note that constraints (4.25) are nonlinear in the capacity bound \( c \). We must derive constraints which are linear in the optimization variables that can be used to ensure that constraints (4.25) are satisfied. Since, as in \( \phi(\bar{c}) \), the linear constraints will not involve \( \beta_{ii}(0) \) and \( \beta_{ii}(T) \) directly, but instead, \( z_{ii}(0) = \beta_{ii}(0)c_i(0) \) and \( z_{ii}(T) = \beta_{ii}(T)c_i(T) \), we would like to restate constraints (4.25) in terms of \( z_{ii} \) as follows

\[
\begin{align*}
    z_{ii}(0) &\geq c_i(0) - \tau_i \mu_i(c_i(0)), \forall \ i \in [n] \\
    z_{ii}(T) &\geq c_i(T) - \tau_i \mu_i(c_i(T)), \forall \ i \in [n].
\end{align*}
\]

We introduce \( \hat{\mu}_i \), a continuous piecewise linear under approximation of \( \mu_i \) defined by

\[
\hat{\mu}_i(y) = \begin{cases}
    \frac{y}{\tau_i}, & 0 \leq y < s^1_i, \\
    \frac{s^1_i}{\tau_i} + \frac{y-s^1_i}{\tau_i}, & s^1_i \leq y < s^2_i, \\
    \frac{s^1_i}{\tau_i} + \frac{s^2_i-s^1_i}{\tau_i} + \frac{y-s^2_i}{\tau_i}, & s^2_i \leq y < s^3_i, \\
    \vdots \\
    \frac{s^1_i}{\tau_i} + \frac{s^2_i-s^1_i}{\tau_i} + \cdots + \frac{s^{m-2}_i-s^{m-3}_i}{\tau_i}, & s^{m-2}_i \leq y < s^{m-1}_i, \\
    \frac{s^1_i}{\tau_i} + \frac{s^2_i-s^1_i}{\tau_i} + \cdots + \frac{s^{m-1}_i-s^{m-2}_i}{\tau_i}, & s^{m-1}_i \leq y,
\end{cases}
\]

(4.35)

where \( 0 < s^1_i \leq s^2_i \leq \cdots \leq s^m_i \) indicate the end points of linear segments of \( \hat{\mu}_i \) and \( 0 < \tau^1_i \leq \tau^2_i \leq \cdots \leq \tau^m_i \) are chosen such that \( \hat{\mu}_i \) is continuous, \( \hat{\mu}_i(y) \leq \mu_i(y) \), and assumptions 1 through 3 hold. Note that since \( \hat{\mu}_i \)
is concave, we can alternatively write

\[
\hat{\mu}_i(y) = \min \left\{ \frac{y}{\tau_i}, \frac{s_i^1}{\tau_i} + \frac{y-s_i^1}{\tau_i}, \frac{s_i^1}{\tau_i} + \frac{s_i^2-s_i^1}{\tau_i} + \frac{y-s_i^2}{\tau_i}, \ldots, \frac{s_i^1}{\tau_i} + \frac{s_i^2-s_i^1}{\tau_i} \ldots + \frac{s_i^{m-2}-s_i^{m-3}}{\tau_i^{m-2}} + \frac{y-s_i^{m-2}}{\tau_i^{m-1}}, \frac{s_i^1}{\tau_i} + \frac{s_i^2-s_i^1}{\tau_i} \ldots + \frac{s_i^{m-1}-s_i^{m-2}}{\tau_i^{m-1}} \right\}
\]

(4.36)

for all \( y \geq 0 \).

Since \( \hat{\mu}_i \) is an under approximation to \( \mu_i \)

\[
c_i(0) - \tau_i \hat{\mu}_i(c_i(0)) \geq c_i(0) - \tau_i \mu_i(c_i(0)), \quad \forall \ i \in [n],
\]

\[
c_i(T) - \tau_i \hat{\mu}_i(c_i(T)) \geq c_i(T) - \tau_i \mu_i(c_i(T)), \quad \forall \ i \in [n].
\]

Thus, imposing the constraints

\[
z_{ii}(0) \geq c_i(0) - \tau_i \hat{\mu}_i(c_i(0)), \quad \forall \ i \in [n]
\]

\[
z_{ii}(T) \geq c_i(T) - \tau_i \hat{\mu}_i(c_i(T)), \quad \forall \ i \in [n]
\]

(4.37)

will ensure that constraints (4.34) hold. We can formulate (4.37) as linear constraints in the control design variables since \( \hat{\mu}_i(y) \) can be calculated as the minimum of a finite number of expressions which are linear in \( y \), as given in (4.36).
In the spirit of LP problem (4.43), we can write the following LP problem to simultaneously solve for routing parameters and adjust constraints $\hat{c}$ to $\hat{c}$ when necessary

$$
\min \sum_{k=0}^{K-1} \sum_{t=1}^{N} (\hat{c}_i(t_k^+) - \hat{c}_i(t_k^-)) \Delta t
$$

subject to

$$
\phi(\hat{c})
$$

for all $k \in \{0, \ldots, K\}$

$$\hat{c}(t_k^+) \geq \hat{c}(t_k^-),$$

$$z_{ii}(t_k) \leq c_i(t_k), \ \forall \ i \in \mathcal{S}_r,$$

$$z_{ii}(t_k) \geq 1 - \tau_i \min \left\{ \frac{c_i(t_k)}{\tau_i^1}, \frac{s_1^1}{\tau_i^1} + \frac{c_i(t_k)-s_1^1}{\tau_i^2}, \frac{s_1^1}{\tau_i^1} + \frac{s_1^2-s_1^1}{\tau_i^2} + \frac{c_i(t_k)-s_1^2}{\tau_i^3}, \ldots \right\}, \ \forall \ i \in [n].$$

Parameters $\beta(t_k)$ can be recovered according to $\beta_{ij}(t_k) = \frac{z_{ii}(t_k)}{c_i(t_k)}$ for all $i, j \in [n], k = \{0, \ldots, K\}$ and interpolating non-linearly between $\beta_{ij}(t_k)$ and $\beta_{ij}(t_{k+1})$ over each interval $i_k$ according to

$$
\beta_{ij}(t) = \frac{1 - (t-t_k)}{\Delta t} \beta_{ij}(t_k) c_i(t_k) + \frac{t-t_k}{\Delta t} \beta_{ij}(t_{k+1}) c_i(t_{k+1})
$$

for all $t_k \leq t < t_{k+1}$. Noting that the constraints of LP problem (4.38) imply that (4.24), (4.25) and (4.26) are satisfied, we can conclude from Theorem 6 that under the feedback control policy defined by (4.27) and (4.28) system (4.20) is positive, conservative and the solution $\bar{x}^\beta(t)$ satisfies $\bar{x}^\beta(t) \leq \hat{c}(t)$ for $0 \leq t \leq T$.

### 4.5.5 Application Example

We applied the routing design strategy developed in Section 4.5.4 to a problem with the same network structure as that of Section 4.3.5. The network diagram is reproduced in Figure 4.11. The connectivity...
The outflow rate profile for each section is given by

\[
\mu_i(y) = \begin{cases} 
\frac{y}{\tau_i}, & 0 \leq y \leq \theta_i \\
\frac{\mu_{sat}^i}{1 + e^{-\rho_i(y - \theta_i)}}, & \theta_i \leq y
\end{cases}
\]

(4.39)

where \(\tau_i\) can be considered the nominal traversal time, \(\mu_{sat}^i\) is the saturation outflow rate. The value of \(\theta_i\) is chosen such that \(\mu_i(y)\) is continuous at \(y = \theta_i\), that is \(\theta_i = \frac{\tau_i \mu_{sat}^i}{2}\). The value of \(\rho_i\) is chosen such that the derivative of \(\mu_i(y)\) is continuous at \(y = \theta_i\), that is \(\rho_i = \frac{4}{\tau_i \mu_{sat}^i}\). Clearly, \(\mu_i(y)\) is increasing and concave for \(0 \leq y \leq \theta_i\) and \(\mu_i(0) = 0\). It can be shown that \(y = \theta_i\) is the inflection point of

\[
\frac{\mu_{sat}^i}{1 + e^{-\rho_i(y - \theta_i)}},
\]

which is concave and increasing for \(y \geq \theta_i\), thus this particular expression for \(\mu_i\) satisfies constraints 1 through 3 given in Section 4.5.2. In this example, \(\tau_i = 0.4\) aircraft per hour and \(\mu_{sat}^i = 20\) aircraft per hour which leads to \(\theta_i = 4\) aircraft and \(\rho_i = 0.5\) aircraft\(^{-1}\), for all \(i \in [n]\).

The inflow rate of sections 1, 2 and 3 is 15 aircraft per hour. Initial conditions were set to 10 material units for all sections in the top and bottom rows and 12 for all sections in the middle row. With this inflow and initial conditions, the state of every section remains below 15 aircraft when flows are routed along the rows of the network (i.e., when \(\beta_{1,4} = \beta_{4,7} = \ldots = \beta_{16,19} = 1\) and similar equalities hold for the second and third row of the network).
Each section except 14 has a constant capacity of 15, i.e., $\bar{c}_i(t) = 15$ material units for all $i \in [n]$ such that $i \neq 14$ and all $t \geq 0$. Section 14, on the other hand, has the piecewise constant capacity profile pictured in Figure 4.13, where each base interval has length $\Delta T = 30$ minutes.

The problem of designing routing parameters to ensure $\hat{x}$, the solution to system (4.20), remains below these capacity constraints, or some adjusted constraints if a feasible solution cannot be found for the given constraints, was solved using the method described in Section 4.5.4. Two different piecewise linear under approximations $\hat{\mu}_i(y)$ of $\mu_i(y)$ were used for each section $i \in [n]$. The first, shown in Figure 4.14(a) is composed of 3 linear segments, and the second, shown in Figure 4.14(b) is composed of 5 linear segments.

Each under approximation consists of a linear segment from $y = 0$ aircraft to $y = \theta_i = 4$ aircraft with a slope of $\frac{1}{\tau_i} = 2.5$ hour$^{-1}$, and thus matches $\mu_i(y)$ exactly in this region. Each under approximation is constant for $y \geq 15$ aircraft, equal to $\mu_i(15) = 19.2$ aircraft per hour. Between $y = 4$ aircraft and $y = 15$ aircraft, each under approximation is divided into 1 and 3 linear segments, respectively. The values at the end points of each segment of the under approximation are equal to the actual values of $\mu_i$ at those points.

We first focus on the 3 segment under approximation shown in Figure 4.14(a). Given the capacity constraint profile $\tilde{c}$ and $\tilde{\mu}$, we formulated linear constraints of LP (4.38) using a base interval for the piecewise linear capacity bound $c$ of $\Delta t = 15$ minutes. The given capacity constraints could not be satisfied exactly, and thus the solution of LP (4.38) includes adjusted capacity constraints $\hat{c}$. The integral of the difference between $\hat{c}$ and $\tilde{c}$, calculated as the cost of LP problem (4.38), is 1.30 aircraft × hour. That is, using this solution, the
Figure 4.14: Nonlinear section outflow rate $\mu$ and the two piecewise linear under approximations $\hat{\mu}$ of $\mu$ used in this example.

The actual section count will be above the constraint $\bar{c}$ by no more than an average of 1.30 aircraft over a one hour time period. The capacity constraint, adjusted capacity constraint and state of the closed loop system (4.20) are shown in Figure 4.15 for section 14 of the network. The corresponding routing parameters for select sections are plotted in Figure 4.16.

The routing solution found here is qualitatively similar to that of the problem solved in Section 4.3.5. In sections 1, 2 and 3, well upstream of the capacity constrained section, the majority of the section outflow is routed to the upper and lower sections of the graph. Closer to the capacity constrained section, in sections 10, 11 and 12, a larger portion of the section outflow is routed to the upper and lower sections of the graph.

The constraints in LP (4.38) depend on the particular choice of $\hat{\mu}$. Lower under approximation lead to more conservative constraints and a solution with a higher cost. To illustrate this, we solved the same problem as stated above with the particular choice of $\hat{\mu}$ shown in Figure 4.14(b). This is a less conservative under approximation of $\mu$ and should lead to a lower cost solution. Indeed, when using this particular $\hat{\mu}$ in solving LP (4.38) a routing strategy was found with a cost of 0.94 aircraft $\times$ hour, which is lower than the cost of the solution using $\hat{\mu}$ in Figure 4.14(a).
Figure 4.15: Capacity constraint $\bar{c}_{14}$, capacity bound $c_{14}$ and state $\tilde{x}_{14}$ of section 14.

Figure 4.16: Routing parameters associated with results plotted in Figure 4.2. Note that, due to symmetry of the problem, several of the routing parameters have identical profiles. Also recall that $\tilde{\beta}_{ij}(t)$ must sum to 1 for each section, and thus recirculation accounts for the remainder of the flow routing (i.e. $\tilde{\beta}_{1,1}(t) = 1 - \tilde{\beta}_{1,4}(t) - \tilde{\beta}_{1,5}(t)$, etc.).
4.6 Systems with a Fixed Linear Outflow Model and Multiple Destinations

4.6.1 Motivation

While the modeling and control design techniques of the previous sections have focused on single destination networks, here we address the problem of multiple destinations. Flights in a given region of airspace may be bound for several different destinations. When routing is used as the control input, care must be taken to ensure that each individual aircraft is able to reach its specified destination. The control design technique for single destination networks presented in Section 4.3.4 can be extended for use with networks with multiple destinations. In this section, modeling and control design techniques are presented to address the problem of routing traffic with multiple destinations.

As in Section 4.3, here we focus on control design for fixed linear outflow systems. The assumption that the outflow of each section is linear in the state of the section is reasonable when operating in the neighborhood of a steady state where the section traversal times are constant. Physically, this corresponds to low density traffic in which aircraft are free to fly at their nominal speeds.

4.6.2 Continuous Time Aggregate Model

In this section, we use the same linear outflow model with fixed section traversal times as was used in Chapter 3 and Section 4.3. However, given that each aircraft has a particular final destination, we aggregate aircraft within each section of the network based on final destination. That is, assuming that there are $R$ distinct destinations, we create $R$ sub-networks that describe the flow of traffic through the $n$ section network for each destination $r \in [R]$. The notation set forth in Chapter 2 is used here, however a superscript is introduced to indicate the associated destination. Network connectivity is specified for each destination. We denote by $O^r_i$, the set of sections into which the flights in section $i$ with destination $r$ can flow. The set of final sections for flights with destination $r$ is denoted by $S^r_F$. The state of each section $i \in [n]$ is broken down into $R$ parts, denoted by $x^r_i(t)$ for all $r \in [R]$. The value of $x^r_i(t)$ is the number of aircraft in section $i$ with destination $r$ at time $t$. Section outflow rates $f^r_i(x^r_i(t), t)$, routing parameters $\beta^r_{ij}(t)$, and routing of flow from each source into each section $b^r_{si}$, are all specified separately for each destination $r \in [R]$. A linear outflow model is used to describe the flow of aircraft with destination $r$ out of each section $i$ of the network with time invariant traversal times $\tau^r_i$, that is

$$f^r_i(x^r_i(t), t) = \frac{x^r_i(t)}{\tau^r_i}.$$
The dynamics of aircraft in section $i$ of the network with destination $r$ can be written as

$$
\dot{x}_r^i(t) = -\frac{x_r^i(t)}{\tau_i} + \sum_{j: i \in O^r_j} \beta_{r,ij}^r(t) \frac{x_r^j(t)}{\tau_j} + \sum_{s=1}^{S} b_{r,s} \delta_s(t),
$$

(4.40)

for all $i \in [n]$ and $r \in [R]$. For each destination $r \in [R]$, the dynamics of the $n$ section sub-network is then described by the following linear system

$$
\dot{x}^r(t) = A^r(\beta^r(t))x^r(t) + B^r d^r(t)
$$

(4.41)

$$
x^r(0) = x^r_0
$$

where

$$
A^r(\beta^r(t)) = A^r_0 + \sum_{i=1}^{n} \sum_{j \in O^r_i} \frac{\beta_{r,ij}^r(t)}{\tau_i} e_j e_i^T,
$$

(4.42)

and $A^r_0 = \text{diag}\left(-\frac{1}{\tau_1}, \ldots, -\frac{1}{\tau_n}\right)$.

Since capacity constraints will be specified for the total number of aircraft in the section, we define the state of the full network as the sum of the states of the sub-networks for each destination. That is, the state of the full network, considering all destinations, $x$ is defined by

$$
x(t) = \sum_{r=1}^{R} x^r(t).
$$

Let $\beta$ represent the routing strategy $\beta_{r,ij}^r$ for each $r \in [R]$ and $i, j \in [n]$. We then designate the solution of the full system under routing strategy $\beta$ by $x^\beta(t)$.

