Large-scale Modeling and Optimization of En Route Air Traffic Flow

by

Dengfeng Sun

B.ENG. (Tsinghua University, China) 2000

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Engineering-Civil and Environmental Engineering

in the

GRADUATE DIVISION
of the
UNIVERSITY of CALIFORNIA at BERKELEY

Committee in charge:

Professor Alexandre M. Bayen, Chair
Professor Mark Hansen
Professor Claire J. Tomlin

Fall 2008

The dissertation of Dengfeng Sun is approved:

Chair Date

Date

Date

University of California at Berkeley

Fall 2008

Large-scale Modeling and Optimization of En Route Air Traffic Flow

by

Dengfeng Sun

Abstract

Large-scale Modeling and Optimization of En Route Air Traffic Flow

by

Dengfeng Sun

Doctor of Philosophy in Engineering-Civil and Environmental Engineering

University of California at Berkeley

Professor Alexandre M. Bayen, Chair

The research presented in this dissertation is motivated by the need for balancing the increasing demand and limited capacity of the *National Airspace System* (NAS), and more generally for large scale networked dynamical systems.

A new paradigm for building an Eulerian-Lagrangian Cell Transmission Model for air traffic flow is developed. It is based on a network flow model constructed from historical air traffic data, and is applied to the entire continental NAS in the United States. This model is called *Large-capacity Cell Transmission Model*, CTM(L), in reference to the Cell Transmission Model in highway traffic. The CTM(L) captures fundamental characteristics, for example aircraft counts in sectors and travel times, of traffic flows in the *Air Traffic Control* (ATC) system. The predictive capabilities of the model are successfully validated against the recorded *Enhanced Traffic Management System* (ETMS) and *Aircraft Situation Display to Industry* (ASDI) data by showing an accurate match between predicted sector counts (based on filed flight plans) and measured sector counts.

Besides the CTM(L), four Eulerian network models are implemented to model high altitude air traffic flow. The four models are applied to high altitude traffic for six *Air Route Traffic Control Centers* (ARTCCs) in the NAS and surrounding airspace. Simulations are carried out for a full day of data for the models, to assess their predictive capabilities. The models' predictions are compared to the recorded flight data. Several error metrics are used to characterize the relative accuracy of the models. The efficiency of the respective models is also compared in terms of computational time and memory requirements for the scenarios of interest. Control strategies are designed and implemented on similar benchmark scenarios for two of the models. They use techniques such as adjoint-based optimization, and the *mixed integer linear program* (MILP). A discussion of the four models' structural differences explains why one model may outperform another.

Finally, the CTM(L) model is used for NAS-wide optimal *Traffic Flow Management* (TFM). A problem of controlling sector aircraft count is posed as an *integer program* (IP) in which the dynamical system appears in the constraints. A problem specific algorithm based on a dual decomposition method is designed to show that the large scale optimization problem which has billions of variables and constraints, can be solved efficiently, making real-time NAS-wide TFM possible. The CTM(L) model and the optimization algorithm for NAS-wide TFM are integrated in FACET, a software developed at NASA Ames Research Center, in collaboration with Metron Aviation.

Professor Alexandre M. Bayen
Dissertation Committee Chair

To my family

Contents

List of Figures

List of Tables

List of Algorithms

Acknowledgements

My deepest gratitude goes to my Ph.D. advisor, Professor Alexandre Bayen, who is, in every sense of the word, a devoted teacher, caring mentor and role model to all his students. His support and advice has helped me to achieve the accomplishment of finishing my Ph.D. program and so much more. His pleasant personality and the respect he has for his students have certainly made this journey enjoyable. I will always be indebted to him for everything he has taught me and given to me.

I would like to thank Professor Mark Hansen, Professor Claire Tomlin, Professor Laurent El Ghaoui, Professor Raja Sengupta, and Professor Samer Madanat for serving on my Dissertation, Preliminary Exams and Qualifying Exams Committee. Professor El Ghaoui's expertise in optimization, Professor Tomlin's expertise on air traffic control and Professor Hansen's expertise on air transportation are extremely valuable sources that have enriched my knowledge and culture in these areas. Professor Steven Glaser has given me unlimited support and encouragement for my Ph.D. research. Professor Mike Ball from the University of Maryland gave me valuable suggestions on traffic flow management during his visit at Berkeley. Professor Carlos Daganzo is gratefully acknowledged for the insightful discussions about his Cell Transmission Model and his suggestion to name the model presented in this dissertation the Large-capacity Cell Transmission Model.

It has been a very pleasant and enriching research partnership with Professor Shankar Sastry and Professor Ruzena Bajcsy, who led me to the area of control and robotics, sensor networks, and computer graphics, and helped me through the most difficult times during my Ph.D. program. I am grateful towards Professor Roberto Horowitz, who gave me a lot of useful advice, and funded me during my early study at Cal.

Thanks go to my lab colleagues: Saurabh Amin, Christian Claudel, Alexis Clinet, Juan Carlos Herrera, Stephane Martinez, Jessica Pennquin, Tarek Rabbani, Charles Robelin, Issam Strub, Andrew Tinka, Daniel Work, Qingfang Wu and Sam Yang – there is never a lack of excellent research partners and wonderful friends. Part of this dissertation is joint work with Alexis, Charles, Issam and Stephane. I am also thankful to Volkan Isler, Sang-Hack Jung (CITRIS), Gabriel Gomes, Laura Munoz, Xiaotian Sun (Mechanical Engineering), Gautam Gupta, Jay Hsiao, Jing Xiong, Helen Yin and Yu Zhang (NEXTOR) for many discussions about research and for such pleasant collaborations.

It is always a blessing to be surrounded by a large group of intelligent and kind people at Cal. My friends Xingyuan Chen, Min Chen, Xu Guan, Yurong Hu, Ching-Ling Huang, Xinghui Huang, Jiayang Juang, Sha Li, Yunfeng Li, Sa Liu, Lei Luo, Ning Ma, Mingyu Shi, Kunpeng Sun, Shan Tang, Jingtao Wang, Zile Wei, Guoyuan Wu, Jianchun Xu, Li Xu, Li Yang, Yafeng Yin, Qiaolin Zhang, Ying Zhang, Yun Zhou and Jiang Zhu have made my life at Cal especially colorful and enjoyable.

This work was supported by NASA under Task Order TO.048.0.BS.AF and NASA NRA Subtopic 4. I owe thanks to my supervisor at NASA Ames Research Center, Dr. Banavar Sridhar, and other researchers Dr. Shon Grabbe, Dr. Kapil Sheth and Dr. Gano Chatterji for their valuable suggestions on my research.

Finally, I would like to thank my family for their endless love and support. This dissertation is dedicated to them.

Chapter 1

Introduction

The aim of this dissertation is to provide efficient modeling and optimization algorithms for en route air traffic to mitigate the imbalance between the demand and capacity of the current *National Airspace System* (NAS) in the United States. In this work, a new aggregate traffic model, the *Large-capacity Cell Transmission Model* or CTM(L), will be developed. Based on the CTM(L), optimization algorithms will be designed for NAS-wide *Traffic Flow Management* (TFM).

Figure 1.1 illustrates the modeling, optimization and control framework of this dissertation. The present chapter will present the background, related research work, contribution and organization of this thesis. This figure can be summarized as follows: The CTM(L) component is one of the modeling contributions of this work, which can incorporate systems disturbances. The optimization algorithm component includes contributions made in the field of optimization, applied to the CTM(L) model. The closed loop diagram depicts the TFM contribution, i.e., the demonstration that the methods developed here are applicable to air traffic control.

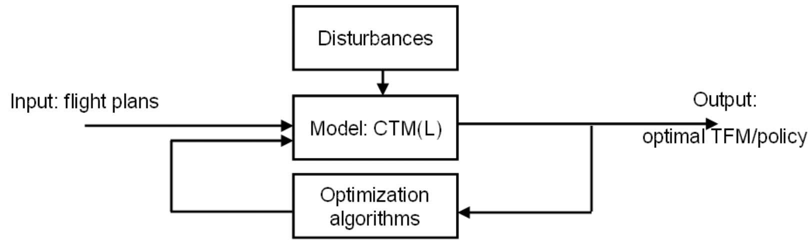

Figure 1.1: Modeling, optimization and control framework used in this work.

1.1 Air traffic systems

The NAS in the United States is a large scale, nonlinear dynamic system with a control authority that is organized hierarchically. A single *Air Traffic Control System Command Center* (ATC-SCC) in Herndon, VA, supervises the overall traffic flow nation-wide. Organized by geographical region, the airspace is divided into 22 (20 in the continental US) *Air Route Traffic Control Centers* (ARTCCs, or simply, Centers), controlling an airspace of up to 60,000 feet [55]. Each Center is sub-divided into smaller regions, called *sectors*, each of them being a portion of airspace containing aircraft under the local control of the responsible Air Traffic Controller. At least one Air Traffic Controller is responsible for each sector, and can manage the traffic flow inside (and sometimes outside) the sector. Within each sector, navigation infrastructure, including jetways, waypoints and navigation aids, is used to help with the traffic flow management.