### 4.6.3 Control Objectives

With the dynamics of the sub-networks associated with each destination and the full network state defined, we can now formally state the problem of interest.

**Problem 5** Let a piecewise constant vector-valued function $t \mapsto \bar{c}(t)$ be given such that $\bar{c}(t) > 0$ for all $0 \leq t \leq T$. Find a (possibly time-varying) routing strategy $\beta(t)$ such that constraints PC are satisfied for system (4.41) for all $r \in [R]$ and

$$
x^\beta(t) \leq \bar{c}(t) \ \forall \ t \in [0,T).
$$

where $x^\beta(t)$ denotes the state of the full network under routing strategy $\beta(t)$. 
4.6.4 Control Design

The proposed control design technique essentially is the application of the control design technique presented in Section 4.3.4 for the fixed linear outflow model with a single destination, to each sub-network. These sub-problems are coupled through the section capacity constraints. Associated with each sub-network $r \in R$, there is a continuous, positive, piecewise linear capacity bound, $c^r(t)$. Over each time interval $i_k$ for $k = 0, \ldots, K - 1$, we parametrize this function as

$$c^r(t) = c^r(t_k) + (t - t_k)m^r(t_k^+)$$

for all $r \in [R]$ and $t_k \leq t < t_{k+1}$ where $m^r$ is constant over intervals $i_k$. We define $c^r$ at the end time of each interval $i_k$ as

$$c^r(t_{k+1}) = c^r(t_k) + \Delta t m^r(t_k^+)$$

for all $r \in [R]$ to ensure continuity. Rather than applying a capacity constraint for each sub-network, the capacity constraint is imposed on the sum of the capacity bounds for each sub-network. That is, we would like to ensure that

$$\sum_{r=1}^{R} c^r(t) \leq \bar{c}(t).$$

The linear constraints $\phi(c)$ are thus modified to form the following set of linear constraints, $\phi_R(c)$.

$$c^r(t_0) \geq x_0^r, \forall r \in [R]$$

for all $k \in \{0, \ldots, K\}$

$$\sum_{r=1}^{R} c^r(t_k) \leq \min\{c(t_k^-), c(t_k^+)\},$$

for all $k \in \{0, \ldots, K\}, r \in [R]$

$$c^r(t_k) \geq 0,$$

$$z_{ij}^r(t_k) \geq 0, \forall i, j \in [n],$$

$$z_{ij}^r(t_k) \leq c_i^r(t_k), \forall i, j \in [n],$$

$$z_{ij}^r(t_k) = 0, \forall j \in [n] \setminus \mathcal{O}_i, \forall i \in [n],$$

$$\sum_{j=1}^{n} z_{ij}^r(t_k) = c_i^r(t_k), \forall i \in [n] \setminus \mathcal{F},$$

$$B_r^T d(t_k^+) + A_r^0 c^r(t_k) + \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{z_{ij}^r(t_k)}{\tau_j} e_j \leq m^r(t_k^+),$$

for all $k \in \{0, \ldots, K - 1\}, r \in [R]$.

$$c^r(t_{k+1}) = c^r(t_k) + \Delta t m^r(t_k^+),$$

$$B_r^T d(t_{k+1}^-) + A_r^0 c^r(t_{k+1}) + \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{z_{ij}^r(t_{k+1})}{\tau_j} e_j \leq m^r(t_k^+).$$
For any feasible point of $\phi_R(\bar{c})$, $\beta^r(t_k)$ can be recovered according to $\beta^r_{ij}(t_k) = \frac{z^r_{ij}(t_k)}{c^r_i(t_k)}$ for all $r \in [R]$, $i, j \in [n]$, $k = \{0, \ldots, K\}$. Routing strategy $\beta^r(t)$ can be found by interpolating non-linearly between $\beta^r_{ij}(t_k)$ and $\beta^r_{ij}(t_{k+1})$ over each interval $i_k$ according to

$$
\beta^r_{ij}(t) = \frac{(1 - \frac{t-t_k}{\Delta t}) \beta^r_{ij}(t_k)c^r_i(t_k) + \frac{t-t_k}{\Delta t} \beta^r_{ij}(t_{k+1})c^r_i(t_{k+1})}{(1 - \frac{t-t_k}{\Delta t}) c^r_i(t_k) + \frac{t-t_k}{\Delta t} c^r_i(t_{k+1})}
$$

for all $r \in [R]$ and $t_k \leq t < t_{k+1}$.

With $\beta^r(t)$ defined in this way and constraints $\phi_R(\bar{c})$, Theorem 4 can be applied to each sub-network to conclude that $A^r(\beta^r(t))$ is PC for all $0 \leq t < T$ and

$$x^r(t) \leq c^r(t).$$

Additionally, the second constraint of $\phi_R(\bar{c})$ and the fact that $c^r(t)$ is piecewise linear for all $r \in [R]$, we can conclude that $x^r \leq \bar{c}(t)$ for all $0 \leq t < T$.

As proposed in Section 4.3.4.2, if linear constraints $\phi_R(\bar{c})$ are infeasible, it is natural to try to alter the desired capacity $\bar{c}$ so as to find a feasible solution, while ensuring that the resulting bounds are close to $\bar{c}$ in some sense. In such a case, we allow $\bar{c}$ to be increased to $\hat{c}$ over intervals of length $\Delta t$ in order to achieve feasible capacity bounds below $\hat{c}$. The problem of finding capacity constraints $\hat{c}$ and capacity bound $c$ can then be written as the following LP problem:

$$
\min \sum_{k=0}^{K-1} \sum_{i=1}^{N} (\hat{c}_i(t^+_k) - \bar{c}_i(t^+_k)) \Delta t
$$
subject to $\phi_R(\hat{c})$

$$\hat{c}(t^+_k) \geq \bar{c}(t^+_k), \ k \in \{0, \ldots, K\}. \tag{4.43}
$$

### 4.6.5 Application Example

In order to illustrate the modeling and control design techniques presented in Section 4.6.2 and Section 4.6.4, respectively, these techniques were applied to an example problem. The full airspace network used in this example is shown in Figure 4.17(a). The network consists of 21 airspace sections, three sources (or origins) and three sinks (or destinations). Arrows indicate allowable flow between sources, sections and sinks. Recirculation is allowed in all sections except final sections. Final sections are 19, 20 and 21. The outflow of section 19 flows into sink 1, the outflow of section 20 flows into sink 2 and the outflow of section 21 flows into
sink 3. Given that there are three distinct sinks (or destinations) in this problem, the full network is divided into three sub-networks. Sub-networks for destinations 1, 2 and 3 are given in Figure 4.17(b), Figure 4.17(c) and Figure 4.17(d), respectively. Notice that each sub-network has fewer than 21 sections. This is due to the fact that specific destinations are unreachable from certain sections in the full network. For instance, focusing on Figure 4.17(b), depicting the sub-network associated with destination 1, it is not possible to travel from sections 18, 20 and 21 to sink 1 given the section interconnection of the full network in Figure 4.17(a). Thus, these sections are omitted from the destination 1 sub-network and flow from other sections in the sub-network to these omitted sections is not allowed.

Initial conditions and inflow rates are depicted graphically in Figure 4.18. Inflow rates are constant for the duration of the planning horizon. The initial states and inflow rates for the full network, as depicted in Figure 4.18(a), are identical to those used in the example in Section 4.3.5. Flow rates and initial conditions are broken down by destination in Figure 4.18(b), Figure 4.18(c) and Figure 4.18(d). That is, in Figure 4.18(b), the inflow rates specified are the inflow rates of flights with destination 1. The values in the boxes in figure
Figure 4.18: Figures 4.18(a), 4.18(b), 4.18(c), and 4.18(d) indicate inflow rates and section initial conditions for all flights in the network and flights with destination 1, 2 and 3, respectively. The inflow at sources is indicated in the ovals on the left of each diagram. The value in each box represents the initial number of aircraft in that section.

Figure 4.18(b) indicate the initial number of aircraft in each section with destination 1, mathematically, this is $x_i^1(0)$ for all $i \in [n]$.

As in the example presented in Section 4.3.5, we chose traversal time $\tau_i^r = 0.4$ hours for all $i \in [n]$ and $r \in [R]$. Each section, except for section 14, has a constant capacity constraint of 15 aircraft, i.e. $\bar{c}_i(t) = 15$ aircraft for all $i \in [n] \setminus \{14\}$ and all $t \geq 0$. Section 14 has the piecewise constant capacity constraint profile pictured in Figure 4.19(a), in which each base interval has a length of $\Delta T = 30$ minutes.

Based on these capacity constraint profiles, we formulated constraints $\phi_R(\bar{c})$ using a base interval for the piecewise linear capacity bounds, $c^r$ for all $r \in [R]$ of $\Delta t = 15$ minutes. The constraints $\phi_R(\bar{c})$ were found to be feasible, which, in turn, implies that problem 5 has a solution. The given capacity constraint, and resulting capacity bounds and simulated states for section 14 are shown in Figure 4.19. The capacity constraint, $\bar{c}_{14}(t)$, capacity bound $c_{14}(t) = \sum_{r=1}^{R} c^r_{14}(t)$ and full state, $x_{14}(t)$, of section 14 are shown in Figure 4.19(a). Capacity bounds $c^r_{14}(t)$ and state $x^r_{14}(t)$ are also given for $r = 1, 2, 3$ in Figure 4.19(b), Figure 4.19(c), and
Figure 4.19: Capacity constraint, resulting capacity bounds and full network state for section 14 in Figure 4.19(a). Resulting capacity bounds and sub-network states for section 14 and destinations 1, 2, and 3, respectively, in Figures 4.19(b), 4.19(c), and 4.19(d).

Figure 4.19(d), respectively. Each sub-network problem involves finding routing parameters $\beta_{ij}(t)$ such that $x_{ri}(t) \leq c_{ri}(t)$ for $0 \leq t \leq T$ for all $r \in [r]$, $i \in [n]$. Figures 4.19(b), 4.19(c), and 4.19(d) indicate the resulting capacity bounds and states for section 14. These three sub problems are coupled through the constraint that $\sum_{r=1}^{R} c_{i}(t) \leq \bar{c}_{i}(t)$. 
Chapter 5

Sliding Mode Control Design for Flow Rate Constrained Systems

5.1 Motivation and Problem Description

In this chapter, we present a method which can be used to design distributed control policies for air traffic systems to match a time varying maximum outflow capacity, when possible. When this is not possible, i.e. when the maximum flow rate through the network is less than the maximum outflow capacity, flights are allowed to fly through the network with no controlled flow restrictions. This method relies on controllers in charge of different sections in the network to locally direct flow. By “distributed” we mean that controllers need only share a small amount of information regarding the states of their sections with each other. Our design strategy builds on techniques from the theory of sliding mode control. Some additional restrictions must be imposed on the resulting sliding mode controller to ensure positivity of the control input. The resulting control strategy is presented as an autonomous hybrid automaton. We provide a guarantee that this control strategy produces a tracking error between the true and maximum outflow which converges to zero, or falls below zero with no controlled flow restrictions, in finite time. A preliminary version of this control design technique is presented in [9].

We present the specifics of the network structure used and describe the aggregate dynamics of flights traveling through the network in Section 5.3. Design requirements and motivation for these requirements are then given. In Section 5.4 we introduce our choice for the structure of the controller, which is motivated by the design requirements stated in Section 5.3. The proposed control law is presented as an autonomous hybrid automaton, which is formally defined in Section 5.5. The control scheme is then shown to satisfy the specified
design requirements over Sections 5.6, 5.7 and 5.8. In order to demonstrate the applicability of our method, we use it to design a control law for a simple network of air traffic flow, results of which are presented in Section 5.9.

5.2 Continuous Time Aggregate Model

We use the continuous time aggregate model presented in Section 2.2 to describe the flow of aircraft through a network of interconnected airspace. As mentioned in Section 2.1, in order to ensure stability of the system, that is, that all aircraft eventually exit the network, the network must be outflow connected. That is, there must exist at least one path from each section in the network to a sink. We make the additional assumption that the interconnection of sections is such that aircraft leaving a given section enter subsequent sections that are at least as close or closer to the sink than the section itself and that at least one of these sections is strictly closer to the sink. For each section $i$ in the network, let $l_i$ be the length of the shortest path from section $i$ to a sink of the network. This requirement can be described mathematically by requiring for each $i \in [n]$ that

$$l_j \leq l_i \forall j \in \mathcal{O}_i,$$

$$\min_{j \in \mathcal{O}_i} l_j < l_i.$$

This organization of sections will be used to prove certain closed loop properties of the system. An example of this type of network structure is given in Figure 5.1.

We use a model with linear outflow with a fixed traversal time, that is

$$f_i(x_i(t), t) = \frac{x_i(t)}{\tau_i}.$$
for all \( i \in [n] \). Flights in each section are assumed to be traveling at the maximum allowable speed, corresponding to a section traversal time of \( \tau_i \). The control input used is airborne delay, which is described in detail in Section 2.2.1. Airborne delay is introduced into the mathematical model through recirculation parameter \( u_i \). The controlled outflow of each section \( i \in [n] \) can then be written as

\[
    f_i(t) = \frac{x_i(t)}{\tau_i} - u_i(t).
\]

Routing parameters \( \beta \) are defined as in Section 2.2. These parameters are fixed and given and assumed to satisfy constraints (2.7) through (2.9). The use of fixed routing parameters is only appropriate when considering operations around a given regime with routing control performed at a higher level. For example, in [35] routing parameters are considered to be fixed and it is suggested that appropriate instantaneous values of these parameters can be found using a simulation tool such as FACET (see [17]).

The time-varying rate at which flights enter the system from source \( A \) is denoted by \( d \). Finally, \( 0 \leq b_{Ai} \leq 1 \) denotes the fraction of the inflow rate from source \( A \) that enters into section \( i \).

The dynamics of section \( i \in S_L \) can now be given by

\[
    \dot{x}_i(t) = -\frac{x_i(t)}{\tau_i} + u_i(t) + \sum_{j:\ j \in O_i} \beta_{ji} \left( \frac{x_j(t)}{\tau_j} - u_j(t) \right) + b_{Ai} d(t).
\] (5.1)

### 5.3 Control Objectives

The global outflow from sections in the final level, namely \( z := \sum_{i \in S_F} f_i \), is used as a performance output. The high level objective is to determine a feedback control policy such that the global outflow \( z \) is as large as possible without exceeding the maximum allowable outflow \( z_d \). The control design problem is then to determine \( u = (u_1, ..., u_n)^T \) such that the basic control design objectives presented in Section 2.2.2 and additional performance requires are satisfied. These requirements are listed below.

1. **Stability:** The system is internally stable in closed-loop.

2. **State Positivity:** The state is positive, that is

\[
    x_i(t) \geq 0, \ \forall \ t \geq 0, \ \forall \ i.
\]
3. **Constrained Control:** The control input $u_i(t)$ must satisfy

$$0 \leq u_i(t) \leq \frac{x_i(t)}{\tau_i}, \quad \forall \ t \geq 0, \ \forall \ i.$$ 

4. **Decentralization:** The control input $u_i(t)$ should depend only on $x_i(t)$, the maximum allowable capacity $z_d$, and a small number of other components of the state vector $x_j, j \neq i$.

5. **Satisfactory Output:** The output $z$ is as large as possible without exceeding $z_d + \epsilon$ where $\epsilon$ is a prescribed error bound. More precisely for $t$ large enough $z(t)$ should satisfy one of the following two alternatives:

   (a) **Matching:** The output $z$ tracks the maximum allowable capacity $z_d$ within $\epsilon$, that is

   $$|z(t) - z_d(t)| = |\epsilon(t)| \leq \epsilon.$$

   (b) **Maximum Outflow:** If the uncontrolled outflow $\sum_{i \in S_F} \frac{F_i}{\tau_i}$ is less than $z_d - \epsilon$ then the control input for each section $i$ is zero. Mathematically, this can be written as

   $$\sum_{i \in S_F} \frac{F_i}{\tau_i} < z_d(t) - \epsilon \Rightarrow u_i(t) = 0, \ \forall \ i.$$ 

The significance of Design Requirements 1 and 2 was discussed in Section 2.2.2. Since the control input $u_i$ acts only to recirculate a fraction of the nominal outflow of section $i$, it must satisfy Design Requirement 3. In order to limit the amount of inter-section communication, $u_i(t)$ should not depend on all the components of the state but instead depend on $x_i(t)$ and some small number of other components of the state vector and the maximum allowable capacity $z_d$, as stated in Design Requirement 4.

We define the desired behavior of the closed-loop system in Design Requirement 5. The quantity $z_d$ can be thought of as a maximum allowable outflow rate or a consumption rate. In the case that $z_d$ represents a maximum allowable outflow rate, as in an air traffic flow management problem, it is reasonable to assume that there is also some cost associated with the time required for flights to travel from the source to sink. In such a problem, the maximum allowable outflow rate could represent the maximum allowable landing rate at an airport and airborne delays are costly for airlines. In this case, we would like to utilize the outflow capacity to the fullest extent in order to reduce the need for airborne holding. Alternatively, $z_d$ can be considered to be a consumption rate, as in an irrigation network. In this type of problem, we would like to supply water to the sink at the prescribed consumption rate. The interpretation of $z_d$ as either an outflow
capacity or a consumption rate lead to the goal of finding a control strategy such that the output of the system \( z \) matches \( z_d \). Hence, Design Requirement 5a.

However, since we assume that the input to the system \( d \) is fixed and not under our control, it will not be possible, in general, for the system output to match \( z_d \). That is, even with flights traveling through the network at the maximum possible rate (in which case \( z = \sum_{i \in S} \frac{x_i}{\tau_i} \)), \( z \) may still be below \( z_d \). In such a case we would like the control input to be zero for all sections, which corresponds to flights traveling through the network at the maximum rate. This control objective is captured in Design Requirement 5b. In other words, a control strategy which satisfies Design Requirement 5 ensures that the network outflow either matches the maximum allowable outflow (with an error within \( \epsilon \) of zero) or, if the maximum allowable outflow rate is too large, the instantaneous outflow rate is maximum.

Since the outflow of the system \( z \) represents a flow rate of air traffic, which must be a nonnegative quantity, the maximum allowable outflow \( z_d \) must also be nonnegative. We need to ensure that some portion of the flights in each section are always allowed to exit the network to avoid having flights build up in the network. Thus, we require that \( z_d \) be bounded away from zero by some positive constant. Additionally, we require that the time derivative of \( z_d \) be bounded. The rationale for this constraint will be illustrated in Section 5.7.1. Thus, we assume that the maximum allowable outflow \( z_d \) satisfies the following constraint.

**Constraint 1** For some constants \( \bar{z}_d > 0 \) and \( G \geq 0 \),

\[
\begin{align*}
  z_d(0) &\geq \bar{z}_d, \\
  \dot{z}_d(t) &= g(t), \quad \forall \ t \geq 0, \\
  |g(t)| &\leq G, \quad \forall \ t \geq 0, \\
  z_d(t) &= \bar{z}_d \Rightarrow g(t) \geq 0.
\end{align*}
\]

Notice that these constraints ensure that \( z_d(t) \geq \bar{z}_d \) for all \( t \geq 0 \). First note that \( z_d(t) \) is a solution to a differential equation with bounded derivative and is thus continuous. Since \( z_d(0) \geq \bar{z}_d \), in order for \( z_d(t) < \bar{z}_d \) there must exist some \( t^* > 0 \) such that \( z_d(t^*) = \bar{z}_d \) and \( \dot{z}_d(t^*) < 0 \). However, this is a contradiction to the final constraint, \( z_d(t) = \bar{z}_d \Rightarrow g(t) \geq 0 \). Thus, \( z_d(t) \geq \bar{z}_d \) for all \( t \geq 0 \).
5.4 Control Strategy and Closed-Loop Properties

5.4.1 Controller Form

In order to achieve the control design objectives set forth in Design Requirements 1 - 5, we propose to use a control law of the form

\[ u_i(t) = \alpha(t) \frac{x_i(t)}{\tau_i}, \quad \forall i, \tag{5.2} \]

for some function \( \alpha \) to be determined later. The rationale for the structure of (5.2) is that (i) it agrees with the interpretation that \( u_i \) is a fraction of the outflow and the satisfaction of Design Requirement 3 is easily recognized, (ii) if we could make \( \alpha \) constant, this structure would give a fully decentralized controller. Notice that in this form, the same recirculation rate \( \alpha \) is applied to all sections in the network. A more practical and realistic motivation for this choice is that we would like to mitigate the build up of flights in individual sections compared to strategies which only control the outflow of the last sections.

With \( u_i \) defined per (5.2), the constraints on \( u_i \) given in Design Requirement 3 are equivalent to the following constraints on \( \alpha \)

\[ 0 \leq \alpha(t) \leq 1, \quad \forall t. \tag{5.3} \]

Substituting the expression for the control input given in (5.2) into equation (5.1), we see that the closed-loop dynamics of each section is

\[ \dot{x}_i(t) = -(1 - \alpha(t)) \frac{x_i(t)}{\tau_i} + \sum_{j \in \Omega_j} \beta_{ji}(1 - \alpha(t)) \frac{x_j(t)}{\tau_j} + b_{Ai}d(t). \tag{5.4} \]

To simplify later developments, we need to introduce some new notation. Due to the specific structure of the interconnected system illustrated in Figure 5.1, the output of the system is the controlled outflow of the system at sink \( B \) and depends only on the flights in \( S_F \). We define the vector \( w \) such that \( w_i = \frac{1}{\tau_i} \) for \( i \in S_F \) and \( w_i = 0 \) otherwise. We combine the dynamics of all individual sections specified by (5.4) in the form

\[ \dot{x}(t) = (1 - \alpha(t)) A x(t) + B_d d(t), \]

\[ z(t) = (1 - \alpha(t)) w^T x(t). \tag{5.5} \]

where matrices \( A, B_d \) are derived from the interconnection of subsystems (5.4).
5.5 Autonomous Hybrid Automaton Formulation

We propose a switching control law for $\alpha$ leading to the following hybrid autonomous automaton in closed-loop. By “autonomous” we mean that the automaton evolves with no external input. This automaton is meant to be a statement of the proposed control law, without proof of any desirable properties. The fact that this control strategy does, in fact, satisfy Design Requirements 1 - 5 will be shown over the following sections. We follow the formulation set forth in [46] while developing this autonomous hybrid automaton.

5.5.1 Automaton Definition

We define the automaton representing the closed-loop system as the octuple

$$H = (Q, X, \text{Init}, f, \text{Dom}, E, G, R)$$

where

- $Q$ is a set of discrete states;
- $X$ is a set of continuous states;
- $\text{Init} \subseteq Q \times X$ is a set of initial states;
- $f : Q \times X \rightarrow \mathbb{R}^{n+3}$ is a vector field;
- $\text{Dom} : Q \rightarrow 2^X$ is a domain;
- $E \subseteq Q \times Q$ is a set of edges;
- $G : E \rightarrow 2^X$ is a guard condition;
- $R : E \times X \rightarrow 2^X$ is a reset relation,

and $2^X$ denotes the power set (set of all subsets) of $X$. The particular choice of these parameters is described below. A graphical representation of $H$ is shown in Figure 5.2.

Since the input to the system $d$ depends on time and the derivative of the allowable outflow $z_d$ is an explicit function of time, time must be included as a state variable of the automaton. Thus we include the variable $s$ as a timer. We define an augmented state vector $x$ which includes the outflow capacity $z_d$, control parameter
Figure 5.2: Graphical representation of the autonomous hybrid automaton $H$. 

\[
\mathbf{x} = \begin{bmatrix}
(1 - \alpha) \mathbf{A} \mathbf{x} + \mathbf{B}(\mathbf{d}(s)) \\
g(s) \\
\eta(x, s) \text{sat}(\frac{\mathbf{e}}{\xi}) \\
1
\end{bmatrix}
\]

- $|\mathbf{e}| > \varepsilon$
- $0 \leq \alpha < 1$
- $(|\mathbf{e}| = \varepsilon) \wedge (\alpha > 0)$
- $(e < 0) \wedge (\alpha = 0)$
- $\mathbf{x} = \mathbf{x}$
\( \alpha \), and time along with the state vector \( x \) as follows

\[
x = \begin{bmatrix} x \\ z_d \\ \alpha \\ s \end{bmatrix}.
\]

The domain of the augmented state vector \( x \) is

\[
X = \left\{ x = \begin{bmatrix} x \\ z_d \\ \alpha \\ s \end{bmatrix} \in \mathbb{R}^{n+3}_+ : z_d \geq \bar{z}_d, \ 0 \leq \alpha < 1 \right\}.
\]

(5.6)

We define the set of discrete states, or modes, of the controller as \( Q = \{ \text{reaching, sliding, off} \} \). We refer to \( (q,x) \in Q \times X \) as the state of the automaton.

While in the reaching phase, the absolute value of the error is greater than \( \epsilon \), \( \alpha > 0 \) and \( \alpha \) varies according to (5.8). The error is within \( \epsilon \) of zero while in the sliding phase, and again, \( \alpha \) varies according to (5.8). While in the phase referred to as "off," \( e < 0 \), \( \alpha = 0 \) and \( \dot{\alpha} = 0 \), i.e. the controller is off. More precisely, for each \( q \in Q \) the dynamics of the augmented state vector \( x \) will evolve according to \( \dot{x} = f(q,x) \) where

\[
\begin{align*}
f \text{ (reaching, } x) &= \begin{bmatrix} (1 - \alpha) Ax + B_d(d(s)) \\ g(s) \\ \eta(x,s) \text{sat } (\xi) \\ 1 \end{bmatrix}, \\
f \text{ (sliding, } x) &= \begin{bmatrix} (1 - \alpha) Ax + B_d(d(s)) \\ g(s) \\ \eta(x,s) \text{sat } (\xi) \\ 1 \end{bmatrix}, \\
f \text{ (off, } x) &= \begin{bmatrix} Ax + B_d(d(s)) \\ g(s) \\ 0 \\ 1 \end{bmatrix}.
\end{align*}
\]
and
\[ \eta(x, t) = \eta_0 + \frac{1}{\tau_{\min}} + c_0 + \frac{1}{\tau_{\min}} \sum_{j \in S_F} \frac{x_j}{w^T x} + G \]  
(5.7)
for some constants \( \eta_0 \geq 0, \ c_0 > 0, \)
\[ \tau_{\min} = \min_{i \in S_m} \{ \tau_i \} \]
and the saturation function defined as follows
\[ \text{sat}(y) = \begin{cases} 
  y, & \text{if } |y| \leq 1 \\
  \text{sgn}(y), & \text{if } |y| > 1 
\end{cases} \]
for some chosen \( \epsilon > 0, \) where \( S_F \) is the set of sections which lead to sections in \( S_F, \) that is \( \cup_{i \in S_F} O_i = S_F. \)
The motivation for this choice of dynamics for \( \alpha, \) namely
\[ \dot{\alpha} = \eta(x, t) \text{sat} \left( \frac{e}{\epsilon} \right) \]
(5.8)
comes from sliding mode control theory and will be made clear in Section 5.7.1.

For each \( q \in Q \) the domain of the augmented state vector is
\[
\text{Dom (reaching)} = \{ x \in X : |e| = |(1 - \alpha)w^T x - z_d| > \epsilon, 0 \leq \alpha < 1 \}
\]
\[
\text{Dom (sliding)} = \{ x \in X : |e| = |(1 - \alpha)w^T x - z_d| \leq \epsilon, 0 \leq \alpha < 1 \}
\]
\[
\text{Dom (off)} = \{ x \in X : e = w^T x - z_d < \epsilon, \alpha = 0 \}.
\]
For all \( q \in Q \) we define
\[ \mathcal{U}_q = \{ q \} \times \text{Dom}(q) \subseteq Q \times X. \]
Notice that
\[ \text{Dom(reaching)} \cup \text{Dom(sliding)} \cup \text{Dom(off)} = X. \]
Therefore, for any initial value of the continuous state, \( x_0 \in \{ x \in X : s = 0 \}, \) there exists a \( q_0 \in \{ \text{reaching, sliding, off} \} \) such that \( x_0 \in \text{Dom}(q_0). \) Thus the set of possible initial states of the automaton is
\[ \text{Init} = \{(q_0, x_0) \in Q \times X : s = 0, x_0 \in \text{Dom}(q_0)\}. \]
Discrete transitions of the automaton from state $q \in Q$ to state $q' \in Q$ take place whenever the continuous state $x$ belongs to the guard $G(q, q')$ where the guards are defined by

$$G(\text{reaching, sliding}) = \{x \in X : (|e| = \epsilon) \land (\alpha > 0)\}$$

$$G(\text{reaching, off}) = \{x \in X : ((e < 0) \land (\alpha = 0)) \lor (w^T x < \bar{z}_d - \epsilon)\}$$

(5.9)

$$G(\text{sliding, off}) = \{x \in X : (e < 0) \land (\alpha = 0)\}$$

$$G(\text{off, sliding}) = \{x \in X : (e = \epsilon)\}.$$

We make the convention that at $t = 0$ the guards are evaluated and any transitions occur before continuous evolution begins. Notice that $\eta(x, t)$ is undefined when $w^T x(t) = 0$. This motives the need for the guard that causes the automaton to transition to the off mode when $w^T x(t) < \bar{z}_d - \epsilon$ to ensure that $\eta(x, t)$ is well defined whenever it is to be calculated.

When any transition takes place, the continuous state of the automaton is updated according to the forward relations defined as follows

$$R((\text{reaching, sliding}), x) = R((\text{sliding, off}), x)$$

$$= R((\text{off, sliding}), x)$$

$$= \{x\},$$

(5.10)

$$R((\text{reaching, off}), x) = \begin{bmatrix} x \\ \bar{z}_d \\ 0 \\ s \end{bmatrix},$$

where

$$x' = R((q, q'), x) \in \text{Dom}(q') \subseteq X$$

is the updated value of the continuous state after a transition of the discrete state from $q \in Q$ to $q' \in Q$. Notice that the continuous state is only reset during a transition from reaching to off. This is due to the fact that when such a transition is triggered by $w^T x$ satisfying $w^T x < \bar{z}_d - \epsilon$, $\alpha$ may not necessarily be equal to zero. Thus, $\alpha$ must explicitly be set to zero to ensure that $x'$ is in Dom(off).
5.5.2 Reformulation of Design Requirements in Terms of the Automaton

In order for the automaton defined in Section 5.5.1 to be both well posed and satisfy Design Requirements 1 - 5, several properties must hold. First, we must show that the automaton is well posed, that is that the transitions described are the only transitions possible. Second, we must show that for any initial state of the automaton, \((q_0, x_0) \in \text{Init}\), the state of the automaton will remain in \(Q \times X\).

When the particular form of maximum allowable outflow \(z_d\) was introduced in Section 5.3 through Constraint 1 we showed that if \(z_d(0) \geq \bar{z}_d\) then \(z_d(t) \geq \bar{z}_d\) for all \(t \geq 0\). With \(s(0) = 0\), the dynamics of \(s\) defined by the automaton ensure that \(s \geq 0\). What must be shown to prove that \(H\) is domain preserving is that \(x(t) \geq 0\) and \(0 \leq \alpha(t) < 1\) for all \(t \geq 0\) and that transitions occur in such a way that \(x(t) \in \text{Dom}(q(t))\) for each \((q(t), x(t))\) for all \(t \geq 0\).

In order for this control law to be well behaved, we must also show that there cannot be an infinite number of switches between discrete states in finite time. Physically, this means that once the automaton has made a transition, it will remain in the current mode for a non-zero length of time. In automaton theory, this property is referred to as “non-Zeno”.

The properties described above along with Design Requirements 1 - 5 will be shown to hold over the following sections.

5.6 Closed-Loop Properties

In this section, we focus on the closed-loop properties of the state \(x\), Design Requirements 1 and 2. Notice that the dynamics of the state variable \(x\) defined by the automaton are either

\[
\dot{x} = (1 - \alpha) Ax + B_d d(s) \tag{5.11}
\]

or

\[
\dot{x} = Ax + B_d d(s) \tag{5.12}
\]

depending on the mode of the automaton. Equation (5.12) is just equation (5.11) with \(\alpha = 0\). Thus, if we can prove positivity and stability of the closed-loop system (5.11) for \(0 \leq \alpha < 1\), these properties will hold for the closed-loop system set forth in the automaton.

Following the proofs of positivity and stability, we will show that with the dynamics of \(\alpha\) specified in the
automaton, $\alpha$ does indeed satisfy $0 \leq \alpha < 1$.

5.6.1 State Positivity

We now address the positivity requirement, Design Requirement 2, and make the following claim.

**Claim 2 (State Positivity)** Assume $\alpha$ satisfies

$$0 \leq \alpha(t) < 1, \ \forall \ t \geq 0$$

and $d$ satisfies

$$d(t) \geq 0, \ \forall \ t \geq 0,$$

then closed-loop system (5.5) is an externally positive system, i.e. if $x(0) \in \mathbb{R}^n_+$ then $x(t) \in \mathbb{R}^n_+$ and $z(t) \in \mathbb{R}^1_+$ for all $t \geq 0$. Thus, Design Requirement 2 holds.

In the proof of Claim 2 we will make use of a positivity theorem for linear time-varying systems given in [28], which is stated here for convenience.