The last few decades have witnessed an almost uninterrupted growth of air traffic [15]. Except for a dip immediately after the tragic events of September 11, 2001, air traffic in the United States has continued to grow at a steady pace. There are different growth scenarios associated both with the magnitude and the composition of future air traffic. The *Terminal Area Forecast* (TAF), prepared every year by the *Federal Aviation Administration* (FAA), projects a continued growth of traffic in the United States. For example, the 35 *Operational Evolution Plan* (OEP) airports enplaned 470.3 million passengers in 2003, which is projected to increase to 847.6 million passengers in 2020 [6]. The total active general aviation aircraft fleet, total hours flown by aircraft and total active pilots will grow accordingly, and air traffic operations will increase as well. Since the main goal of Air Traffic Controllers is to maintain a safe separation between aircraft while guiding them to their destinations, an imbalance between the growth of air traffic and the capacities (more generally, airspace capacity) of Air Traffic Controllers poses potential problems to the air traffic control system. Figure 1.2, generated using the *Future ATM Concepts Evaluation Tool* (FACET) [14], shows an example of the traffic situation on January 1st, 2005.

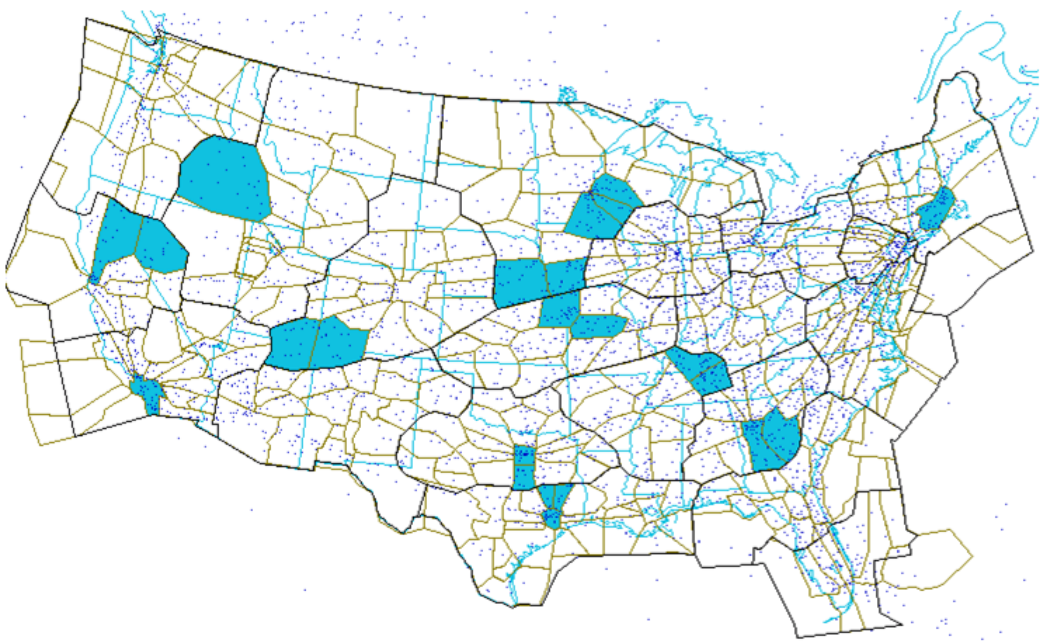

Figure 1.2: An illustration of sector loads in the NAS. Aircraft are presented by dots. Shaded areas represent overloaded sections in the NAS in the morning of January 1st, 2005. It is a result of the imbalance between demand and capacity in each area, which poses potential safety problems.

Thus, the imbalance between the growth of air traffic and airspace capacities has motivated the design of a semi-automated *Air Traffic Control* (ATC) system to help Air Traffic Controllers manage the increasing complexity of traffic flow in the en route airspace [51].

Recently, there has been an increase in research to develop analysis tools and methods that partially automate some of what is manually performed by Air Traffic Controllers today. Any automation process for an existing system poses numerous challenges. Two key problems appear: (i) understanding the characteristics of the current system, in order to predict its reaction under stress or severe conditions; (ii) developing scientific approaches in order to optimize traffic flow and relieve Air Traffic Controllers from a part of their traffic control workload. The rest of this chapter will address background work and promising approaches to tackle these two problems.

1.2 Traffic Flow Management (TFM)

TFM is the management of air traffic in the NAS, based on system capacity and demand [1; 55]. This is accomplished by using a "system approach," which is a management approach that considers the impact of individual actions as a whole. TFM handles demand management and typically deals with traffic at the ARTCC level, i.e. 10 to 20 sectors. TFM personnel consider the demand on the system and use tools, known as *Traffic Management Initiatives* (TMIs), which includes *Special Traffic Management Programs* (STMPs), *Ground Delay Programs* (GDPs), *Airspace Flow Programs* (AFPs), *Miles-in-Trail* (MIT) restrictions and *Coded Departure Routes* (CDRs), to help manage the flow of air traffic [1].

- A STMP is a long range, strategic initiative that is implemented when a location requires special handling to accommodate above normal traffic demand. Currently, the FAA does not store flight plans more than two hours in advance, therefore STMP becomes important when the demand is unpredictable, providing long range planning capacity. In this context, being able to perform quick analysis of NAS-wide traffic with a two-hour notice is a very useful capability.

- A GDP is a traffic management procedure in which aircraft are delayed at their departure airport to manage the capacity demand at their arrival airport. GDPs are implemented to ensure that arrivals at an airport are manageable. They are also used to mitigate the impact of weather on the air traffic system.

- Introduced in the summer of 2006, AFPs are traffic management processes that identify constraints in the en route system. This is a significant new step in en route traffic management. The major objective of AFPs is to provide enhanced en route traffic management during severe weather events.

- MIT describes the number of miles between the aircraft departuring an airport, over a fix, at a certain altitude, through a sector or that is route specific. MITs are used to apportion traffic into a manageable flow as well as provide space for additional traffic to enter the flow.

- CDRs are a combination of coded air traffic routings and refined coordination procedures. They are used to mitigate the impact of weather or other events on the NAS.

In the current TFM system, controllers interpret data and then make decisions; in the future, computers might make short-term aircraft separation decisions and human beings will become more of a manager of the airspace [55]. This calls for the creation of decision support tools, which can help humans make more optimal decisions.

1.3 Related work

This section presents work related to the modeling of airspace and optimization of air traffic flow.

Investigations of traffic flow models have been studied intensively in the *Air Traffic Management* (ATM) community to mitigate the imbalance between the demand and capacity of the NAS. Functions in the existing NAS models include numerous aspects of the ATC system, such as the modeling of runway, airport and airspace capacity and operations [59; 27; 31; 53; 18; 7]. A simulation tool, FACET [14], is the first accurate NAS model capable of modeling system-wide airspace operations over the U.S. and providing an environment for the development and evaluation of advanced ATM concepts. This dissertation presents a framework for the modeling and optimization of en route air traffic flow. It is mainly related to NAS models for airborne *aircraft* and *airspace*, which are the two major aspects in the following literature review.

Numerous researchers have contributed to air traffic modeling and optimization, focusing on conflict detection and resolution for *aircraft*. In particular, Mueller et al. investigated the aircraft conflict resolution problem under specific flow management constraints, including meeting *miles-in-trail* and the required time of arrival at the next waypoint [52]. Bilimoria et al. studied aircraft conflict resolution with an arrival time constraint [13]. Paielli and Erzberger developed a conflict detection algorithm, focused on reducing operational errors [56]. Decentralized/distributed algorithms have been an important optimization tool for these models since they provide a procedural approach to the solution of these problems. For example, Resmerita et al. proposed a distributed multi-agent framework for conflict free navigation in a discrete environment with a two-phase approach (a conflict-resolution phase followed by an agent-accommodation phase) [62]; distributed algorithms that yield maximal solutions for the conflict resolution problem in multi-agent cases are presented in [61]. In the work [62; 61] and [33], arriving aircraft are guaranteed safe approach trajectories to choose from, and the available airspace and arrivals runway capacities can be efficiently utilized. Mao et al. studied the stability and performance of intersecting aircraft flows under decentralized conflict avoidance rules in a very unique and novel way [48]. Distributed robust receding horizon control was used by Kuwata et al. for multi-vehicle guidance [42]. Other approaches have focused on designing conflict-free trajectories using constant-speed, heading-change maneuvers [36]. In fact, numerous articles tackle the problem at the individual aircraft level, for example in the articles [14; 62; 61]. Dever et al. proposed a trajectory interpolation algorithm that performs a smooth transformation of vehicle maneuvers across a continuous range of boundary conditions while enforcing nonlinear system equations of motion as well as nonlinear equality and inequality constraints [26]. Devasia et al. used a token-based approach to design automation procedures for TFM [25], which in contrast works at a higher level of abstraction of the system.

Recently, there has been an increased interest in a specific modeling and optimization aspect of the *airspace*, usually known as the *Dynamic Airspace Configuration* (DAC). Among the DAC models, in the article [36], the capacity of the sector is determined by acceptable output-flow-rate; the proposed optimization algorithms are decentralized and can be used for sectors with multiple intersections. Kopardekar and Green designed the *Airspace Restriction Planner* (ARP) to predict and manage sector congestion problems in [40].

In general, most previous NAS models consider the behavior of every aircraft individually. For a NAS-wide study of the ATC system, extensive traffic forecast simulations (including all airborne aircraft) are too computationally expensive to include systematic investigations of traffic patterns that would lead to sector overload. As a result, a new class of traffic flow models has emerged from recent studies: *Eulerian* models, which are control volume based [49]. This is in contrast to *Lagrangian* models, which are trajectory-based and take into account all aircraft trajectories [14; 11].

Eulerian models have two main advantages over Lagrangian models. (i) They are computationally tractable, and their computational complexity does not depend on the number of aircraft, but on the size of the network problem. (ii) Their theoretic control structure enables the use of standard control and optimization methods to analyze the models.

The first Eulerian air traffic flow model was proposed by Menon et al. [49]. This model is powered by a discretized version of the *Lighthill-Whitham-Richards* (LWR) *partial differential equation* (PDE) [45; 63] and inspired by the Daganzo *Cell Transmission Model* [22; 23]. This Eulerian model [49] has since inspired several research groups to generate similar models that include a stochastic framework, leading to results in the expected sense [66; 70]. Two dimensional models [50] have also emerged, in the hope of better capturing flow patterns. An important characteristic of these approaches [49; 50; 66; 70] is the diffusion and dispersion that occur in the models. While this is not a problem in a stochastic framework (since the results are in the expected sense), it is more problematic for the deterministic models [49; 50] because this can potentially lead to aircraft losses or inaccurate predictions (this fact has been reported in the literature [9]). A first attempt to resolve these issues was proposed in a continuous time-continuous space model in [9], based directly on the LWR PDE. While this approach solves the diffusion problem, its computational tractability is disputable (since it depends on required space discretization), and the resulting optimization programs require heavy computations based on adjoint problems. Based on Eulerian modeling, a two-level control system for optimal TFM was recently developed [67], in which the inner-level control module takes in the optimal inflow and outflow commands generated by the outer control module as reference inputs and uses hybrid aircraft models to search for optimal trajectories.

In this dissertation, a discrete space, discrete time aggregate Eulerian-Lagrangian model of the airspace is proposed, which is control volume based (Eulerian) and takes into account the *Origin-Destination* (OD) information of the flights (Lagrangian) using a multicommodity flow formulation. This model has no diffusion and can be cast in the form of an integer linear dynamical system. It is computationally less expensive than previous models.

This work is related to existing *frameworks*, which are methods that rely on a flow model developed to solve TFM problems. Frameworks enable the use of control techniques, optimization tools in particular. Several frameworks to solve TFM problems have been proposed in the ATM community. In particular, Bertsimas and Stock Patterson [11] developed a seminal framework using a 0-1 integer programming method for the deterministic, multi-airport *Air Traffic Flow Management Problem* (TFMP) that addresses the capacity restrictions in the en route airspace. The TFMP was shown to be NP-hard (equivalent to job-shop scheduling [29]). This framework can be applied to many general network-based NAS models. The description of the state of aircraft is made through the dynamics of individual aircraft, therefore it is a Lagrangian model. This work is probably one of the most famous optimization frameworks used in the literature because of its generality relying on a *mixed integer linear program* (MILP), and the subsequent complexity analysis (NP-hardness) done for specific subcases of the problem.

Using the framework of aggregate air traffic flow models, optimization for TFM can usually be cast in the form of control and optimization of a networked system. The field of control and optimization of physical networks is a very wide area, for which numerous research efforts have led to the development of several methods dealing with the networks of distributed parameter systems. Several of them will be mentioned here because of their relevance to our work. In the context of traffic, networks of interconnected roads are modeled and studied in the recent book by Garavello and Piccoli [28] and can be used for the study of highway traffic flow. A variety of techniques exist for the optimization and control of physical networks. Frequency domain approaches have been used by Litrico et al. in the context of canal network control for the Saint-Venant equations [46], and provide useful control techniques when the underlying equations of flows are linear. They rely on the application of a spectral representation of these equations. Malaterre applied linear quadratic optimal control theory to the automatic control of two different eight-pool irrigation canals [47], using techniques similar to the ones presented in Section 3.3 of Chapter 3 in this thesis, which rely on the discretization of the underlying flow model. Several approaches have been developed to deal with nonlinear phenomenon present in physical networks, which often happens in transportation networks. In particular, a nonlinear output feedback method was studied in [8] for a compartmental network flow system. From a macroscopic point of view, Haut et al. modeled the junctions in a road network, which presents physically acceptable solutions for the capacity drop phenomenon in highway systems [32]. Coron et al. presented methods based on Lyapunov functions for a hydraulic application, namely the level and flow regulation in a horizontal open channel [21]. A decentralized nonlinear control approach was used in [41] for fluid flow networks, where actuator valves and flow rate sensors are collocated in individual branches and do not exchange information. A similar model was used for optimal control of supply networks in [39].

Although the study in this dissertation focuses on en route air traffic, the model proposed in this work can be extended to take into account airports; as a result, the solutions generated by the optimization algorithm based on the model can include the *Ground Delay Program* (GDP). Besides, an en route air traffic model is useful to mitigate the impact of the shortfall of airport capacities, and it can help better use the airspace for airborne aircraft to make the entire NAS more responsive to unanticipated changes in system capacities.

1.4 Contributions

The contributions of this dissertation are now summarized:

- **A control theoretical model of the NAS.** This dissertation presents a new model called the *Large-capacity Cell Transmission Model*, which is also known as CTM(L). The following is a summary of the characteristics of the CTM(L).
  - The terminology CTM(L) is in reference to the seminal Daganzo *Cell Transmission Model* (CTM), which was commonly used for highway traffic [22; 23]. The term "large-capacity" refers to the fact that there is no capacity imposed on a single cell of the network, but on a set of cells (whose sizes are defined by time, in contrast to CTM which is defined by space/distance) corresponding to a sector. This is fundamentally different from highway traffic, and specifically addresses the needs of air traffic control.
  - In contrast to most other existing Eulerian models, the model proposed in this dissertation includes Lagrangian features: despite the aggregation, it takes into account the *Origin-Destination* (OD) information of the flights, which eliminates the splitting and diffusion problems existing in some Eulerian models [49; 50]. For this purpose, a multi-commodity flow structure [20] has been included in the model to enable the description

of traffic path by path. This is one of this work's fundamental contributions to NAS modeling. The multicommodity network model incorporates the topology of the NAS and the resulting flow patterns. Therefore, the model is physically meaningful and tractable for control and optimization because it already incorporates routing information in the flow pattern.

  - The linear time invariant dynamics of the CTM(L), in which the transition matrix is nilpotent, greatly facilitates the design (optimization) and analysis of the model.
  - The CTM(L) is scalable: the granularity of the model is dependent on the time step (one minute in this study), which can be changed to different time scales and can represent models at different levels, e.g., from the sector level to the center level of the NAS.

- **Parameter identification of the multicommodity network.** The CTM(L) is developed based on a multicommodity network model of the NAS. This dissertation presents a sequence of techniques which automatically identify the parameters of the network, using recorded *Aircraft Situation Display to Industry* (ASDI) and *Enhanced Traffic Management System* (ETMS) data. In contrast to numerous aforementioned articles, this work makes extensive use of massive air traffic data sets. Once being formulated, the mathematical model is an input to a program, which specifically constructs the model for the entire NAS using the air traffic data. It is obviously the key to providing the air traffic control community with a working tool.

- **Model validation.** The CTM(L) was successfully validated against recorded ASDI/ETMS data for a whole year and for the whole NAS, i.e. 20 continental ARTCCs. It is an important feature of this work, compared to other models proposed in the literature. The accuracy of this model (in particular its predictive capabilities) has been assessed in practice.

- **Comparison of the model to three other existing models.** In addition to developing the CTM(L), three other aggregate Eulerian models are also presented for en route (high altitude) air traffic flow: (i) a modified version of the Menon model [49] adapted to fit a general network topology, (ii) a new application of the Lax Wendroff scheme to a PDE model developed in [9], and (iii) a two-dimensional Menon model [50] at a NAS-wide level. These four models have been implemented and their predictive capabilities are compared using the same benchmark problem for fairness of the comparison. This study is the first NAS-wide implementation of the four aforementioned models and the first comparison of their respective performance on the same benchmark scenario.

- **Formulation of traffic flow optimization problems.** Using the CTM(L), this dissertation presents a framework for the formulation of optimization problems for Traffic Flow Management. This framework can be applied to general aggregate traffic flow models.

- **Tractable optimization algorithm.** An optimization algorithm based on a dual decomposition method is designed for the Large-capacity Cell Transmission Model, which makes a NAS-wide Traffic Flow Management problem with billions of variables and constraints solvable in real-time. This contribution algorithmically solves the dilemma of accuracy versus tractability often faced in modeling. Because of the efficiency of the method, we *can* afford to work with massive models of air traffic, which provide accurate models of fine grained features of the flows.