**Theorem 7** The linear time-varying system

$$\dot{x}(t) = A(t)x(t) + B(t)d(t), x(t_0) = x_0$$

$$z(t) = C(t)x(t) + D(t)d(t),$$

where $x(t) \in \mathbb{R}^n$, $d(t) \in \mathbb{R}^m$, $z(t) \in \mathbb{R}^p$, and $A(t), B(t), C(t), D(t)$, are real matrices of appropriate dimensions with continuous-time entries, is externally positive, that is for every $x_0 \in \mathbb{R}^n_+$ and $d(t) \in \mathbb{R}^m_+$ the state vector $x(t) \in \mathbb{R}^n_+$ and $z(t) \in \mathbb{R}^p_+$ for $t \geq t_0$ if and only if

(a) the off diagonal entries of $A(t)$, $a_{ij}(t), i \neq j$ satisfy

$$\int_{t_0}^{t} a_{ij}(\tau)d\tau \geq 0 \text{ for } i \neq j, i,j = 1,\ldots,n$$

(b) $B(t) \in \mathbb{R}^{n \times m}_+, C(t) \in \mathbb{R}^{p \times n}_+, D(t) \in \mathbb{R}^{p \times m}_+$, for $t \geq 0$.

We now proceed with the proof of Claim 2.
Proof 6 The system matrices of the closed-loop system (5.5) corresponding to the structure of system (5.13) are

\[
A(t) = (1 - \alpha(t)) A, \\
B(t) = B_d, \\
C(t) = (1 - \alpha(t)) w^T, \\
D(t) = 0.
\]

We will first address requirement (b). As defined, the components of \(B_d\) are nonnegative, thus \(B_d \in \mathbb{R}^{n \times 1}_+\). With \(0 \leq \alpha(t) < 1\) and the elements of \(w\) all nonnegative, \(C(t) \in \mathbb{R}^{1 \times n}_+\). Clearly \(D(t) = 0 \in \mathbb{R}_+\) and by assumption, \(d(t) \in \mathbb{R}^n_+\). Thus, condition (b) holds.

Next, we discuss the requirement that the system matrices be continuous. While in any one of the phases reaching, sliding and off, \(\alpha\) is continuous. From the reset relations given in (5.10) it can be seen that the only instant at which \(\alpha\) may be discontinuous is at a transition from reaching to off. Thus, there is only one possible discontinuity in \(\alpha\) and correspondingly in the system matrices \(A(t)\) and \(C(t)\), and the result of Theorem 7 can be applied before and after this discontinuity.

What remains to be shown is that condition (a) holds. Looking at the dynamics for an single section, given in equation (5.4), we can see that the off diagonal elements of the closed-loop \(A(t)\) matrix are

\[
a_{ij}(t) = \frac{\beta_{ji}}{\tau_j} (1 - \alpha(t))
\]

which is nonnegative, thus (a) holds and we can conclude that system (5.5) is externally positive.

5.6.2 Stability

We now focus on closed-loop stability, Design Requirement 1. We show that, as long as the function \(\alpha\) satisfies \(0 \leq \alpha(t) < 1\) for all \(t \geq 0\), the closed-loop system (5.5) exhibits global asymptotic stability. This is mainly due to the fact that, because of its compartmental structure, system (5.5) admits a fixed Lyapunov function, which is independent of the time-varying function \(\alpha\).

It should be noted, however, that traditional Lyapunov stability results, as given, e.g., in [29], cannot be readily applied in the present problem, since our Lyapunov function candidate will only be decreasing over
\( \mathbb{R}_n^+ \), which does not contain the origin of state space in its interior. We thus need to provide a variant of these results suited to positive systems. This is the content of the following theorem.

**Theorem 8** Let system (5.5) be a positive system (i.e., \( x(t) \in \mathbb{R}_n^+ \) for all \( t \)) and \( x = 0 \) be an equilibrium point of system (5.5). Assume there exists a continuously differentiable function \( V : \mathbb{R}_n^+ \to \mathbb{R} \) such that

\[
V(0) = 0 \text{ and } V(x) > 0, \quad \forall \ x \in \mathbb{R}_n^+ - \{0\} \\
\lim_{\|x\| \to \infty, \ x \in \mathbb{R}_n^+} V(x) = +\infty
\]

(5.14) (5.15)

\[
\dot{V}(x) < 0, \quad \forall \ x \in \mathbb{R}_n^+ - \{0\}
\]

(5.16)

Then, \( x = 0 \) is globally asymptotically stable.

The proof follows similar steps as the proof of Lyapunov’s stability theorem given in [29] with major differences arising from the fact that here we are dealing with positive systems and the proof that trajectories of the system are contained within a level set of \( V \) requires the use of this fact. The proof is provided in the Appendix.

We will show that the closed-loop system (5.5) is stable for any \( \alpha(t) \) satisfying \( 0 \leq \alpha(t) < 1 \). This property is stated in the following claim.

**Claim 3 (Stability)** Assume \( \alpha \) satisfies

\[
0 \leq \alpha(t) < 1, \quad \forall \ t \geq 0
\]

then closed-loop system (5.5) is globally asymptotically stable, that is Design Requirement 1 holds.

**Proof 7** First we note that since Claim 2 holds, i.e. the closed-loop system is positive, Theorem 8 is applicable. Now we can use Theorem 8 to show that the closed-loop system is stable. We define a function

\[
V(x) = l^T x
\]

where \( l \) is a column vector of length \( n \) and \( l_i \) is the length of the shortest path from section \( i \) to a sink of the network. Since we are dealing with positive systems, it is clear that \( V(x) \geq 0 \) for all \( t \geq 0 \). Also note that,
on $\mathbb{R}^n_+$, $V$ is radially unbounded, $V(0) = 0$, and $V(x(t)) > 0$ if $x(t) \neq 0$. Letting $\dot{V}$ designate the derivative of $V$ along trajectories of system (5.5), we have

$$
\dot{V} = l^T \dot{x}
$$

$$
= (1 - \alpha) \sum_{i=1}^{n} \left[ -l_i \frac{x_i}{\tau_i} + l_i \sum_{j \in \mathcal{O}_i} \beta_{ji} \frac{x_j}{\tau_j} \right]
$$

$$
= - (1 - \alpha) \sum_{i=1}^{n} l_i \frac{x_i}{\tau_i} - \sum_{i=1}^{n} \sum_{j \in \mathcal{O}_i} l_i \beta_{ji} \frac{x_j}{\tau_j}
$$

$$
= - (1 - \alpha) \sum_{i=1}^{n} \left[ \left( l_i - \sum_{j \in \mathcal{O}_i} l_j \beta_{ij} \right) \frac{x_i}{\tau_i} \right]
$$

Note that, given our assumptions on the structure of the network described in Section 5.2, $l_j \leq l_i$ for all $j \in \mathcal{O}_i$ and $l_j < l_i$ for at least one $j \in \mathcal{O}_i$. Combining this with the fact that $\sum_{j \in \mathcal{O}_i} \beta_{ij} = 1$, we can conclude that

$$
l_i - \sum_{j \in \mathcal{O}_i} l_j \beta_{ij} > 0
$$

and thus we have

$$
\dot{V} = -(1 - \alpha) \sum_{i=1}^{n} \left[ \left( l_i - \sum_{j \in \mathcal{O}_i} l_j \beta_{ij} \right) \frac{x_i}{\tau_i} \right] \leq 0, \ \forall \ x \in \mathbb{R}^n_+, \ \text{since} \ 0 \leq \alpha < 1. \ \text{Further}, \ \dot{V}(x(t)) < 0 \ \text{for all} \ x(t) > 0. \ \text{Thus}, \ \text{using Theorem 8 we conclude that the closed-loop system is globally asymptotically stable for} \ 0 \leq \alpha(t) < 1 \ \text{and} \ x(0) \in \mathbb{R}^n_+.
$$

5.7 Performance and Controller Properties

5.7.1 Output Matching

Focusing on Design Requirement 5a, we now show that with the dynamics of $\alpha$ given in equation (5.8) the output of the system $z$ will become close to the maximum allowable outflow $z_d$. That is, the quantity

$$
e(t) = z(t) - z_d(t)
$$

will eventually be within $\epsilon$ of zero. The particular choice for the dynamics of $\alpha$ defined in (5.8) is motivated by sliding mode control theory. The process of designing a sliding mode controller for system (5.5) involves
selecting a sliding mode manifold. In this case, the manifold is $e = 0$. Next, dynamics for $\alpha$ are chosen to drive the system to $e = 0$ and then slide along this manifold, which is precisely how we would like our system to behave.

In this section, we will prove that the resulting behavior of the closed-loop system with the dynamics of $\alpha$ defined in (5.8) is as desired, which is stated formally in the following claim.

**Claim 4** Assume

(a) the closed-loop system given by (5.5) is positive, that is $x(t) \geq 0$ for all $t \geq 0$,

(b) $\dot{\alpha}$ is given in (5.8),

(c) $\alpha$ satisfies $0 \leq \alpha < 1$,

(d) the dynamics of $z_d$ satisfy Constraint 1,

then the absolute value of the error between the closed-loop system output $z$ and the maximum allowable outflow $z_d$ defined by

$$|e| = |z - z_d|$$

will decrease until $|e| \leq \epsilon$.

**Proof 8** Showing that $|e|$ decreases until $|e| \leq \epsilon$ is equivalent to showing that $V$ defined by

$$V = \frac{1}{2} e^2$$

decreases while $|e| > \epsilon$.

Define $\rho(x, t)$ as

$$\rho(x, t) = \frac{1}{\tau_{\min}} \left( 1 + \sum_{j \in \mathcal{S}_p} \frac{x_j}{\tau_j} \right) + \frac{G}{w^T x}.$$ 

With this definition of $\rho(x, t)$, we can rewrite $\eta(x, t)$ as

$$\eta(x, t) = \eta_0 + \frac{c_0}{w^T x} + \rho(x, t). \quad (5.17)$$
Computing $\dot{V}$ along trajectories of the system, we have

$$
\dot{V} = e \left[ (1 - \alpha) w^T \dot{x} - g(t) \right] - e \dot{\alpha} w^T x \\
= e \left[ (1 - \alpha) w^T \dot{x} - g(t) \right] - e \left( \eta_0 + \frac{c_0}{w^T x} + \rho(x, t) \right) \text{sat} \left( \frac{\epsilon}{\epsilon} \right) w^T x \\
= e \left\{ (1 - \alpha) w^T \dot{x} - g(t) - w^T x \rho(x, t) \text{sat} \left( \frac{\epsilon}{\epsilon} \right) \right\} - e \left( \eta_0 + \frac{c_0}{w^T x} \right) \text{sat} \left( \frac{\epsilon}{\epsilon} \right) w^T x.
$$

For $|e| > \epsilon$, $\text{sat} \left( \frac{\epsilon}{\epsilon} \right) e$ is replaced by $|e|$ and $\dot{V}$ can be written as

$$
\dot{V} = e \left[ (1 - \alpha) w^T \dot{x} - g(t) \right] - w^T x \rho(x, t) |e| - \left( \eta_0 + \frac{c_0}{w^T x} \right) |e| w^T x.
$$

Given that $x(t) \geq 0$, if

$$
\rho (x, t) \geq \left| \frac{(1 - \alpha) w^T \dot{x} - g(t)}{w^T x} \right| \tag{5.18}
$$

then $\dot{V}$ has the following bound

$$
\dot{V} \leq -\eta_0 (w^T x) |e| - c_0 |e| \leq -c_0 |e| < 0, \quad (5.19)
$$

recalling that $\eta_0 \geq 0$ and $c_0 > 0$.

We will show that (5.18) is satisfied by finding an upper bound on the right hand side of this inequality and then showing that $\rho(x, t)$ is in fact equal to this upper bound. We can split up the absolute value expression on the right hand side of (5.18) and find upper bounds on the terms

$$
\left| \frac{(1 - \alpha) w^T \dot{x}}{w^T x} \right| \tag{5.20}
$$

and

$$
\left| \frac{g(t)}{w^T x} \right| \tag{5.21}
$$

separately.

We will begin by finding an upper bound for (5.20), recalling that $\alpha$ is assumed to satisfy $0 \leq \alpha \leq 1$. With this bound on $\alpha$ we have

$$
\left| \frac{(1 - \alpha) w^T \dot{x}}{w^T x} \right| \leq \frac{w^T \dot{x}}{w^T x}.
$$
and we proceed by finding a bound on $|w^T \dot{x}|$. Since $\gamma_i = 0$ for all $i \in S_F$ we find

$$w^T \dot{x} = \sum_{i \in S_F} \frac{\dot{x}_i}{\tau_i}$$

$$= \sum_{i \in S_F} \left\{ \frac{1}{\tau_i} \left[ \frac{1 - \alpha}{\tau_i} x_i + \sum_{j \in S_P} \frac{1 - \alpha}{\tau_j} \beta_{ji} x_j \right] \right\}$$

$$= (1 - \alpha) \left\{ - \sum_{i \in S_F} \frac{x_i}{\tau_i^2} + \sum_{i \in S_F} \frac{1}{\tau_i} \sum_{j \in S_P} \frac{\beta_{ji} x_j}{\tau_j} \right\}.$$

Using the fact that $0 \leq \alpha \leq 1$ and $x_i \geq 0\; \forall \; i$, we have the following bound on $|w^T \dot{x}|$

$$|w^T \dot{x}| = (1 - \alpha) \left| - \sum_{i \in S_F} \frac{x_i}{\tau_i^2} + \sum_{i \in S_F} \frac{1}{\tau_i} \sum_{j \in S_P} \frac{\beta_{ji} x_j}{\tau_j} \right|$$

$$\leq \left| - \sum_{i \in S_F} \frac{x_i}{\tau_i^2} + \sum_{i \in S_F} \frac{1}{\tau_i} \sum_{j \in S_P} \frac{\beta_{ji} x_j}{\tau_j} \right|$$

$$\leq \sum_{i \in S_F} \frac{x_i}{\tau_i^2} + \sum_{i \in S_F} \frac{1}{\tau_i} \sum_{j \in S_P} \frac{\beta_{ji} x_j}{\tau_j}.$$

By defining

$$\tau_{min} = \min_{i \in S_F} \{ \tau_i \}$$

we have

$$\sum_{i \in S_F} \frac{x_i}{\tau_i^2} \leq \frac{1}{\tau_{min}} \sum_{i \in S_F} \frac{x_i}{\tau_i}$$

$$= \frac{1}{\tau_{min}} w^T x$$

(5.23)

and

$$\sum_{i \in S_F} \frac{1}{\tau_i} \sum_{j \in S_P} \frac{\beta_{ji} x_j}{\tau_j} \leq \frac{1}{\tau_{min}} \sum_{i \in S_F} \sum_{j \in S_P} \frac{\beta_{ji} x_j}{\tau_j}.$$

(5.24)

Notice that

$$\sum_{i \in S_F} \sum_{j \in S_P} \frac{\beta_{ji} x_j}{\tau_j} = \sum_{j \in S_P} \frac{x_j}{\tau_j}.$$  

(5.25)

Due to the interconnection structure of the system, all flights entering sections in $S_F$ must come from sections in $S_P$, and all flights leaving a section in $S_F$ must enter a section in $S_F$. Therefore, summing over all sections leading into all sections in $S_F$ we have the sum of all flights leaving sections in $S_P$. 

95
Using (5.23), (5.24), and (5.25) with (5.22) we arrive at the following bound on $|w^T \dot{x}|$

$$|w^T \dot{x}| \leq \frac{1}{\tau_{\text{min}}} w^T x + \frac{1}{\tau_{\text{min}}} \sum_{j \in \mathcal{S}_p} \frac{x_j}{\tau_j}. \quad (5.26)$$

Finally, we arrive at the desired bound

$$\left| \frac{(1 - \alpha) w^T \dot{x}}{w^T x} \right| \leq \frac{1}{\tau_{\text{min}}} \left( 1 + \sum_{j \in \mathcal{S}_p} \frac{x_j}{w^T x} \right).$$

An upper bound for term (5.21) follows directly from Constraint 1 giving

$$\left| \frac{g(t)}{w^T x} \right| \leq \frac{G}{w^T x}.$$ 

Thus

$$\left| \frac{(1 - \alpha) w^T \dot{x} - g(t)}{w^T x} \right| \leq \frac{1}{\tau_{\text{min}}} \left( 1 + \sum_{j \in \mathcal{S}_p} \frac{x_j}{w^T x} \right) + \frac{G}{w^T x}. \quad (5.27)$$

The term $\rho(x, t)$ is equal to the right hand side of (5.27), thus inequality (5.18).

We can conclude that $\dot{V}$ is strictly negative when $|e| > \epsilon$, and thus $V = \frac{1}{2} e^2$ (and consequently $|e|$) decreases when $|e| > \epsilon$ and the claim holds.

Note that because $\alpha$ depends on the state of all sections in level $\mathcal{F}$, the proposed control law is not fully decentralized, in spite of the structure (5.2): the control input applied to each section not only depends on its local state, but also on the global outflow of the system and the maximum allowable outflow.