- **Implementation of the CTM(L).** The CTM(L) is integrated in FACET [14], a software developed at NASA Ames Research Center. This contribution merits to be mentioned: FACET is currently becoming one of the reference air traffic control software systems in use at NASA and FAA. The architecture integration required to assemble the optimization proposed in this work and the software is nontrivial, and constitute a technological achievement, performed jointly with Metron Aviation that underlines the usefulness of the work.

- **Implementation of the optimization algorithm.** The optimization algorithm developed in this study is implemented in software to solve NAS-wide Traffic Flow Management problems on a global scale, which is the goal of this dissertation.

1.5 Organization of this work

This dissertation is organized as follows. In Chapter 2, a Large-capacity Cell Transmission Model is derived. The validation of the model is presented in the same chapter. In Chapter 3, the performance of the model is compared to three existing models. This includes the comparisons of their accuracy and of their computational efficiency. Optimal control algorithms are developed for two of the models. In Chapter 4, based on the Large-capacity Cell Transmission Model, a dual decomposition algorithm is developed for NAS-wide Traffic Flow Management problems, which is formulated with billions of variables and constraints, involving approximately 6,000 aircraft over the course of one to four hours. Chapter 5 briefly summarizes the work of this dissertation and future directions.

Chapter 2

Large-capacity Cell Transmission Model

This chapter presents the *Large-capacity Cell Transmission Model*, shortened to CTM(L), for en route air traffic flow, which serves as the underlying flow model for the rest of this dissertation. This model is control volume based (Eulerian) and takes into account the Origin-Destination information of the flights (Lagrangian). The construction of a graph-theoretic multicommodity network model, which serves as the Lagrangian routing of these flows, is outlined first. The evolution of air traffic flow on this graph is modeled as a discrete time dynamical system. The forecast capabilities of the model are validated for the entire NAS using ASDI/ETMS data. In particular, it is shown that a very important metric for TFM (aircraft count) is reproduced adequately by the model, which serves as a validation for it.

2.1 Graph-theoretic model

This section proposes a method to build a network structure model that serves as a routing network for the flows (Lagrangian component of model), and to construct a dynamic aggregate model of the traffic on this network (Eulerian component), which describes the evolution of the flows on the network.

The objective of automated model building is to produce a method that constructs a graph-theoretic multicommodity model of air traffic flows directly from track data (ASDI/ETMS data files, in the present case). In this research, several pattern recognition methods have been implemented to automatically build the components of a multicommodity network model of the observed flows. The suite of algorithms investigated includes a variety of techniques, some of which rely purely on flight tracks, others use additional information that can be extracted from ASDI/ETMS data (e.g. flight plan data). In general, applying canned algorithms to network flow model building problems does not provide satisfactory results because of the specificity of high altitude traffic. This fact led to the approach outlined later in this section.

These investigations are summarized below as an illustration of the difficulty of automated model building.

1. *K-Means* [5]. The K-means algorithm groups data into clusters defined by "cluster centers" or "cluster means." A "cluster center" is the mean of the data points in that cluster. The algorithm assigns data points to clusters by finding the nearest cluster mean and assigning the data point to that cluster.

2. *Generalized Principal Component Analysis (GPCA) Algorithm* [75; 76]. GPCA is an algebraic geometric approach to the problem of estimating clusters from sample data points. GPCA automatically determines the number of clusters, and unlike K-means [5], it does not need this information *a priori*.

3. *Flight plan based algorithms*. A flight goes from a departure airport to an arrival airport

by traveling through a set of fixes (waypoints). The trajectories are classified based on sequences of waypoints. Waypoint-based flow pattern classification can be considered “noise-free”, since the trajectory of a flight is defined by strings of characters. Similarly, jetways are also used as a classification criteria. Because jetways are similar to highways, they act as guidelines that flights should follow. Flights using the same jetway are supposed to use similar trajectories.

The list of such algorithms can be extended at will, but they do not perform well in practice. There are several explanations for this fact. (i) The mathematical optimum leading to the convergence of these methods is not necessarily a relevant metric for air traffic; in other words, a suboptimal solution might be physically more relevant than the optimum because the optimized cost function does not reflect the patterns being identified. (ii) The nature of the data makes it impossible to classify flows based on proximity, even for classification criteria involving strings (as is the case for flight plan information, which consists of an enormous number of acronyms).

In addition to these general considerations, specific reasons prevent the above algorithms from being applicable. (i) The K-means algorithm requires *a priori* knowledge of the number of clusters, which are not known in the present case. Furthermore, it is extremely sensitive to the initial guess, which makes it hard to use in an automated manner. (ii) Waypoint-based classification is inappropriate because of the extremely large number of different waypoint acronyms in the NAS. Even though this data is “noise-free” (defined by strings), its size is prohibitive for the present study. (iii) Jetway-based classification is not applicable, since ASDI/ETMS data does not provide the location of the merge point of an aircraft onto a jetway, leading to the well-known underconstrained OD estimation problem in highway traffic [35]. All of these difficulties are a motivation for the

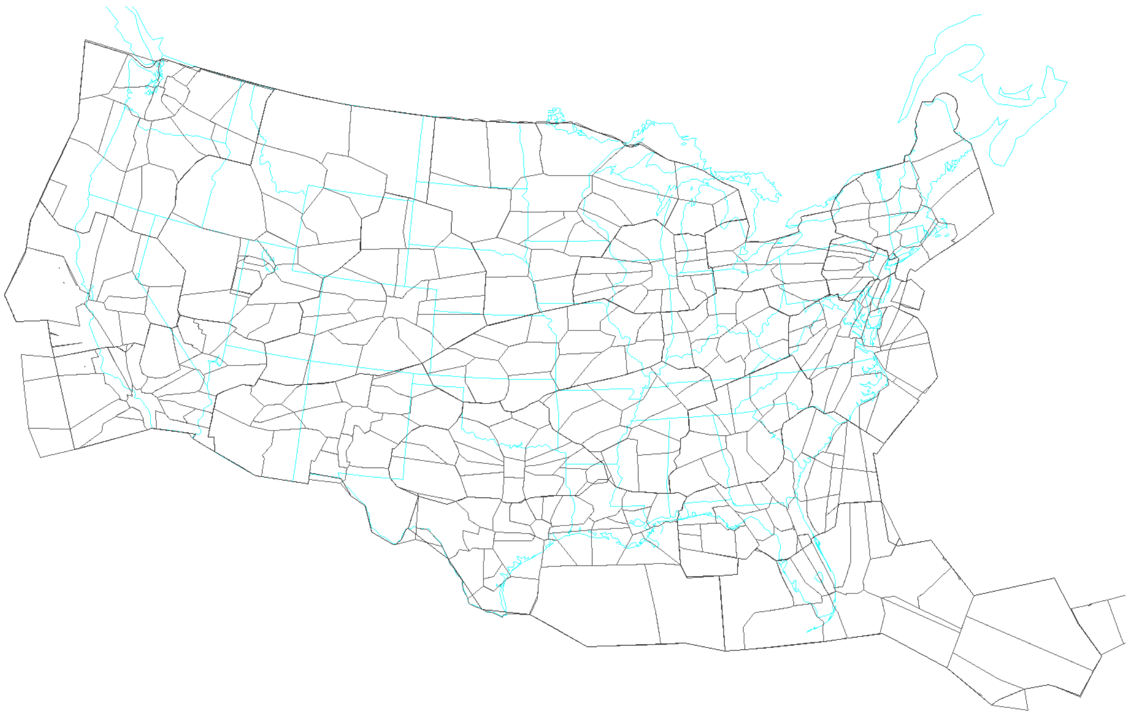

Figure 2.1: Map of the airspace considered in this study: the entire continental NAS including 284 high-altitude sectors in the U.S. Figure obtained using FACET.

method presented in the next section.

2.1.1 Definitions

The system to be modeled is the continental en route U.S. airspace, which is the size of 20 ARTCCs including 284 high-altitude sectors, with altitudes of 24,000 feet and above (Figure 2.1) [14]. All non-military flights traveling through the considered airspace are included in the scope of this work. The ASDI/ETMS data used in this study, provides the position and altitude of all airborne aircraft in the U.S. every minute. Additional information related to flight plans or other flight parameters, such as speed and heading, are also provided in the data, but are not used to build the present aggregate model.

2.1.2 Construction of the graph

As will be shown in the next section, the model must be sufficiently fine, so that flights following different flow patterns within a sector can be distinguished. However, the granularity must not be too small in comparison to the size of a sector for the model to remain meaningful and tractable.

Vertices (nodes)

The graph representing the flows is constructed as follows: two *vertices* are created at the boundary of each pair of neighboring sectors. For any two neighboring sectors s_1 and s_2, the vertices at the boundary of s_1 and s_2 are denoted by $V_{\{s_1,s_2\}}$ and $V_{\{s_2,s_1\}}$. Vertex $V_{\{s_1,s_2\}}$ will be used to represent flights going from s_1 to s_2, while vertex $V_{\{s_2,s_1\}}$ will be used to represent flights going from s_2 to s_1. The computation of the exact physical location of the vertices will be described at the end of this section; assume for the moment that each vertex $V_{\{s_1,s_2\}}$ is located at a point of the boundary of sectors s_1 and s_2. Note that $V_{\{s_2,s_1\}}$ is not necessarily located at the same point as $V_{\{s_1,s_2\}}$. The physical location of the vertices is important to represent the graph on a map, but it has no influence on the topology of the graph itself.