### 5.7.2 Constraints on Controller

The assumptions of Claims 2, 3 and 4 all include that $0 \leq \alpha < 1$. However, this control law does not ensure that $\alpha$ remain positive. If the controlled outflow is less than the maximum allowable outflow, i.e.

$$(1 - \alpha(t)) w^T x(t) < z_d(t)$$

over some time interval (i.e., if $e < 0$ causing $\alpha$ to decrease over an interval), it is possible for $\alpha$ to be driven negative. This means that the input $u_i(t) = \alpha(t) \frac{x(t)}{\tau_i}$ for all $i \in [n]$ may be negative for some interval of time.

In order to ensure that the control input is always physically meaningful, as captured by Design Requirement 3, we can simply set $\alpha$ to zero whenever relation (5.8) drives $\alpha$ negative. This motivates the transition from
the reaching and sliding modes of the automaton, under which \( \dot{\alpha} \) is defined by equation (5.8), to the off mode, under which \( \dot{\alpha} = 0 \). As we have proved earlier, setting \( \alpha \) to zero in such a way does not affect closed-loop stability.

This control law does ensure that \( \alpha \) remain bounded above by one. In order to show this, we need that \( z_d(t) \geq \bar{z}_d \) and we must show that \( \eta(x, t) \) has a finite constant upper bound. The fact that \( \eta(x, t) \) can be shown to have a finite upper bound whenever \( \alpha \) varies according to (5.8) is a result of the transitions defined in the automaton. Since \( \eta(x, t) \) is undefined when \( w^T x(t) = 0 \), we propose setting \( \alpha(t) = 0 \) with \( \dot{\alpha}(t) = 0 \) when \( w^T x(t) \) falls below \( \bar{z}_d - \epsilon \). This is achieved by the guard which triggers a transition from reaching to off when \( w^T x < \bar{z}_d - \epsilon \). Note that there is no need to explicitly define a transition from sliding to off to ensure that \( w^T x \geq \bar{z}_d - \epsilon \). While in the sliding phase,

\[
| (1 - \alpha) w^T x - z_d | \leq \epsilon.
\]

With \( z_d \geq \bar{z}_d \) and \( 0 \leq \alpha < 1 \), this relation implies that

\[
w^T x \geq \bar{z}_d - \epsilon.
\]

The guard from sliding to off triggers a transition when \( \alpha = 0 \) and

\[
e = w^T x - z_d < 0.
\]

Thus, while in the sliding phase, \( w^T x \geq \bar{z}_d - \epsilon \), and no explicit transition to off need be defined to ensure this. Through the definition of the automaton, it is guaranteed that whenever \( \alpha \) varies according to (5.8), \( w^T x \) remains bounded away from zero and \( \eta \) is well defined.

With the control strategy defined by the automaton and \( \eta(x, t) \) defined in equation (5.7) we can make the following claim

**Claim 5 (Constrained Control)** Assume that

(a) the closed-loop system is given by (5.5) with initial conditions \( x(0) \in \mathbb{R}^n_+ \),

(b) the control strategy given in automaton \( H \) is used to generate the control input \( \alpha \) with initial condition \( 0 \leq \alpha(0) < 1 \),
(c) the inflow to the system $d$ is finite and nonnegative, i.e. $d(t) \geq 0$ for all $t \geq 0$, and there exists a finite $t^* \geq 0$ such that $d(t) = 0$ for all $t \geq t^*$,

(d) the dynamics of $z_d$ satisfy Constraint 1,

then $\alpha$ satisfies

$$0 \leq \alpha(t) < 1, \forall t \geq 0.$$  

and it follows that Design Requirement 3 holds.

**Proof 9** The switching control law defined by automaton $H$ sets $\alpha$ to zero with $\dot{\alpha} = 0$ whenever $\alpha$ may be driven negative. Thus we can see that $\alpha(t) \geq 0$ for all $t \geq 0$.

In order to show that $\alpha(t) < 1$, we must first show that $|\dot{\alpha}(t)|$ has a finite upper bound, which is equivalent to showing that $\eta(x,t)$ has a finite upper bound when $w^T x(t) \geq \bar{z}_d - \epsilon$, recalling that $\eta(x,t) \geq 0$. Recall $\eta(x,t)$ defined in (5.7) as

$$\eta(x,t) = \eta_0 + \frac{1}{\tau_{\text{min}}} + \frac{c_0}{\tau_{\text{min}}} \sum_{j \in S_P} \frac{x_j}{\tau_j} + G,$$

where $\eta_0, c_0, \tau_{\text{min}}$ and $G$ are positive constants. In order to find an upper bound for

$$\sum_{j \in S_P} \frac{x_j}{\tau_j} w^T x,$$

let

$$\tau'_\text{min} = \min_{i \in S_P} \{\tau_i\},$$

and define

$$D = \int_0^\infty d(t) \, dt$$

which, given condition (c), is nonnegative and finite. Then

$$\sum_{j \in S_P} \frac{x_j}{\tau_j} \leq \frac{D}{\tau'_\text{min}}$$

and

$$\eta(x,t) \leq \eta_0 + \frac{1}{\tau_{\text{min}}} + \frac{c_0}{\tau_{\text{min}}\tau'_\text{min}} + G.$$  \hspace{1cm} (5.28)

Thus, $\eta(x,t)$ has a finite upper bound and $|\dot{\alpha}(t)|$ has a finite upper bound.

Unlike $x(t)$, $\alpha(t)$ is not necessarily continuous. During a transition from reaching to off, the value of $\alpha$ is
reset to zero. However, since $\alpha$ is fixed at zero in the off phase, the only phases in which $\alpha$ may possibly become greater than one are reaching and sliding. While the automaton is in either the reaching or sliding phases, $\alpha$ is allowed to evolve continuously according to the given dynamics.

We will consider the sliding phase first. The only points at which $\dot{\alpha}$ does not exist is at transitions into or out of this phase. When transitioning into or out of the sliding phase, $\alpha = 0$, thus at any point at which $\alpha$ may equal one during the sliding phase, $\alpha$ is both continuous and differentiable.

At any time $t$ such that $\alpha(t) = 1$,

$$e(t) = (1 - \alpha(t)) w^T x(t) - z_d(t)$$

$$= -z_d(t)$$

$$\leq -\bar{z}_d$$

$$< 0$$

where the inequalities follow from Constraint 1. Thus, with $\dot{\alpha}$ defined in (5.8) we see that $\dot{\alpha}(t) < 0$ when $\alpha(t) = 1$.

With the assumption that $0 \leq \alpha(0) < 1$ and the transitions defined by $H$, at any transition time, $t_1$ (including $t_1 = 0$), $\alpha(t_1) < 1$. Assume that there exists $t_2 > t_1$ such that

$$\alpha(t_2) > 1.$$

and let $t_0$ be the first point in $(t_1, t_2)$ such that $\alpha(t_0) = 1$. Note that since $\dot{\alpha}$ is bounded, the point at which $\alpha = 1$ cannot be an accumulation point, thus this time $t_0$ is well defined.

Since $\alpha$ is differentiable at $t_0$, the following limit exists and

$$\lim_{t \to t_0} \frac{\alpha(t) - \alpha(t_0)}{t - t_0} = -\gamma < 0.$$

Given that this limit exists we know that, for all $\epsilon > 0$ there exists $\delta > 0$ such that

$$|t - t_0| < \delta \Rightarrow \left| \frac{\alpha(t) - \alpha(t_0)}{t - t_0} + \gamma \right| < \epsilon.$$
Equivalently,

\[ |t - t_0| < \delta \Rightarrow -\gamma - \epsilon < \frac{\alpha(t) - \alpha(t_0)}{t - t_0} < -\gamma + \epsilon. \]

Since \( \gamma > 0 \) we can choose \( \epsilon = \frac{\gamma}{2} > 0 \) and we have

\[ \frac{-3\gamma}{2} < \frac{\alpha(t) - \alpha(t_0)}{t - t_0} < \frac{-\gamma}{2}. \]

For \( t = t_0 - \frac{\delta}{2} < t_0 \), we have

\[ \left( \frac{3\gamma}{2} \right) \left( \frac{\delta}{2} \right) > \alpha \left( t_0 - \frac{\delta}{2} \right) - \alpha(t_0) > \left( \frac{\gamma}{2} \right) \left( \frac{\delta}{2} \right) > 0 \]

and it follows that \( \alpha(t_0 - \frac{\delta}{2}) > \alpha(t_0) = 1 \). Since \( \alpha \) is continuous, there must exist a \( t' \) in \( (t_1, t_0 - \frac{\delta}{2}) \) such that \( \alpha(t') = 1 \), which is a contradiction to the assumption that \( t_0 \) is the first point in \( (t_1, t_2) \) such that \( \alpha(t_0) = 1 \).

Thus, the point \( t_2 \) such that \( \alpha(t_2) > 1 \) does not exist and we can conclude that \( \alpha(t) \leq 1 \) for all \( t \geq t_1 \).

If \( \alpha(t^*) = 1 \) for any \( t^* \geq t_1 \), it must be a local maximum of \( \alpha \). However, since \( \alpha \) is continuous and differentiable near any point such that \( \alpha(t^*) = 1 \), a local maximum must be a critical point, at which \( \dot{\alpha}(t^*) \) should be equal to zero. As shown above, \( \alpha(t^*) < 0 \), thus \( \alpha(t^*) \) cannot be a local maximum. We can conclude that \( \alpha(t) < 1 \) for all \( t \geq t_1 \) when in the sliding phase.

Now we will consider the reaching phase. In this phase, \( \alpha \) will be set to zero if \( w^T x(t) < \bar{z}_d - \epsilon \), which may result in a discontinuity in \( \alpha \) and \( \dot{\alpha} \) will not exist at this point. However, since the dynamics of \( x \) and \( \alpha \) are the same in the reaching and sliding phase, the only difference between these two phases, is this possible discontinuity in \( \alpha \) which sets \( \alpha \) to zero. Thus, the same results hold, i.e. \( \alpha < 1 \) while the automaton is in the reaching phase.

We can conclude that \( 0 \leq \alpha(t) < 1 \) for all \( t \geq 0 \).

5.8 Properties of the Automaton

Now that we have shown the positivity and stability of the closed-loop system (5.5) in Claims 2 and 3, described the output behavior under the proposed control law in Claim 4 and the shown that the constraints on the control input \( \alpha \) hold in Claim 5, we can show that the properties of the automaton discussed in Section 5.5.2 hold. First we address the domain preserving property, which can be stated formally in the following claim.
Claim 6  \( H \) is domain preserving, i.e. the set of reachable states is a subset of \( Q \times \text{Dom} \).

Proof 10  The guards defined in (5.9) follow from the need to keep \( \alpha(t) \geq 0 \) for all \( t \geq 0 \) and \( w^T x(t) \geq z_d - \epsilon \) when \( \eta(x, t) \) is to be calculated. What may not be clear is why transitions from a state in either \( U_{\text{sliding}} \) or \( U_{\text{off}} \) to another in \( U_{\text{reaching}} \) are not possible. While the state of the automaton is in \( U_{\text{sliding}} \), \( \alpha \) evolves according to (5.8). As shown in Section 5.7.1, using this control law will ensure that \( |e| \leq \epsilon \). If \( \alpha \) happens to be driven negative, the controller is turned off and \( \alpha \) is fixed at zero. This corresponds to a transition from a state in \( U_{\text{sliding}} \) to a state in \( U_{\text{off}} \), thus, it is not possible for the automaton to transition from \( U_{\text{sliding}} \) to \( U_{\text{reaching}} \).

While the state of the automaton is in \( U_{\text{off}} \), \( e(t) < \epsilon \) and \( \alpha(t) = 0 \). In order for the state to enter \( U_{\text{reaching}} \), \( e(t) \) must become greater than \( \epsilon \). Since both the maximum outflow capacity \( z_d \) and the outflow of the system \( z \) are continuous functions of time, the outflow error \( e \) is also a continuous function of time. The error must pass through \( \epsilon \) before becoming greater than \( \epsilon \). However, when \( (q, x) \) is in \( U_{\text{off}} \), the guard between off and sliding will be triggered when \( e = \epsilon \) which causes a transition from \( U_{\text{off}} \) to \( U_{\text{sliding}} \). Thus, a transition from \( U_{\text{off}} \) to \( U_{\text{reaching}} \) is not possible.

Notice that \( \text{Dom}(\text{reaching}) \) includes values of \( w^T x \) which are strictly less than \( \bar{z}_d - \epsilon \). With the convention that at \( t = 0 \) the guards are evaluated and any transitions occur before continuous evolution begins. When a transition from reaching to off occurs, \( \alpha \) may not necessarily be equal to zero, thus the reset relation \( R((\text{reaching}, \text{off}), x) \) explicitly sets \( \alpha \) to zero after this transition. Also note that with \( w^T x(t) < \bar{z}_d - \epsilon \), \( e(t) \) will be strictly less than zero after the transition since

\[
e(t) = w^T x(t) - z_d \leq (\bar{z}_d - \epsilon) - z_d \leq -\epsilon < 0
\]

and \( x \) is in \( \text{Dom}(\text{off}) \) after the transition.

The guard \( G(\text{sliding}, \text{off}) \) does not explicitly contain a condition to ensure that \( w^T x \geq \bar{z}_d - \epsilon \). This is because, as long as \( x \in \text{Dom}(\text{sliding}) \)

\[
(1 - \alpha)w^T x - z_d \geq -\epsilon
\]

and it follows that \( w^T x \geq \bar{z}_d - \epsilon \). Therefore, no additional guard is needed to ensure that a transition from sliding to off occurs when \( w^T x < \bar{z}_d - \epsilon \).
As discussed previously, \( \text{Dom}(\text{reaching}) \cup \text{Dom}(\text{sliding}) \cup \text{Dom}(\text{off}) = X \). Therefore, for any initial value of the continuous state, \( x_0 \in \{ x \in X : s = 0 \} \), there exists \( q_0 \in \{ \text{reaching}, \text{sliding}, \text{off} \} \) such that \( x_0 \in \text{Dom}(q_0) \).

For any \( d(\cdot) \) satisfying \( d(t) \geq 0 \) for all \( t \geq 0 \) from Claim 2 we can conclude that \( x(t) \in \mathbb{R}_+^n \) for all \( t \geq 0 \). From Constraint 1 we have \( z(t) \in \mathbb{R}_+^n \) for all \( t \geq 0 \). Claim 5 together with the reset relations given above yield \( 0 \leq \alpha < 1 \). And clearly, \( s(0) = 0 \) and \( \dot{s} = 1 \) results in \( s(t) \geq 0 \) for all \( t \geq 0 \). Thus, \( x \in \mathbb{R}_+^{n+3} \) for all \( t \geq 0 \). Combining this result with the switching rules given in (5.9) we can conclude that for any \( (q_0, x_0) \in \text{Init} \), the state of the automaton will remain in \( Q \times X \). Thus, \( H \) is domain preserving.

As discussed in Section 5.5.2, having introduced the notion of \textit{Satisfactory Output} in Section 5.3, the proof that the control law defined by the automaton is satisfactory, that is, satisfies Design Requirement 5, amounts to showing that \( U_{\text{reaching}} \) is transient and \( A = U_{\text{sliding}} \cup U_{\text{off}} \) is absorbing. These properties are stated in the following two claims.

\textbf{Claim 7} The set \( U_{\text{reaching}} \) is transient, i.e. there exists \( T > 0 \) such that \( x(t) \notin U_{\text{reaching}} \) for all \( t \geq T \).

\textbf{Proof 11} We begin by ignoring the constraints on the controller discussed in Section 5.7.2, i.e. assuming the dynamics of \( \alpha \) is given by (5.8) and \( \alpha \) is allowed to take on negative values, and \( \eta \) is always well defined. The natural performance measure for the closed-loop system is the time required for the system to reach the boundary of \( |e| \leq \epsilon \), which, as mentioned in Section 5.3, we will denote as \( T_\epsilon \). Define

\[
W = \sqrt{2V} = |e|
\]

and notice that

\[
D^+W \leq -c_0
\]

where \( D^+ \) indicates the upper right-hand derivative. Now we can use the comparison lemma (see \cite{29}) along with the upper bound on \( \dot{V} \) given in (5.19) to state

\[
W(e(t)) \leq W(e(0)) - c_0 t. \tag{5.29}
\]

Use (5.29) to find a value of \( T_\epsilon \) which satisfies

\[
|e(T_\epsilon)| \leq \epsilon. \tag{5.30}
\]

102
From

\[ |e(0)| - c_0 T \leq \epsilon \]

we find that the smallest value of \( T \) which satisfies (5.30) is

\[ T = \frac{|e(0)| - \epsilon}{c_0} \]

For all \( t \geq T \), we know that \( |e(t)| \leq \epsilon \).