Links (edges)

For any sectors s_1, s_2 and s_3, if s_1 and s_2 share a boundary and if s_2 and s_3 are neighbors, two *directed links* are created: one from vertex $V_{\{s_1,s_2\}}$ to vertex $V_{\{s_2,s_3\}}$ and one from vertex $V_{\{s_3,s_2\}}$ to vertex $V_{\{s_2,s_1\}}$. In the rest of this work, the term *link* refers to a directed link [4]. Figure 2.2 illustrates the concept of a link.

Interface between considered region and rest of the airspace

The region of the airspace considered in the model must be connected, for practical reasons. Note that it does not need to be convex. Let us denote S as the set of coordinates (latitude, longitude) of the points that belong to the considered region. A point in the airspace belongs to the considered region if its coordinates are in S and if its altitude is above 24,000 feet. The considered region must be interfaced with the airspace around it. Therefore, in the model, a "sector" called *low* is created for the purpose of this study, consisting of the points of the airspace whose coordinates are in the set S and whose altitude is below 24,000 feet. An additional "sector" labeled *none* is created, consisting of the points of the airspace whose coordinates are not in the set S. In practice, the portion of the airspace labeled *none* corresponds to the sectors surrounding the region of interest (see Figure 2.1). The appellation "sector" for these two regions of airspace is not understood in the ATC sense. Instead, it is used to indicate that, in the same manner as described above, vertices are created at the boundary of these additional "sectors" and the sectors in the regions considered in the study. These vertices, and the corresponding links, are used to take into account climbing, descents, and flights entering or exiting the considered region. Figure 2.2 shows a few examples of vertices and links. Note that not all vertices and links are represented in this figure.

Multicommodity network

For a complete network model including the whole continental NAS of the US, a multicommodity flow structure [20] is used in CTM(L). Flights are clustered based on their entry-exit node pairs (origin-destination pairs) in the network. Each pair corresponds to a *path* consisting of

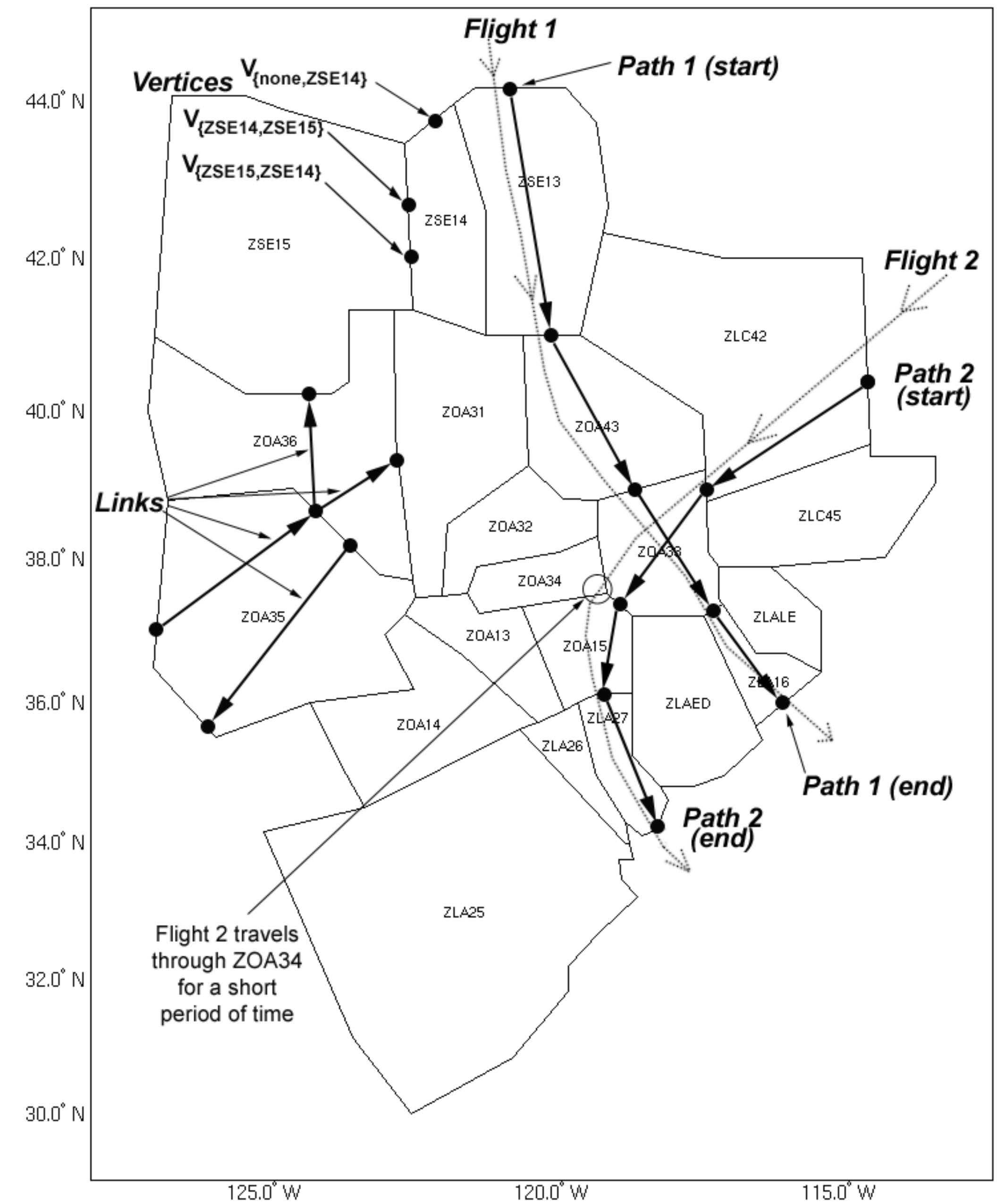

Figure 2.2: Illustration of the notion of vertices, links, trajectories and paths used for the construction of the network model.

links between these nodes. [1] If two or more paths have one link in common, this link will be duplicated, using a multicommodity flow structure. Figure 2.3 illustrates decoupled multicommodity network models for several destination airports. In fact, these decoupled multicommodity networks are trees, with air traffic flows originating from the airports in the continental US. Note that only a portion of the origin airports are shown in the figure for clarity. An aggregation of the trees for all destination airports provides a complete (NAS-wide) network model.

A complete NAS-wide network is shown in Figure 2.4.

Size of the network model and computational cost

Using the method described above, in the multicommodity network model, there are 284 high-altitude sectors, 1598 links, and 1841 nodes. It takes 102 hours to extract the flight information (latitude, longitude, flight time, etc.) and build a database of links and nodes information from the ASDI/ETMS data for a whole year. It takes five minutes to build the multicommodity network model on a desktop computer with a 1.4GHz CPU, 1GB RAM running Linux.

2.1.3 Classification of trajectories and estimation of link travel times

The graph constructed above is used to encode the paths followed by streams of flights traveling through the considered region of the airspace. In order to identify the behavior of these streams of aircraft on the links, their trajectories must be assigned to a set of links of the graph. In this identification phase, the position of all airborne aircraft is provided by ASDI/ETMS data, from which sector information about all flights can be deduced.

[1] In the present study, a path is uniquely defined by an origin-destination pair; it is straightforward to extend this stucture to include multiple paths between an origin-destination pair as long as the flight plan contains enough information to determine which path the flight takes.

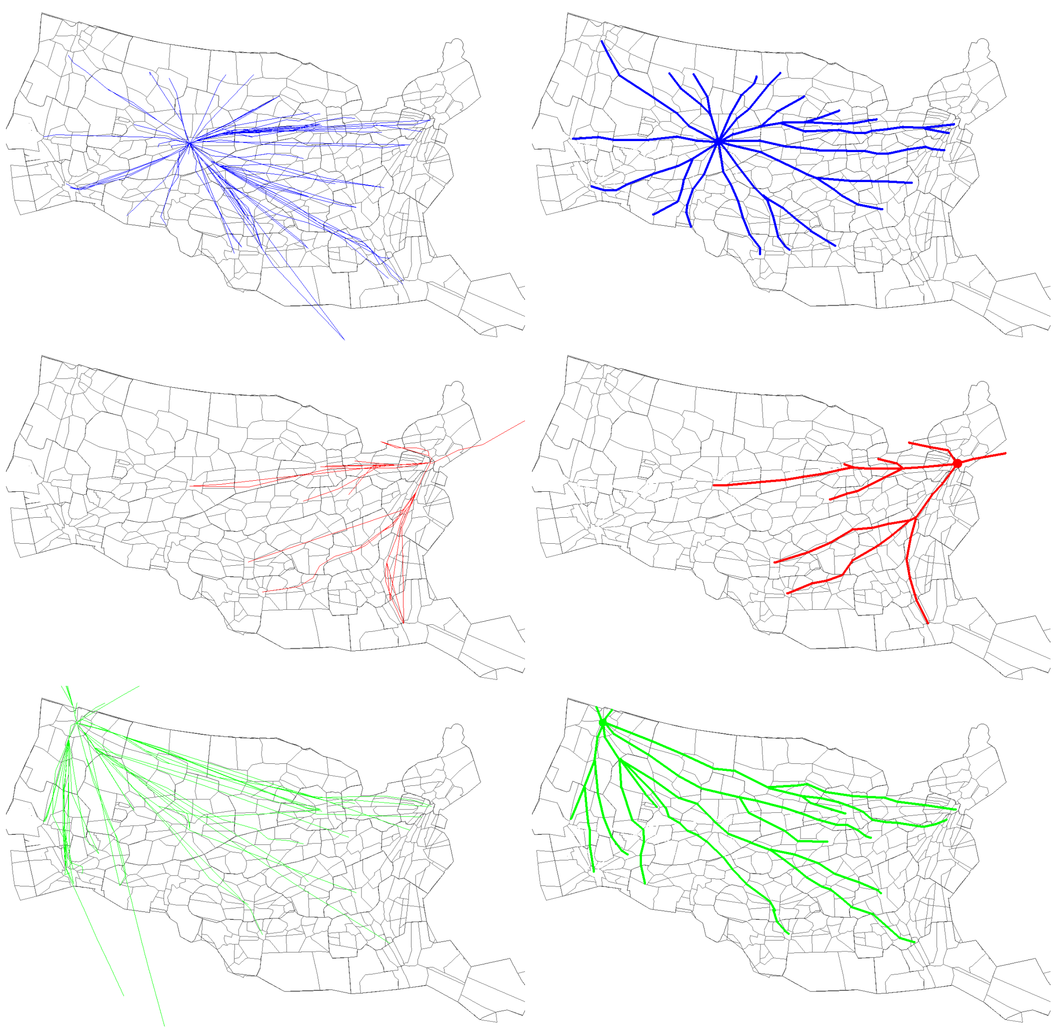

Figure 2.3: Illustration of decoupled multicommodity network models by destination for airports (DEN, LGA, SEA). An aggregation of the trees corresponding to destination airports provides a complete multicommodity network level model. **Left**: recorded flight tracks (a few data points only shown); **Right**: corresponding air traffic flow trees constructed by the model to incorporate knowledge of the destination in the model.

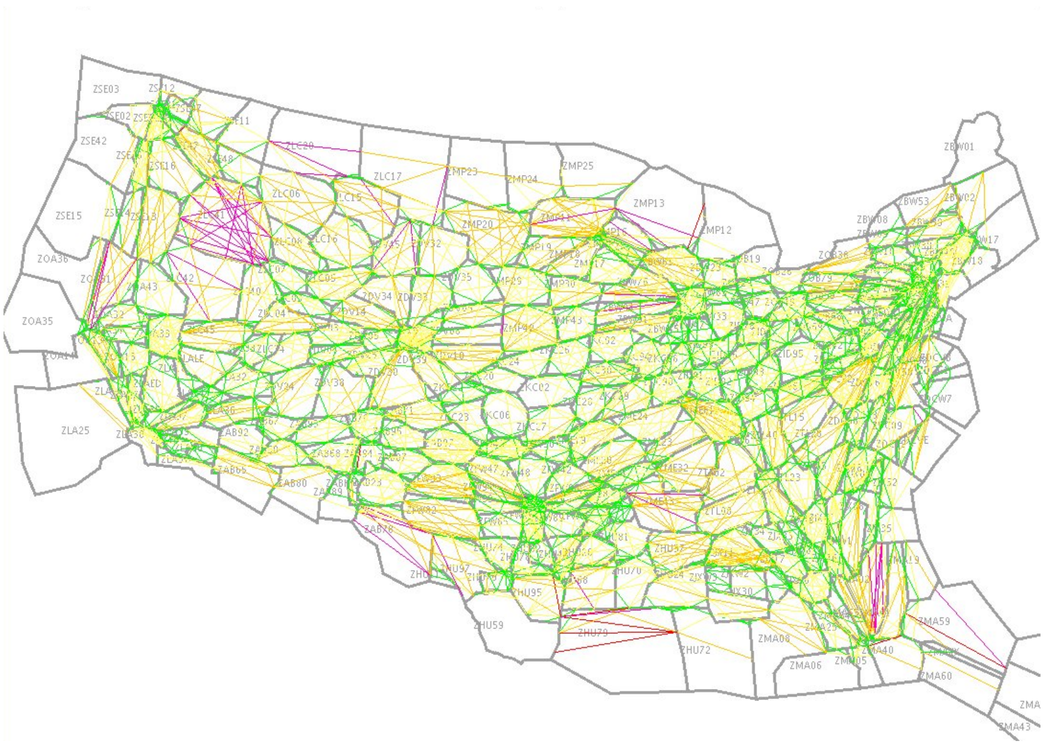

Figure 2.4: A complete continental NAS-wide network model.

Fundamental assignment rule

The fundamental assignment rule is as follows: when a flight crosses the boundary between two sectors s_1 and s_2, coming from s_1 and going to s_2, it is assumed in the model that the flight passes through a unique vertex $V_{\{s_1,s_2\}}$. The time at which the flight is modeled to pass through that vertex is the time at which it crosses the boundary between the corresponding sectors. It is then assumed in the model that the flight travels from vertex to vertex, using links between these vertices. The travel time of the flight on each link can easily be calculated from the times at which the flight passes each vertex. The sequence of links used by a flight is referred to as a *path*. In Figure 2.2, *path 1* is the representation of *flight 1* in the model, based on the fundamental assignment rule.

Exception to the fundamental assignment rule

A refinement to the fundamental assignment rule is introduced, in order to take into account flights that stay in a given sector (s_2) for a short period of time, while traveling from a sector (s_1) to another sector (s_3), in case sectors s_1 and s_3 are neighbors. In that case, the flight is usually handed off by the controller in s_1 directly to the controller in s_3, after the controller in s_2 has been informed by the controller in s_1 that the aircraft will be in s_2 for a short period of time. Therefore, this particular aircraft should not be represented as being in sector s_2, since it does not significantly increase the workload of the controller in s_2. If the sectors s_1 and s_3 are neighbors, the flight is modeled passing through vertex $V_{\{s_1,s_3\}}$ instead of vertices $V_{\{s_1,s_2\}}$ and $V_{\{s_2,s_3\}}$. The definition of the time at which the flight is modeled passing through vertex $V_{\{s_1,s_3\}}$ is not as straightforward as for the case of the fundamental assignment rule. In the model, the flight is assumed to pass through vertex

$V_{\{s_1,s_3\}}$ at the time instant corresponding to the average of the time at which it crossed the boundary of sectors s_1 and s_2, and the time at which it crossed the boundary of sectors s_2 and s_3. This rule is applied if a flight stays in a sector for less than two minutes. Note that if the sectors s_1 and s_3 are not neighbors, the sequence of vertices $V_{\{s_1,s_2\}}$ and $V_{\{s_2,s_3\}}$ is maintained in the model. In Figure 2.2, *path 2* is the representation of *flight 2* in the model, based on the assignment rule described in this paragraph. Namely, *flight 2* stays in ZOA34 for a short period of time, and the sectors before and after ZOA34 in the trajectory of *flight 2*, ZOA33 and ZOA15, are neighbors. Therefore, *flight 2* is represented in the model traveling through the following sequence of vertices: $V_{\{\text{ZLC42,ZOA33}\}}$, $V_{\{\text{ZOA33,ZOA15}\}}$, $V_{\{\text{ZOA15,ZLA27}\}}$. If only the fundamental rule was applied, the sequence of vertices of *flight 2* would be: $V_{\{\text{ZLC42,ZOA33}\}}$, $V_{\{\text{ZOA33,ZOA34}\}}$, $V_{\{\text{ZOA34,ZOA15}\}}$, $V_{\{\text{ZOA15,ZLA27}\}}$.

Interpolation required to decrease error on travel times estimation

ASDI/ETMS data provides aircraft positions every minute. Interpolation is needed to reduce the error on the boundary crossing times and locations. Without interpolation, the error on the crossing time (i.e. the time when the flight is modeled to pass through a vertex) can be as large as 59 seconds, which leads to a possible error on the travel time estimation through a link as large as one minute. Interpolation is based on the following reasonable assumptions: (i) The speed of an aircraft remains constant between the two records, which are one minute apart. (ii) It flies in a straight line during that same time interval. The computation of sector boundary crossing locations and times requires the implementation of a procedure that determines the point of the trajectory's intersection between two consecutive flight data records, which is a segment and the boundary of two sectors.

Determination of the physical location of vertices

Once the sequence of vertices is determined for all flights, the exact physical location of each vertex can be computed. To determine the location of vertex $V_{\{s_1,s_2\}}$, all flights passing through that vertex, coming from sector s_1 and going to sector s_2, are considered, and the points at which each of them crossed the boundary of s_1 and s_2 are computed. The location of vertex $V_{\{s_1,s_2\}}$ is the center (average) of those points of boundary crossing. Note that the center (average) is not taken in the plane, but along the unfolded boundary of s_1 and s_2.