However, if \( \alpha \) is driven to zero with \( c < 0 \) or if \( w^T x \) falls below \( \bar{z}_d - \epsilon \) before \( |e| \leq \epsilon \), the state of automaton \( H \) will switch from \( U_{\text{reaching}} \) to \( U_{\text{off}} \). Thus, the state of \( H \) may exit \( U_{\text{reaching}} \) prior to the time at which \( |e| \leq \epsilon \).

We see that the automaton transitions out of \( U_{\text{reaching}} \) in at most \( T \) time units by entering either \( U_{\text{sliding}} \) or \( U_{\text{off}} \). As described in the proof of Claim 6, transitions from either \( U_{\text{sliding}} \) or \( U_{\text{off}} \) to \( U_{\text{reaching}} \) are not possible. Thus, \( H \) is not in \( U_{\text{reaching}} \) for all \( t \geq T \), that is \( U_{\text{reaching}} \) is transient.

Note that if \( \alpha \) is found to be positive for all \( t > 0 \) then it can be said that \( |e| \leq \epsilon \) for all \( t \geq T \) and in this way \( T \) can be considered a performance measure for the system.

Now we state the claim that \( U_{\text{sliding}} \cup U_{\text{off}} \) is absorbing.

**Claim 8 (Satisfactory Output)** The set \( A = U_{\text{sliding}} \cup U_{\text{off}} \) is absorbing, i.e. if \( (q(t^*), x(t^*)) \in A \) for some \( t^* \geq 0 \), then \( (q(t), x(t)) \in A \) for all \( t \geq t^* \) and Design Requirement 5 holds.

**Proof 12** Follows directly from Claims 6 and 7.

We are now in a position to prove that the automaton is well behaved in that only a finite number of discrete state transitions can occur in finite time, which is captured in the following claim.

**Claim 9** \( H \) is non-Zeno, i.e. the automaton cannot make an infinite number of discrete state transitions in finite time.

**Proof 13** Proving that \( H \) is non-Zeno is equivalent to showing that between switching from \( U_{\text{sliding}} \) to \( U_{\text{off}} \) and vice versa, the automaton spends a non-zero length of time in each of these modes. Let us first examine
a transition from off to sliding. Such a transition occurs when \( e = \epsilon \). Once the automaton is in the sliding mode, it cannot transition to off until \( e < 0 \). With \( e \) defined by

\[
e = (1 - \alpha)w^T x - z_d(t)
\]

we see that \( e \) is continuous since \( x, \alpha \) and \( z_d \) are each continuous. Thus, if \( \dot{e} \) is bounded, it takes a non-zero length of time for \( e \) to decrease from \( \epsilon \) to zero. The time derivative of \( e \),

\[
\dot{e} = (1 - \alpha)w^T \dot{x} - \dot{\alpha}w^T x - g,
\]

is bounded since \( \alpha \) is bounded, \( x \) is bounded by \( D = \int_0^\infty d(t)dt \), \( g \) must be bounded by Constraint 1, \( w^T \dot{x} \) has a bound given in (5.26) (which depends on \( x \) which is also bounded) and \( \dot{\alpha} \) has a bound given in (5.28). We can conclude that the automaton will remain in \( U_{\text{sliding}} \) for a non-zero length of time. When the automaton transitions from sliding to off, \( e < 0 \) and a transition back to sliding will not occur until \( e = \epsilon \). By a similar argument as that above, we see that the automaton will remain in \( U_{\text{sliding}} \) for a non-zero length of time. We can conclude that \( H \) is non-Zeno.

5.8.1 Summary of Properties of the Automaton

In Section 5.5.1 we defined an autonomous hybrid automaton to describe the proposed control strategy to satisfy the design requirements set forth in Section 5.3. We defined three domains of the continuous state, each of which is a subset of the allowable range of values of the continuous state. The values that the state vector \( x \) and control input \( \alpha \) (which make up part of the continuous state of the automaton) can take on follow from Design Requirements 2 and 3. Additionally, these domains were used to distinguish between regions in which the control input dynamics are “on” and “off” and to specify acceptable regions of operations corresponding to Design Requirement 5. We then showed that any trajectory of the automaton starting from a valid initial condition remains in this established domain, thus Design Requirements 2 and 3 are satisfied. We showed that trajectories of the automaton are confined to the acceptable regions of \( Q \times X \), namely \( U_{\text{sliding}} \) and \( U_{\text{off}} \), after finite time. We also provided an upper bound on the time at which the state of the automaton will belong to \( U_{\text{sliding}} \cup U_{\text{off}} \). The state vector \( x \) belonging to Dom(sliding) and Dom(off) corresponds to Design Requirements 5a and 5b being satisfied, respectively.
Figure 5.3: Airspace between airports \( A \) and \( B \) divided into sections of 1-D flow, arrows indicate direction of flow.

### 5.9 Application Example

In this section, we focus on the network presented in Figure 5.3 representing the flow of aircraft between two airports and apply the sliding mode controller developed in Section 5.7.1. Compartments correspond to sections of airspace and the state and dynamics of each section represent aggregate quantities, see [7] or [8] for more details. This kind of Eulerian model has been proposed recently in [35]. In this network, all aircraft take off from airport \( A \) and land at airport \( B \). The state-space representation of this model is given by

\[
\dot{x} = \begin{bmatrix}
-\frac{1}{\tau_1} & 0 & 0 & 0 \\
0 & -\frac{1}{\tau_2} & 0 & 0 \\
\frac{1}{\tau_1} & \beta_{23} & -\frac{1}{\tau_3} & 0 \\
0 & \beta_{24} & 0 & -\frac{1}{\tau_4}
\end{bmatrix} x + \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
-1 & -\beta_{23} & 1 & 0 \\
0 & -\beta_{24} & 0 & 1
\end{bmatrix} u + \begin{bmatrix}
\gamma_1 \\
\gamma_2 \\
0 \\
0
\end{bmatrix} d
\]

(5.31)

\[
z = \begin{bmatrix}
0 & 0 & 1 & \frac{1}{\tau_4}
\end{bmatrix} x + \begin{bmatrix}
0 & 0 & -1 & -1
\end{bmatrix} u,
\]

where \( x = [x_1, \ldots, x_4]^T \) is a vector of state variables and \( u = [u_1, \ldots, u_4]^T \) is the control input.

The parameter values used in the simulation are

\[
\tau_1 = 0.625, \tau_2 = 0.938, \tau_3 = 0.208, \tau_4 = 0.250, \\
\beta_{23} = 0.7, \beta_{24} = 0.3, \gamma_1 = 0.5, \gamma_2 = 0.5.
\]

The traversal times \( \tau_i \) have units of hours and were chosen to be comparable to the parameter values used in [35]. The maximum allowable outflow is generated by a first-order system with time-constant 0.5. The system and model values used here are the same as those used in the example in [8]. The constant values in
were chosen to be $\eta_0 = 0$, $c_0 = 10$. We chose $\epsilon = 1$, indicating that an acceptable error in the landing rate is 1 aircraft per hour.

The maximum allowable outflow $z_d$ is generated by a target system defined by

$$
\dot{\xi} = \tilde{A}\xi + \tilde{B}\tilde{d}
$$

$$
z_d = \tilde{C}\xi,
$$

with $\xi(t) \in \mathbb{R}$ for all $t \geq 0$.

In order to ensure that the maximum allowable outflow and its derivative satisfy Constraint 1 and to find an upper bound $G$ for the derivative $\dot{z}_d$ of the output of system (5.32) defined above, we make the following assumptions: $\xi_0 = \tilde{z}_d$, $\tilde{d}(t)$ is a step or square wave function which satisfies $\tilde{z}_d \leq \tilde{d}(t) \leq \tilde{d}_0$ for some $\tilde{d}_0 \geq \tilde{z}_d$ and (5.32) is a stable first order positive linear system with $\tilde{A} < 0$, $\tilde{B} > 0$ and $\tilde{C} \geq 0$. We can then solve for $\xi(t)$ and differentiate to find the maximum of $\dot{\xi}(t)$ and thus arrive at the following bound

$$
|\tilde{C}\dot{\xi}| \leq \tilde{B}\tilde{d}_0,
$$

and thus, $G = \tilde{B}\tilde{d}_0$.

For the first simulation presented, the target system parameter values used are

$$
\tilde{A} = -0.5, \quad \tilde{B} = 1, \quad \tilde{C} = 0.5, \quad \tilde{z}_d = 2, \quad \tilde{d}_0 = 50,
$$

noting that the chosen value for $\epsilon$, namely $\epsilon = 1$, is less than $\tilde{z}_d$. The initial conditions for the system, target system and controller are $x(0) = [1, 1, 1, 1]^T$, $\xi(0) = 2\tilde{z}_d$ and $\alpha(0) = 0$, respectively. With these initial conditions, the initial error is approximately 6.8 aircraft per hour. The input $d(t)$ to system (5.31) is a square wave function oscillating between 0 and 50 aircraft per hour, with a period of 24 hours and pulse width of 50%. With this input, a total of 600 aircraft enter the system over a 24 hour period. The input $\tilde{d}(t)$ to the target system (5.32) is a square wave function oscillating between $\tilde{z}_d$ and 50 aircraft per hour, with a period of 24 hours and pulse width of 50%. The maximum allowable landing rate capacity $z_d$, uncontrolled system output, and controlled landing rate are plotted in Figure 5.4.

A plot of control gain $\alpha$ over the course of the simulation is presented in Figure 5.4. Under the proposed control, the landing rate error is guaranteed to be within $\epsilon$ of zero, or negative with $\alpha = 0$, by $T_e = 0.58$ hours. In simulation, the error is actually within $\epsilon$ of zero before 0.04 hours (2.4 minutes). Over the course
of the 24 hours of the simulation, the uncontrolled system accumulates a total error of 120 aircraft, while the controlled system accumulates an error of only 1 aircraft while the controller is not off. At about 16.3 hours, the outflow of the system is unable to match the allowable outflow, even with flights traveling through the network at the maximum allowable rate. At this point, the controller is turned off (i.e. \( \alpha = 0, \dot{\alpha} = 0 \)). After this point, an additional error of approximately 19.24 aircraft is accumulated.

A second simulation was performed to illustrate the transition of the autonomous hybrid automaton from the sliding regime to off and back to sliding. The same system (5.31), system generating the maximum allowable outflow (5.32), parameters, and initial conditions used in the first simulation were used here. The only difference between the two simulations is that the input \( d \) to system (5.31) for the second simulation is a departure rate of 40 aircraft per hour for the first 15 hours, dropping to zero after 15 hours. The input \( \tilde{d} \) to the system generating the maximum allowable outflow (5.32) is 50 aircraft per hour for the first 12 hours, dropping to \( \bar{z}_d \) after 12 hours. The resulting uncontrolled landing rate and maximum allowable landing rate are shown in Figure 5.5.

Similar results to those given for the first simulation are shown in Figure 5.5. Within 0.04 hours, the controlled landing rate is within \( \epsilon \) of the maximum allowable landing rate. At approximately 4.6 hours, the controlled landing rate falls to more then \( \epsilon \) below the maximum allowable landing rate. At this point \( \alpha = 0 \) and \( \epsilon < 0 \), so the automaton transitions to the off state, fixing the control parameter \( \alpha \) at zero. At approximately 12.4 hours, the controlled landing rate is again within \( \epsilon \) of the maximum allowable landing rate. At this point, the automaton transitions from the off state to the sliding state and \( \alpha \) moves off of \( \alpha = 0 \).

During first sliding phase the flow of aircraft through the network is mitigated so as to match the maximum
allowable landing rate. Between 4.6 hours and 12.4 hours, the maximum allowable landing rate is too high for the system to match. During this time frame, aircraft are allowed to flow through the system with no controlled delay. At approximately 12.4 hours, the maximum allowable outflow has dropped below the natural outflow rate of the system. At this point, the controller is turned on and $\alpha$ is again adjusted to meter the outflow rate of the system so as to match the maximum outflow rate with an error of at most $\epsilon$. 

Figure 5.5: Maximum allowable landing rate capacity, uncontrolled landing rate, and controlled landing rate (left) and control parameter $\alpha$ as a function of time (right).
Chapter 6

Application of \textit{SMC} Method in Flight-by-Flight Simulation

6.1 Model Assumptions and Relevance to Air Traffic Flow Management

In the development of continuous time aggregate models for air traffic flow management, several assumptions are made, including infinite divisibility of the state and the ability to implement a continuously varying control parameter. In an ATFM application, the state represents the number of aircraft in each section, which is a discrete quantity. However, these models have been shown to accurately describe the flow of aircraft in dense traffic \cite{45, 44}. Control input is specified at the aggregate level. This aggregate control must be translated into flight-by-flight commands for each individual aircraft involved in the problem. The implementation of an aggregate model and control method for ATFM problems requires the use of some disaggregation method to translate the aggregate control input to a flight-by-flight control input. As suggested in \cite{43}, rounding heuristics can be used to generate integer values of control inputs when needed.

The identity of each aircraft is lost in an aggregate model. Thus, when routing is used as a control parameter, all aircraft involved in the problem must have the same destination airport. Such a problem is proposed, for example, in \cite{37}. If multiple destinations are required, traffic flow can be aggregated based on destination.

In current operations, human air traffic controllers direct individual aircraft. In order to achieve the control design objective, air traffic controllers can use the routing or delay parameters generated by the proposed methods as a guideline when determining air traffic control commands for individual aircraft.

In this chapter, we investigate the applicability of an aggregate control technique on a realistic simulation of flights in the NAS through FACET. We use the \textit{SMC} technique described in Chapter 5. Flight information
is used to generate an aggregate model which can be used to describe the flow of aircraft in the region of the NAS of interest. Using this model, the SMC method is used to generate the delay parameter $\alpha$. The disaggregation of this control parameter is achieved by implementing delay as speed reduction. All flights involved in the problem are reduced by the same factor $(1 - \alpha)$. It is not reasonable to assume that this control input can be continuously varied over time due to logistical complications in communicating these changes to flights. Thus we propose to set a value of the control parameter and keep it constant over some fixed time horizon, updating the value periodically.

### 6.2 Motivation and Problem Description

As detailed in Section 1.2.1, many approaches to air traffic management problems with either en route sector capacity constraints or flow rate constraints assume a deterministic forecast of these constraints. It is understood that without replanning, if the realized weather conditions call for more restrictive constraints, constraint violations will occur.

In this chapter, we propose using the SMC method, described in detail in Chapter 5, to react to flow rate constraints as they are realized in real time. This method generates airborne delay in order to satisfy flow rate constraints in real time. However, airborne delay is costly and logistically difficult to implement and we would therefore like to limit the use of airborne delay used for stringent constraint satisfaction. Thus, we propose using a two step approach. First, schedule ground delay based on the uncertain constraint forecast using a control algorithm which requires a deterministic constraint prediction. As the constraints are realized in real time, use the SMC method to apply further delay in the form of airborne delay in order to satisfy the realized constraints. Using this combined strategy, we make use of the uncertain constraint forecast to delay flights on the ground and apply airborne delay when necessary to reduce capacity constraint violations when the realized constraints are more restrictive than those predicted.

Specifically, we are interested in the following problem.

**Problem 6** Given uncertain predicted arrival rate constraints and real time updates of these constraints, control traffic arriving at the Knox arrival fix (OXI) of Chicago O’Hare International airport (ORD), through a combination of departure delays and airborne delays, to supply flights at or below the prescribed rate.

This is a simplified version of the problem addressed in [37]. As in [37], we focus on traffic on August 24, 2005 in the 3 hour time window from 14:00 CDT to 17:00 CDT (19:00 UTC to 22:00 UTC).
We address this problem using the two phase solution outlined above. First, we use the *IP Delay Scheduling* method presented in [37] to schedule departure delays given the uncertain prediction of the arrival constraints available at the beginning of this 3 hour time window. An overview of the *IP Delay Scheduling* method is given in Section 6.4. As the realized arrival constraints are revealed over time, we use the *SMC* method described in Chapter 5 to determine airborne delay required to satisfy the actual arrival constraints.

### 6.3 Overview of FACET

In this chapter, we focus on the implementation of the proposed control technique on a realistic simulation of the NAS. The Future ATM Concepts Evaluation Tool (FACET) [17] is a software tool developed at NASA Ames Research center to simulate air traffic in the NAS. It is a high fidelity flight-by-flight simulation tool used to support advanced ATM concept development and analysis. This tool can be used to play back traffic data from a particular date and time. FACET advances each flight in the NAS along its own four-dimensional trajectory. Position, heading and ground speed information for each flight is updated once every time step, where the time step of the simulation is specified by the user. The software includes the geometric representation of Air Route Traffic Control Centers (ARTCCs or “Centers”) and airspace sectors (low, high and super-high). More details about the architecture and capabilities of FACET can be found in [17]. Using FACET a variety of control inputs can be applied to implement a specific control algorithm. For example, a flight plan, ground speed or airborne delay can be specified for each flight in the simulation at each time step.

In this chapter, FACET is used to play back actual traffic data. During this play back, information is collected which is used to develop an aggregate model to describe a particular region of airspace of interest. After the development of this model, FACET is used to simulate flights involved in the problem of interest and apply the proposed control input. By implementing the proposed control technique through the FACET simulation of flights in the NAS, we are able to obtain realistic performance measures for the application of these techniques.

### 6.4 Overview of Integer Program Solution

In this section, we give an overview of the *IP Delay Scheduling* method presented in [37]. The *IP Delay Scheduling* method is used to schedule departure delays and pre-departure re-routes to satisfy deterministic airspace capacity constraints of en route sectors and arrival rate constraints at arrival fixes A discrete time linear model is used to describe the flow of aircraft through the network of interest. A linear integer program
(IP) is proposed to generate the control input parameters consisting of route and departure time selection for each flight involved.

Input required for this method is a list of flights and one or more routes that each flight may take from its origin airport to the destination airport of interest. The route information required is a list of the sectors visited by the flight along the particular route and the time that the aircraft spends in each sector (referred to as the dwell time in that sector). Given that a discrete time aggregate model is used to describe the air traffic network, the dwell times must be rounded to an integer multiple of the time step used in the model. Constraint information required includes the available capacity of each en route sector and the arrival rate constraints at the arrival fixes for each time step in the planning horizon.

As mentioned above, an IP is used as the solution method. The constraints of the IP are formulated to describe the dynamics of the flow of aircraft through the network of interest. Constraints are imposed to ensure that each aircraft spends the specified dwell time in each sector before advancing to the next sector. Additional constraints are imposed to ensure that the given sector capacity and arrival rate constraints are satisfied and that arrivals are separated by some minimum inter arrival time. The objective of the IP is the minimization of ground delay and additional flight time due to pre-departure re-routes.

In [37], the time step size is 60 seconds. In the work presented here, we use a time step of 12 seconds and the algorithm presented in [37] is modified accordingly. The smaller time step is used in this work in order to reduce errors caused by rounding departure and dwell times to an integer number of time steps. Although the IP Delay Scheduling method has the capability of assigning pre-departure re-routes, we restrict this solution to departure delays in order to simplify the problem formulation and analysis of results. Additionally, en route capacity constraints are not considered in the problem addressed here.

6.5 SMC Implementation Details

The derivation of the SMC method is described in detail in Section 5. In this section, the details of developing an empirical model of the region of airspace of interest using flight data and the implementation of aggregate control input in the flight-by-flight FACET simulation are discussed.

6.5.1 Generation of Aggregate Model Parameters

Data from May 8, 2007 was used to generate the parameters for the aggregate model. This day was chosen because it has been determined to have good weather throughout the NAS. Using departure times and flight
plans for this date, FACET was used to simulate the movement of flights throughout the NAS. A time step of 12 seconds was used to advance flights along their trajectories.

Only flights with the destination of ORD and flight plans which takes them through the OXI arrival fix at some time in the specified time range of 14:00 CDT and 17:00 CDT were used for the generation of the model. For each of these selected flights, the latitude, longitude and ground speed were logged.

High altitude sectors within a 700 nm range of ORD were used to divide the airspace of interest into a network of interconnected sections. That is, each high altitude sector acts as a section in the network. An aircraft is considered to be within a given sector if the latitude-longitude position of the aircraft are within the latitude-longitude boundaries of the given sector (i.e. the altitude of the flight and altitude boundaries of the sectors were not taken into consideration). Only sectors which were used by flights matching the above selection criteria were used. For the specific date and time of interest, 50 sectors were used to model the region of airspace of interest. For each time step, the position of each flight at that time step was used to generate sector counts (the number of aircraft in the sector). Transitions of flights from one sector to another were also determined.

The connectivity of the sectors was found based on the flow of aircraft from one sector to the next. The sectors used in the model and their connectivity is shown in Figure 6.1. Using the sector count and transition data, traversal times for each sector and routing parameters for adjacent sectors were calculated. Average traversal times range from 0.5 minutes to 15 minutes. These traversal times were generated using the selected flights only and flight plans through these sectors may only go through a small section of a given sector, resulting in a low traversal time.

Using this aggregate model and the actual initial sector counts and departures, an estimate of the state of the system was obtained using model (5.5). As an example of typical results obtained with this model, the actual and estimated sector count for sector ZAU34 is plotted Figure 6.2. This sector is the final sector of the network which leads to the arrival fix. In Figure 6.2(a), the actual and estimated state of this sector is plotted at each time step. The actual state is the number of aircraft in that sector as found through the FACET simulation. The estimated state is the number of aircraft in the sector estimated using the aggregate model of the network.

The aggregate model is only able to capture the average behavior of the flow of aircraft throughout the network. Thus it is expected that the high frequency changes in state observed in the FACET simulation data will not be captured by the state estimate generated using the aggregate model. We can see from 6.2(a)
Figure 6.1: FACET screen capture of the region of airspace of interest for this problem. Sectors included in the aggregate model are outlined in white, the grey lines indicate flows of air traffic between these sectors leading to the arrival fix, indicated by the white circle.
that this is indeed the case, with the difference between the actual and estimated number of aircraft in the sector differing by an average of 1 aircraft per time step and as much as 5.5 aircraft per time step. The actual and estimated state averaged over 15 minute time periods are plotted in Figure 6.2(b). As expected, the averaged values of the actual and estimated state are more closely matched than the values at each time step. In this case, the difference between the actual and estimated average sector counts differ by an average of 0.25 aircraft and at most 1.4 aircraft over each 15 minute time interval.

The accuracy of the model state as an estimate of the actual state can be improved by periodically resetting the value of the state of each sector in the model system to the state of the actual system. Resetting the state of the model every 30 minutes, the difference between the actual and estimated number of aircraft in ZAU34 differs by an average of 0.78 aircraft per time step and a maximum of 2.6 aircraft per time step.

Looking at the average sector count over 15 minute time intervals, the differences actually increase slightly to an average of 0.34 and a maximum of 1.5 aircraft per 15 minute interval.

Given that the objective of this problem is to regulate arrivals at a capacity constrained arrival fix, we now compare the actual and estimated arrivals at the arrival fix. Actual and estimated arrivals at the OXI fix summed over 15 minute time periods are plotted in Figure 6.3. No state reset was used in the generation of the data plotted in Figure 6.3(a). The state of the estimated system was reset to the state of the actual system every 30 minutes to generate the estimated OXI arrivals shown in Figure 6.3(b). Resetting the model state reduces the average and maximum difference between actual and estimated arrivals in 15 minute time periods from 1.3 aircraft and 3.3 aircraft to 0.93 aircraft and 2.7 aircraft, respectively.
Figure 6.3: Actual and estimated number of arrivals at the arrival fix summed over 15 minute time intervals with (a) no state reset (b) the state of the model system reset to the state of the actual system every 30 minutes.

### 6.5.2 Generation and Application of Control Input

Given the empirical network model generate using the method described in Section 6.5.1 and arrival rate constraints revealed in real time, the SMC algorithm is used to determine a scalar control parameter $\alpha(t)$. Every 15 minutes, arrival rate constraints are given for the following 15 minute interval. Given these constraints and departures for the next 15 minutes, the solution to the model system and the associated time varying control input $\alpha(t)$ are found for the following 15 minute interval. The average value of $\alpha(t)$ over this 15 minute interval is found. This average value, $\hat{\alpha}$, is applied as the actual control input to the real system through FACET. Control is implemented on the flights in the FACET simulation by reducing the ground speed of all airborne flights by a factor of $(1 - \hat{\alpha})$. In order to improve the accuracy of the arrival rate estimate, the state of the model system is updated every 30 minutes to the value of the state of the actual system at that time.

### 6.6 Problem Details

We focus on the traffic of August 24, 2005 in the 3 hour time window from 14:00 CDT to 17:00 CDT (19:00 UTC to 22:00 UTC). A total of 116 flights with destination ORD arriving through the OXI fix are either airborne or depart during this time period. Of these flights, 80 arrive at the OXI fix during this time period. These arrivals, summed over 15 minute time periods, are plotted in Figure 6.4

The 50 high altitude airspace sectors used to generate the aggregate model are used for both the SMC
FACET was used to gather route and dwell time information for each flight that departs in that 3 hour window. The route for a flight consists of the sequence of sectors that the flight visits on its path from the departure airport to the OXI fix. The dwell time of a flight in a given sector is rounded to an integer number of time steps. In this case, 12 second time steps were used in an attempt to decrease errors due to the rounding of these dwell times.

The *IP Delay Scheduling* method is used to schedule departure delay and thus can only be used to schedule delay for flights which depart during this time window. Thus, a limited number of these 80 arrival flights are subject to delays calculated by the *IP Delay Scheduling* method. In this case, a total of 52 flights depart during this time window and arrive at the OXI fix during this time window.

Although the *IP Delay Scheduling* method can be used to ensure that en route sector capacity constraints are satisfied, in order to simplify the problem and comparison between the *IP Delay Scheduling* method and the *SMC* method, en route sector capacity constraints were not considered in this example. Spacing constraints were set to ensure that controlled arrivals are separated by at least 60 seconds.

The *SMC* method assigns airborne delay to flights scheduled to arrive at the OXI fix during this window. Thus, all airborne flights scheduled to arrive at OXI during this window may be subject to airborne delay as specified by the *SMC* solution.
6.7 Simulation Results

Four capacity constraint profiles are used in these simulations and are given in Figure 6.5. Profile 1 in Figure 6.5(a) is the baseline constraint prediction. An arrival rate of 10 aircraft per 15 minute time period is considered nominal. Profile 1 begins and ends with constraints of 10 aircraft per 15 minute time period. The capacity constrained time frame is the interval of time over which capacity constraints are lower than this nominal value. The capacity constrained time frame of Profile 1 is from 14:30 CDT to 16:30 CDT. The remaining three profiles are variants of Profile 1, in which the severity and timing of the capacity constraints are altered compared to Profile 1. Profile 2 has more severe constraints than profile 1, with constraints lowered by 2 aircraft per 15 minute time period during the capacity constrained time frame. Profile 3 is a shift of Profile 1 forward in time. And finally, Profile 4 is a shift of Profile 2 back in time.
We assume that one of these four profiles represents the predicted arrival constraints for the 3 hour time period of interest. The arrival rate constraints which are actually realized may differ from the predicted constraint profile and are revealed in real time in 15 minute increments. Several FACET simulations were performed using different combinations of these profiles as the predicted arrival constraint profile and the realized arrival constraint profile and various combinations of the delay scheduling algorithms to control the flights involved. A description of each of these simulations is given below, with results summarized in Table 6.7. Given that arrival constraints are specified for 15 minute intervals, we are concerned with satisfying these constraints for these intervals. Thus, in order to evaluate the performance of each of the control techniques implemented, we sum the actual arrivals over each of these 15 minute intervals. We count arrival constraint violations as the number of arrivals in a given 15 minute time period in excess of the arrival constraint during that same 15 minute period. Similarly, underutilization is calculated as the arrival constraint for a given 15 minute time period minus the number of actual arrivals in that 15 minute time period. Performance metrics reported in Table 6.7 include the total arrival violations over the entire planning window, the maximum constraint violation in any 15 minute interval and the total underutilization over the entire planning window. The total delay scheduled, number of flights affected by delay, maximum delay incurred by any one flight and the average delay over all flights involved in the scenario are reported for departure delays, airborne delays and the combination of departure and airborne delay.

Results for the case in which no control action is taken are presented in the first section of Table 6.7, labeled “No Control.” Each constraint profile was used as the realized constraint profile. These results provide a baseline for the performance measures of total constraint violations, the maximum constraint violation and capacity underutilization. Arrival rate results for constraint Profile 1 with no control are shown graphically in Figure 6.6. The reported values of violations and underutilization are calculated from the arrival constraint and actual arrivals summed over 15 minute intervals.

In section “IP Departure Delay – Perfect Prediction” of Table 6.7 results are given for FACET simulations using the IP Delay Scheduling method. The profile listed in a given row is used as both the predicted and realized constraint profile. This is in contrast with the next section and the final section of the table in which the predicted constraint profile is always Profile 1. A FACET simulation was performed using the prescribed departure delays to generate the actual arrivals at OXI. Sample results for Profile 1 are shown in Figure 6.7.

Looking at the results for all constraint profiles under section “IP Departure Delay – Perfect Prediction” in Table 6.7, it can be seen that even with perfect constraint profile prediction, the constraint profile is violated. This is due to errors in rounding the the departure time and arrival time at the arrival fix to the beginning
Figure 6.6: Constraint Profile 1 and actual arrivals when no control action is applied. These results correspond to the Constraint Profile 1 row under section “No Control” of Table 6.7.

Figure 6.7: Constraint Profile 1 and actual arrivals when departure delays were set using the IP Delay Scheduling method with Profile 1 as the predicted constraint profile. These results correspond to the Constraint Profile 1 row under sections “IP Departure Delay – Perfect Prediction” and “IP Departure Delay – Profile 1 Predicted” of Table 6.7.
or end of a discrete time step. Errors are also introduced as the flight is simulated to fly through en route airspace.

Next, the *IP Delay Scheduling* method is used to generate departure delays given Profile 1 as the predicted arrival constraint profile. All four arrival constraint profiles are used in separate FACET simulations as the realized constraints. However, the departures generated using Profile 1 as the predicted constraint profile are implemented regardless of the realized constraint profile. The results of these simulations are reported in section “IP Departure Delay – Profile 1 Predicted” in Table 6.7. Results given in the first row of this section correspond to Profile 1 used as both the predicted and realized constraint profile and are identical to the results in the first row of section “IP Departure Delay – Perfect Prediction” in Table 6.7, which are also plotted in Figure 6.7.

Finally, the *SMC* and *IP Delay Scheduling* methods were used in combination to make use of both the uncertain arrival constraint prediction and constraints as they are realized in real time. In this case, the *IP Delay Scheduling* method was used to generate departure delays given Profile 1 as the predicted arrival constraint profile. Four separate FACET simulations were then performed using the resulting departure delays calculated for Profile 1 with realized arrival constraints of Profiles 1, 2, 3 and 4. The *SMC* method was applied to generate airborne delay to satisfy arrival constraints as they were revealed in real time. Results of these simulations are presented in section “Sliding Mode Control with IP Departure Delay for Profile 1” of Table 6.7. The arrival constraints and actual arrivals for the case in which Profile 1 is the realized constraint are plotted in Figure 6.8.
Figure 6.8: Constraint Profile 1 and actual arrivals when departure delays were set using the *IP Delay Scheduling* method with Profile 1 as the predicted constraint profile and the *SMC* method was used to add airborne delay in order to satisfy arrival rate constraints as they were revealed in real time. These results correspond to the Constraint Profile 1 row under section “Sliding Mode Control with IP Departure Delay for Profile 1” of Table 6.7.

Figure 6.9: Control parameter generated by the *SMC* method for the FACET simulation results shown in Figure 6.8. The dashed curve is the continuous value of $\alpha$ generated by the *SMC* method using the aggregate model estimate of the state of the system. The solid curve is the piecewise constant average $\hat{\alpha}$ of the continuous value of $\alpha$. The control parameter input used for the FACET simulation is the averaged $\hat{\alpha}$. 

122
Table 6.1: Performance and delay results for the application of various delay scheduling algorithms and procedures.

<table>
<thead>
<tr>
<th>Constraint Profile</th>
<th>Departure Delay</th>
<th>Airborne Delay</th>
<th>Total Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total Violations</td>
<td>Maximum Violation</td>
<td>Underutilization</td>
</tr>
<tr>
<td>No Control</td>
<td>1 14 5 24</td>
<td>0 0 0 0</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td></td>
<td>2 25 7 19</td>
<td>0 0 0 0</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td></td>
<td>3 12 5 24</td>
<td>0 0 0 0</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td></td>
<td>4 13 5 23</td>
<td>0 0 0 0</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td>IP Departure Delay – Perfect Prediction</td>
<td>1 2 2 20</td>
<td>1322 46 100 17</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td></td>
<td>2 7 3 20</td>
<td>2353 50 123 29</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td></td>
<td>3 3 3 19</td>
<td>1186 47 105 15</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td></td>
<td>4 2 2 23</td>
<td>1163 41 101 15</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td>IP Departure Delay – Profile 1 Predicted</td>
<td>1 2 2 20</td>
<td>1322 46 100 17</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td></td>
<td>2 10 2 12</td>
<td>1322 46 100 17</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td></td>
<td>3 3 2 23</td>
<td>1322 46 100 17</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td></td>
<td>4 5 3 23</td>
<td>1322 46 100 17</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td>Sliding Mode Control with IP Departure Delay for Profile 1</td>
<td>1 0 0 27</td>
<td>1322 46 100 17</td>
<td>505 57 18 6</td>
</tr>
<tr>
<td></td>
<td>2 2 2 25</td>
<td>1322 46 100 17</td>
<td>1630 50 70 20</td>
</tr>
<tr>
<td></td>
<td>3 0 0 28</td>
<td>1322 46 100 17</td>
<td>739 58 28 9</td>
</tr>
<tr>
<td></td>
<td>4 1 1 34</td>
<td>1322 46 100 17</td>
<td>554 51 27 7</td>
</tr>
</tbody>
</table>
The “No Control” scenarios give a baseline for the total constraint violations and maximum constraint violations obtained when each of the constraint profiles is the realized constraint profile, and no control action is taken. The delay calculated using the IP Delay Scheduling method is the theoretical minimum delay required to satisfy the arrival constraints, with violations of these constraints occurring in simulation due to errors discussed above. However, in practice, it is generally not possible to achieve this minimum delay solution since a perfect prediction of the arrival constraints does not exist.