2.1.4 Travel time analysis

For each link of the graph, the flight times for a full year (Oct. 1st, 2004 to Sept. 30, 2005) of ASDI/ETMS data are aggregated. The mean of this distribution is computed, and its value is chosen to represent the "time length" of the link, i.e. the aggregated travel time along the link. Figure 2.5 shows a typical distribution of the travel time. The expected travel time of a flight through a link is used to determine the length of the link. As will be seen in the subsequent sections describing the proposed CTM(L), each link is divided into several *cells*. The number of aircraft in a cell will be used as a coordinate of the state in the model derived below. In the present setting, cells correspond to one minute of flight time.

Figure 2.6 shows another distribution of the travel time, with two dominant peaks. In this case, the link (ZOA32-ZOA43-ZLC42) is split into two links, with link lengths defined by the travel times corresponding to the two peaks.

In the present derivation of the graph-theoretic model, the mean of the distribution of the flight times is used as the "length" of the link. In practice, there are several variations that should

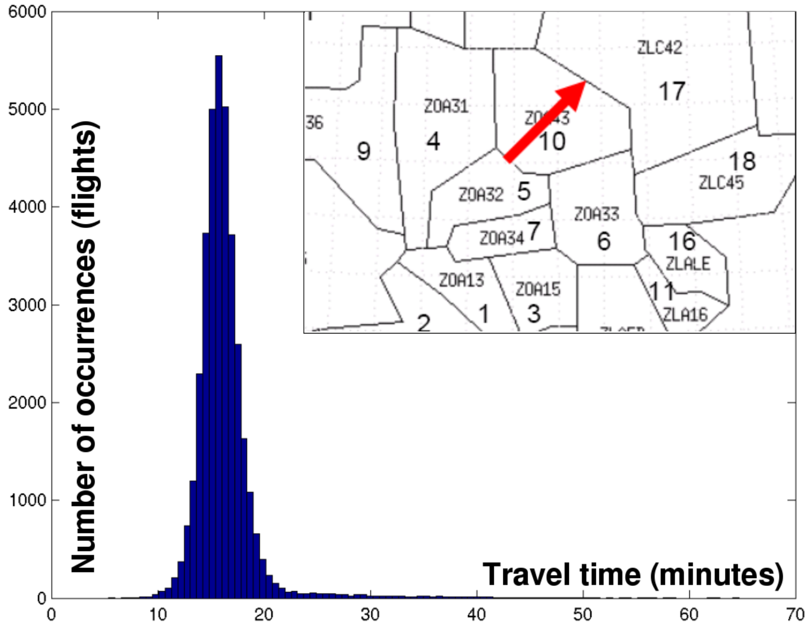

Figure 2.5: Distribution of travel time on one link (ZOA32-ZOA43-ZLC42). One full year of aggregated data.

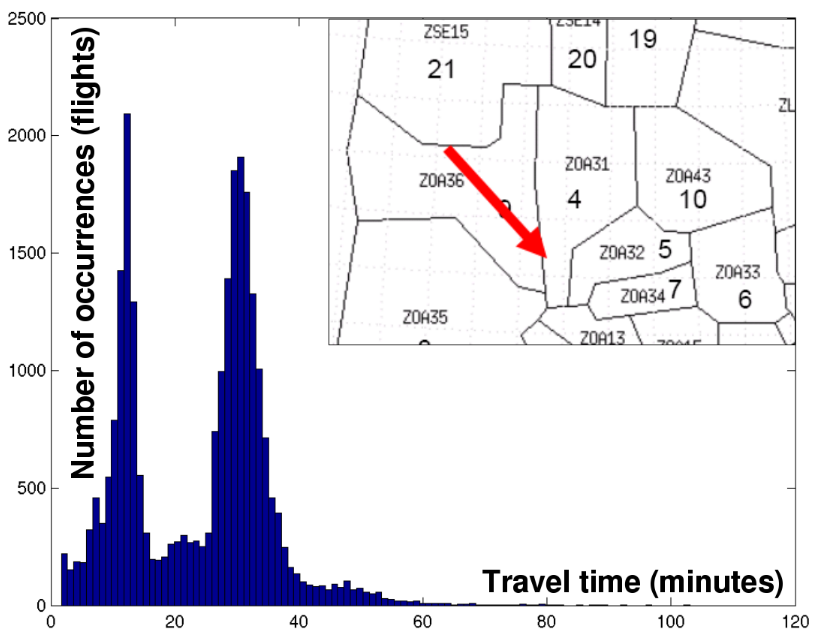

Figure 2.6: Distribution of travel time on one link (ZSE15-ZOA26-ZOA31). One full year of aggregated data.

be taken into account. For example, the length (travel time) of a link generally changes, therefore, a time-varying graph model can be derived based on the time-varying link length. Figure 2.7 shows the distribution of travel time on one link (ZOA31-ZOA13-ZOA14) in different months of a year. Figure 2.8 shows the mean and standard deviation of the travel time of the link ZOA31-ZOA13-ZOA14 for different months in a year. The means of the travel time of different months are similar, but the variances differ for different months. The analysis of the variations of the travel time is outside the scope of this work. Another way to capture the time-varying feature of the travel time would be to perform the travel time identification with the "clustering" methods, inspired by the work of Hoffman et al. [34].

Depending on the objectives of modeling, different types of the travel time distribution can be used. For example, if the priority interest is in building a stochastic air traffic model, the type of distribution will be one of the most important characteristics. When building a time-varying model, seasonal, monthly, weekly, daily and hourly distribution will be more important.

2.2 Derivation of the new Eulerian-Lagrangian model

2.2.1 The Large-capacity Cell Transmission Model or CTM(L)

In this section, the new Eulerian-Lagrangian model, the *Large-capacity Cell Transmission Model*, or CTM(L), is developed. This model is inspired by the *Lighthill-Whitham-Richards* theory [45; 63], and by the Daganzo Cell Transmission Model [22; 23], which is commonly used in highway traffic modeling. The CTM(L) is assembled with the graph-theoretic multicommodity network, which is constructed from historical ASDI/ETMS traffic data handling the Lagrangian routing of the flow, as described in the previous section. The model is reduced to a linear time

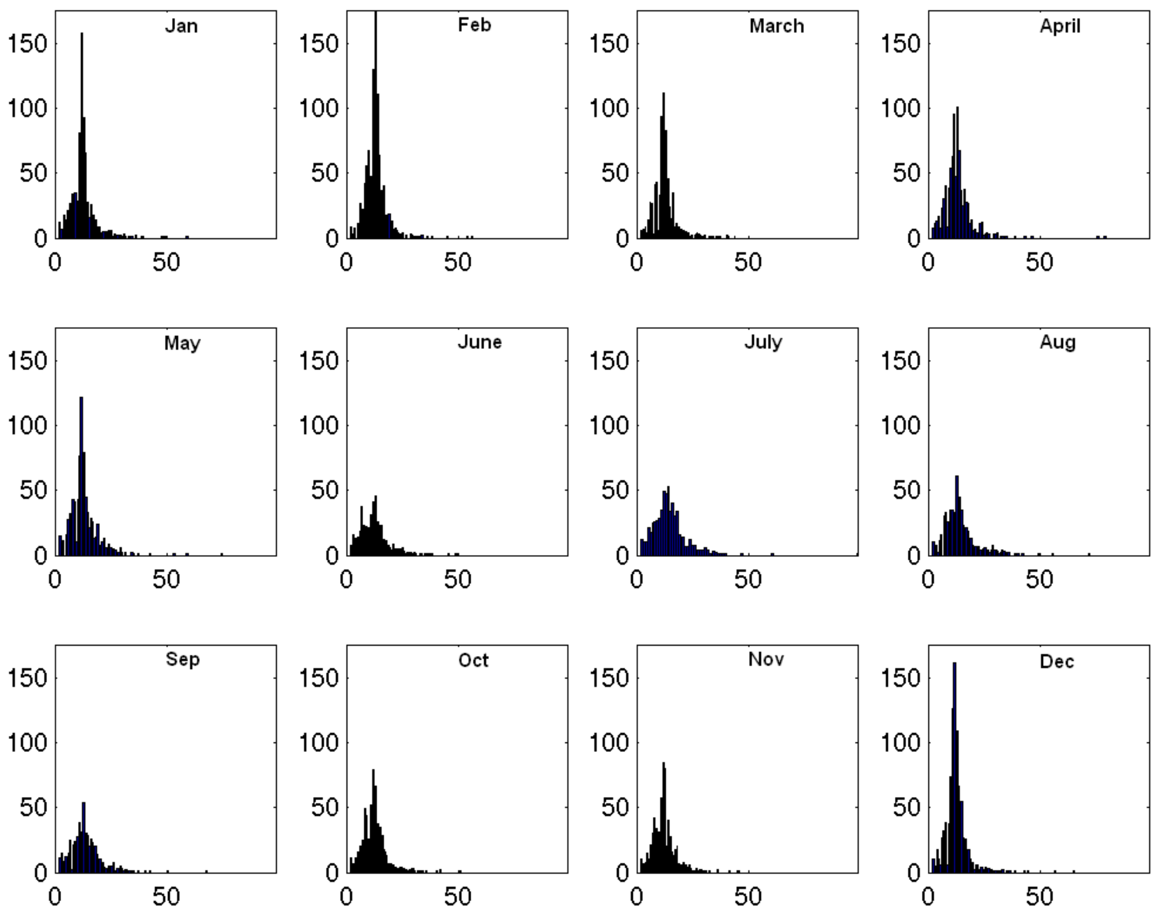

Figure 2.7: Distributions of travel time on one link (ZOA31-ZOA13-ZOA14) in different months of a year. Horizontal axis: travel time, in minutes through the link; Vertical axis: number of occurrences of the travel time.