In the “IP Departure Delay – Profile 1 Predicted” section, results are shown for scenarios in which the predicted constraint profile may not match the realized profile. In this case, violations of the arrival rate constraints are reduced compared to the “No Control” case due to the similarities of the predicted arrival constraint profiles. However, significant constraint violations do occur.

We use the SMC method to apply airborne holding in addition to the departure delays scheduled using the IP Delay Scheduling method in order to decrease constraint violations. Results of these simulations are given in Table 6.7 under “Sliding Mode Control with IP Departure Delay for Profile 1.” In these scenarios, Profile 1 is the predicted constraint profile and departure delays calculated for this profile are implemented. However, as the actual arrival constraints are realized, the SMC method is used to calculate additional delay required to satisfy these realized constraints and implemented via airborne delay. The total delay implemented using this method is 1.2 to 1.7 times of that imposed by the IP Delay Scheduling method in the case that perfect prediction is possible. Given that perfect constraint prediction is not possible, it may be advantageous to trade additional delay in order to reduce constraint violations. Whether or not it is advantages to incur additional delay to satisfy the realized constraints depends on the relative importance of satisfying arrival rate constraints compared to the cost of additional airborne delay.

Underutilization is also reported for each control method. This is a rough measure of the conservativeness of each solution method. The procedure of using the SMC method on top of the IP Delay Scheduling method with an inaccurate constraint prediction increases the underutilization of the available arrival capacity by 1.25 to 1.5 times that of the IP Delay Scheduling method with perfect constraint prediction. However, 62% to 70% of the available capacity is used.
Chapter 7

Conclusions

7.1 Summary

In this work, we addressed a variety of fundamental problems in air traffic flow management. Specifically we addressed the problems of (a) generating fixed routing parameters to minimize delay and satisfy integral constraints, (b) generating time varying routing parameters to satisfy piecewise constant capacity constraints for fixed linear, uncertain linear, and nonlinear aggregate models, and (c) scheduling airborne delay to satisfy time varying flow rate constraints. LP based methods are used to approach problems (a) and (b) and sliding mode control theory is used to approach problem (c). Each method uses a continuous time aggregate model to describe the flow of air traffic through the region of airspace of interest.

These specific problems and solution methods were chosen in order to address problems that have so far received little attention in the literature. Little work has been done with the use of routing parameters as the control input. Here we develop a routing parameter design technique to minimize delay in an infinite horizon problem. We also show that routing control design can be effectively used to satisfy time varying airspace capacity constraints for continuous time aggregate models of air traffic. This method makes use of a deterministic capacity constraint forecast and could be developed into a receding horizon control method in order to make use of updated capacity constraint forecasts as they are realized. Alternatively, uncertainty in airspace capacity constraints could be explicitly incorporated into the control design problem using a method similar to that used in Section 4.4 to account for uncertainty in section traversal times.

The SMC method directly addresses the problem of planning for air traffic flow management with uncertain constraints. This method requires no prediction of the constraints and instead reacts in real time to constraints as they are realized. Given that this method controls flights to satisfy constraints revealed in real
time, it relies on airborne delay to control the traffic flow network in order to satisfy these constraints. Since airborne delay is costly, we proposed to use the SMC method in combination with one of the many scheduling algorithms available in the literature to schedule departure delays using an uncertain constraint forecast. Applied in this way, the SMC method is used to schedule airborne delay to satisfy flow rate constraints when they differ from the predicted flow rate constraints.

The development of the aggregate models used for the control design techniques presented here relies on some assumptions about the air traffic networks modeled. It is assumed that aircraft are evenly distributed throughout sections of airspace, that all aircraft travel through a given section at the same average speed and that the number of aircraft in a section is an infinitely divisible quantity. Despite these assumptions, the SMC method was found to perform well in a realistic simulation of the NAS through FACET. The use of the SMC method decreased arrival rate constraint violations compared to no control and departure delay scheduling using an inaccurate prediction of the arrival rate constraints.

7.2 Model Assumptions and Discussion

Each of the methods presented is suited for use in certain scenarios and has its own strengths and weaknesses. In this section, possible applications for each modeling and control method, along with assumptions made in the problem formulation and modeling of air traffic, are discussed for each method developed.

7.2.1 LP Routing (Integral)

LP Routing (Integral) method is used to generate static routing parameters for an infinite horizon problem with the objective of minimizing a measure of delay. Delay is formulated as an integral of the state of the network, giving an indication of how quickly traffic can be emptied from the network under the specified routing strategy. This method makes use of a fixed linear section outflow model, which can accurately describe air traffic when spacing requirements are not affecting the natural flow of traffic. That is, when traffic density is low enough such that inter-aircraft spacing required for safety is satisfied without the need to apply control actions. In this type of scenario, aircraft would be allowed to fly at their preferred, nominal speed, and thus the outflow rate of each section of airspace can be estimated to depend linearly on the number of aircraft in that section. Solutions would have to be checked a posteriori to ensure that the number of aircraft in each section remains in the linear outflow rate regime. This modeling and solution method could be used to identify the routes through a network of airspace with minimal delay under clear weather conditions. The resulting cost is an indication of the baseline, or nominal delay expected for flights.
traveling through the region of airspace of interest.

One advantage of the LP Routing (Integral) solution method is that it is a solution to an infinite horizon problem. Additionally, fixed routing parameters are given as the proposed control input. With fixed constant routing parameters for an infinite time horizon, strategic routing and scheduling plans can be made. Additionally, the capability to adjust solutions by adding constraints to limit traffic through certain regions of airspace or along specific paths through the network is given through the use of integral constraints.

Under adverse weather conditions or other constraints, section traversal times may be uncertain or time varying. This model incorporates neither a forecast of these types of variations in traversal time nor a method of representing uncertainty in traversal times. Thus, this control design technique is limited to strategic planning for fair weather scenarios.

### 7.2.2 LP Routing (Capacity)

The LP Routing (Capacity) methods make use of three different models to describe air traffic. The objective of each of the LP Routing (Capacity) methods is to design routing parameters to satisfy time varying capacity constraints. It is assumed that airspace capacity constraints are time varying and predictable over a finite time horizon. This is a reasonable assumption when considering the effects of weather on air traffic management. The effect of adverse weather typically is described as the reduction of airspace capacity of affected regions of airspace. Weather predictions can be translated into airspace capacity estimates with acceptable accuracy over finite time horizons, typically 2 to 3 hours.

Results for the fixed linear model presented in Section 4.3 are relevant when traffic density is low and section outflow rates can be assumed to depend linearly on the number of aircraft in each section. However, the more significant contribution of this method is that it serves as a basis for the development of routing control design techniques for the models used in subsequent sections which incorporate uncertainty in traversal times and nonlinear outflow rates.

In Section 4.4, in addition to time varying capacity constraints, the model and solution method incorporates variation and uncertainty in section traversal times. Variation or uncertainty in section traversal time could arise for a variety of reasons. For example, sections used by mixed aircraft types will have a range of traversal times, depending on each aircraft’s nominal flight speed. Alternatively, section traversal time can be affected by adverse weather. Flights traveling through a given section of airspace may need to be routed around a region of severe weather, increasing the path length traveled through that section and increasing travel
time. Given the uncertainty of weather predictions, in both severity and location, the specific location and magnitude of the re-routing (and related traversal time increase) is also uncertain. The model used in Section 4.4 can describe both a range of possible traversal times and uncertainty in traversal times. A drawback of this method is that the saturating outflow rate effect seen in dense traffic is not incorporated in this model. Incorporating saturating section outflow rates in the dynamic model is the focus of Section 4.5. Time varying, state dependent, routing control is derived for the capacity constraint problem. Again, a capacity constraint prediction is required over a finite time horizon. This solution method is best applied when traffic is predicted to be dense enough such that section outflow rates are in the nonlinear or saturating region. That is, when traffic is dense enough such that control action is required in order to ensure proper spacing between aircraft, thus reducing section outflow compared to a purely linear outflow rate estimate.

7.2.3 \textit{SMC}

While the control design methods of Chapter 4 focus on control design given constraint predictions over a finite time horizon, the \textit{SMC} method is used to derive a control design technique to satisfy time varying constraints that are revealed in real time. As opposed to section capacity constraints, this method is designed to satisfy network outflow rates. Such constraints can arise when metering traffic traveling into an arrival rate constrained airport or a Flow Constrained Area (FCA). The benefit of this method is that a prediction of the flow rate constraints is not required. However, airborne delay is the only control parameter available when using this method and could lead to an unacceptable amount of airborne delay. Thus the ideal application of this method, as illustrated in Chapter 6, is as a tactical delay scheduler implemented on top of a strategic planner.

As presented, the \textit{SMC} method applies the same speed reduction to flights in all sections of the network. Delaying flights in sections far from the network sink has no affect on the network outflow and will likely introduce unnecessary delay. In order to reduce delay in sections far from the network sink, the controlled speed reduction in each section could be discounted based on the distance from the section to the sink. That is, sections close to the sink would be given the full specified speed reduction, while flights in sections farther from the sink would be given little or no speed reduction.

7.3 Possible Extensions

Both the \textit{LP Routing (Capacity)} method and the \textit{SMC} methods can be modified and combined to better address the flow rate constraint problem. While the focus of the \textit{LP Routing (Capacity)} methods is routing
control design in the presence of section capacity constraints, all of these methods can be used to solve the arrival rate constraint problem, similar to the problem address by the SMC method. In all three models used in Chapter 4, section outflow is dependent on the number of aircraft in each section. Thus, constraints on the network outflow rate can be imposed through constraints on the state of final sections. For example, consider the problem of satisfying flow rate constraints at an arrival fix, introduced in Chapter 6. An airspace network could be defined to describe the region of airspace of interest and consist of only one final section leading to the constrained arrival fix. The flow rate out of that final section depends on the number of aircraft in the section. Thus, a capacity constraint for that final section can be calculated such the outflow of that section (or maximum possible outflow of the section in the case of the uncertain traversal time model) at full capacity is at or below the arrival rate constraint. Such a solution would require a prediction of the arrival rate constraint. This method could be used to strategically schedule routing to satisfy the predicted arrival rate constraint.

A tactical control algorithm, such as the SMC method, could be used to react in real time to the realized arrival rate constraint. Given that the SMC method is a nonlinear control method, it could be modified to be used with the nonlinear, saturating outflow model used in Section 4.5. The SMC method could then be used in conjunction with any one of the LP Routing (Capacity) methods modified for the flow rate constraint problem.

In this way, we could combine the strategic routing solution used to satisfy capacity constraints and network outflow rates given predictions of both types of constraints over a finite time horizon, with the tactical control method to satisfy the actual flow rate constraints as they are revealed in real time. The use of the SMC method to delay traffic may push section capacity beyond the specified bounds. The hope is that with a reasonable constraint prediction, only small adjustments to section outflow rates would be required to satisfy the network outflow rate constraints revealed in real time and thus capacity constraint violations would be acceptably small.

Routing and airborne delay are the control inputs available for each of the LP Routing (Capacity) methods. Departure rates could be incorporated as an additional control input with some changes to the network model. Additional sections can be added between sources (representing departure airports in this scenario) and the initial network sections to model a departure queue. Flights ready to depart from source $s$ would enter a departure queue section, $q$ at rate $b_{sq}(t)$, previously defined as the departure rate. Rather than forcing these flights into the airspace network, they would enter the queue sections. The outflow of the departure queue sections can be regulated by routing parameters, just as flow rates are regulated in regular
sections. The control problem would then include the choice of these routing parameters and the outflow of the departure queue sections would represent the controlled departure rate from the associated airport. Using the nonlinear outflow model, the dynamics of these queue sections could be defined in such a way that the saturating outflow rate is the maximum departure rate. The use of the queue sections ensures that both the maximum departure rate is not exceeded and that aircraft are not forced to depart before they are available, given that only flights which have entered the queue are available to enter the network. Recirculation of flights in the queue effectively reduces the departure rate from departure airports. Given a large capacity for these queue sections, aircraft would be allowed to accumulate in the queues, which represents ground delay. Thus, ground delay could be used to avoid exceeding airspace sector capacity constraints. In order to provide incentive for flights to leave the departure queue, the cost of the LP program could be modified to incorporate the number of aircraft in departure queue sections at the end of the planning horizon.
Appendix

Proof of Theorem 8

Given any \( x(0) \in \mathbb{R}^n_+ \) we can find \( \psi > V(x(0)) \). Given that \( V \) satisfies (5.15), there exists \( r > 0 \) such that whenever \( x \in \mathbb{R}^n_+ \) and \( ||x|| > r \), \( V(x) > \psi \). Define

\[
\Omega_\psi = \{ x \in \mathbb{R}^n_+ \mid V(x) \leq \psi \}.
\]

and

\[
W_r = \{ x \in \mathbb{R}^n_+ \mid ||x|| \leq r \}
\]

Defined in this way, we see that \( \Omega_\psi \subset W_r \).

We can see that \( \Omega_\psi \) is positively invariant since \( \Omega_\psi \) can be expressed as

\[
\Omega_\psi = \{ x \in \mathbb{R}^n \mid V(x) \leq \psi \} \cap \mathbb{R}^n_+.
\]

Thus, the boundary of \( \Omega_\psi \) consists of two parts, one defined by a level set of \( V \)

\[
V(x) = \psi,
\]

and the other defined by the set of hyperplanes

\[
\{ x \in \mathbb{R}^n \mid x_i = 0 \text{ for some } i \in \{1, \ldots, n\} \}.
\]

A trajectory \( x(t) \) cannot escape \( \Omega_\psi \) through the part of the boundary defined by the level set \( V(x) = \psi \).
because $V$ satisfies equation (5.16) and thus

$$
\dot{V}(x) < 0 \Rightarrow V(x) < V(x(0)) \leq \psi, \ \forall \ t \geq 0.
$$

The trajectory cannot escape through any of the $x_i = 0$ hyperplanes because the system is positive (i.e. $x(0) \in \mathbb{R}^n_+ \rightarrow x_i(t) \geq 0, \ \forall \ t \geq 0$).

Clearly, $\Omega_\psi$ is a compact set (it is a closed set bounded by $\|x\| = r$). Because of this, system (5.5) has a unique solution for all $t \geq 0$ whenever $x(0) \in \Omega_\psi$ (using Theorem 3.3, [29]).

To show asymptotic stability, we need to show that $\lim_{t \to \infty} x(t) = 0$. It is sufficient to show that $\lim_{t \to \infty} V(x(t)) = 0$. Given that $V$ is positive and satisfies (5.16),

$$
\lim_{t \to \infty} V(x(t)) = c. \quad (7.1)
$$

We can show that $c = 0$ by a contradiction argument. Suppose $c > 0$. By continuity of $V(x)$, there exists $d > 0$ such that $W_d \subset \Omega_c$ where

$$
W_d = \left\{ x \in \mathbb{R}^n_+ \mid \|x\| \leq d \right\}
$$

and

$$
\Omega_c = \left\{ x \in \mathbb{R}^n_+ \mid V(x) \leq c \right\}.
$$

The fact that $V(x(t))$ satisfies (7.1) implies that the trajectory $x(t)$ lies outside of the set $W_d$ for all $t \geq 0$. Let

$$
\gamma = -\max_{d \leq |x| \leq r, \ x \in \mathbb{R}^n_+} \dot{V}(x).
$$

By (5.16), $\gamma > 0$. It follows that

$$
V(x(t)) = V(x(0)) + \int_0^t \dot{V}(x(\tau))d\tau \leq V(x(0)) - \gamma t.
$$

We see that the right hand side will eventually become negative, thus we have a contradiction since we assumed that $c > 0$. Thus, $\lim_{t \to \infty} V(x(t)) = 0$ and the origin is asymptotically stable for any initial condition $x(0) \in \mathbb{R}^n_+$. \hfill \blacksquare
References