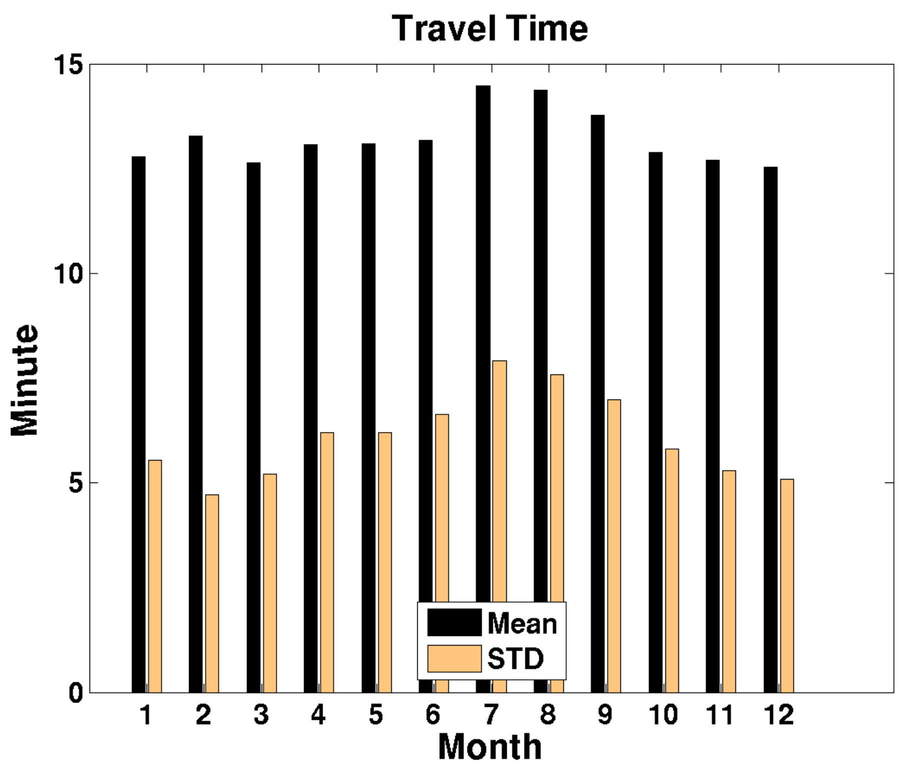

Figure 2.8: Mean and standard deviation of travel time on one link (ZOA31-ZOA13-ZOA14) in different months of a year.

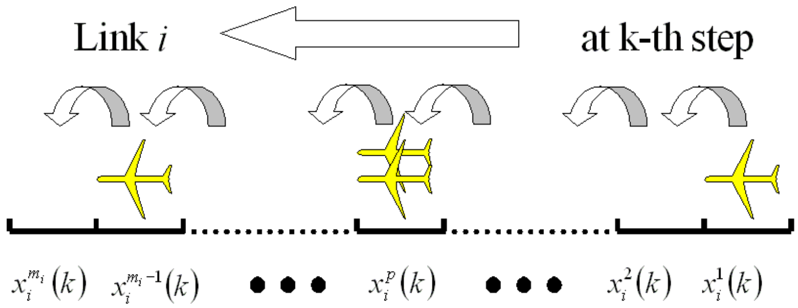

Figure 2.9: Illustration of the CTM(L) at link level: everywhere inside the link, $x_i^{p+1}(k+1) = x_i^p(k)$, unless some control action is applied.

invariant dynamical system for this network topology, in which the state is a vector of aggregate aircraft counts. The controlled input to the model is delay control, which can take several forms: speed change, *vector for spacing* (VFS), *holding pattern* (HP), etc. These different forms will correspond to the different time increments in which they are applied.

Link level model

In the CTM(L) derived next, a *link* is used in the graph-theoretic sense (i.e., an edge of a graph [4]). When the flights are clustered based on the entry-exit node pairs in a specified sector, a *link* can also be viewed as the connection between the entry point and the exit point incident to this sector.

1. *Assumptions*. To formulate the model at a link level, the following assumptions are made:

   (a) Each link is modeled as a directional edge. In Figure 2.9, the arrow represents the flow direction. In other words, the graph is unidirectional.

   (b) All aircraft in a given link fly at an aggregate speed. This speed can be obtained by aggregating the speed (obtained from the ASDI/ETMS data) of all aircraft following this link.

(c) The number of cells in one link is scaled by the steps of expected travel time. In the implementation, one minute is taken as a unit time step. For example, if it takes around 12 minutes for an aircraft to fly across sector ZOA33, following one particular link, then this link would be divided into 12 segments, called *cells*. The choice of the cell length (time discretization) is arbitrary of course. In the model, a link indexed by i has m_i cells. The smaller the time step, the more accurate the model, but the computation is more complex.

(d) At the link level, only aircraft whose altitude is above 24,000 feet are taken into account for the calculation of aircraft count. This choice is arbitrary and can be adjusted to any user-defined level.

(e) The control strategy (based on the application of delay to aircraft) is mainly used as the controlled input to the model, which can be implemented in many forms: speed change, VFS as well as HP. It is supposed to be applied in one minute time increments.

(f) The model is *deterministic*. No statistical factor, such as the impact of the weather, is taken into consideration at this stage. Note that it could be added later using a stochastic framework [66; 70].

(g) In this model, all values, including the states, inputs and outputs, should be integer. This might increase the complexity of the computations or analysis, but provides higher accuracy.

2. *Definitions*. The following definitions are used in this work.

(a) The state of link i at time k is given by $x_i(k) := [x_i^{m_i}(k), \cdots, x_i^1(k)]^T$, an $m_i \times 1$

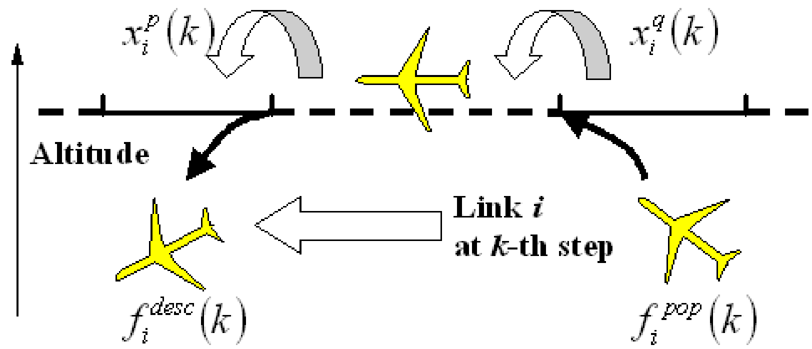

Figure 2.10: Illustration of descent and climb inputs to the model, where $x_i^{p+1}(k+1) = x_i^p(k) - f_i^{\text{desc}}(k)$, and $x_i^{q+1}(k+1) = x_i^q(k) + f_i^{\text{climb}}(k)$ are satisfied, unless some control action is applied.

vector, whose element $x_i^p(k)$ represents the aircraft count in cell p of link i at time k. For example, in Figure 2.9, $x_i^p(k) = 2$, because there are two aircraft in the p-th cell at time instant of k. m_i is the number of cells in this link.

(b) The forcing input, $f_i^{\text{in}}(k)$, is a scalar input that models the entry count from the boundary of the domain of interest into this link during a unit time interval from k to $k+1$. For example, if there are five aircraft entering link i from $k = 3$ to $k = 4$, then $f_i^{\text{in}}(3) = 5$.

(c) The descent input, $f_i^{\text{desc}}(k)$, is also a scalar input, which denotes the number of aircraft leaving link i during a unit time interval from k to $k+1$, because of the descent to a lower flight level. For example, in Figure 2.10, $f_i^{\text{desc}}(k) = 1$.

(d) The climb to en route input ("climb input," in short), $f_i^{\text{climb}}(k)$, is another scalar input which means the number of aircraft entering link i during a unit time interval from k to $k+1$, because of the climb from a lower flight level. Also, in Figure 2.10, $f_i^{\text{climb}}(k) = 1$.

(e) The control input, $u_i(k)$, is an $m_i \times 1$ vector, representing delay-based control. This type of control is common in the presence of congestion: traffic managers will typically use delays as a way of controlling air traffic flows in the en route airspace when aircraft are

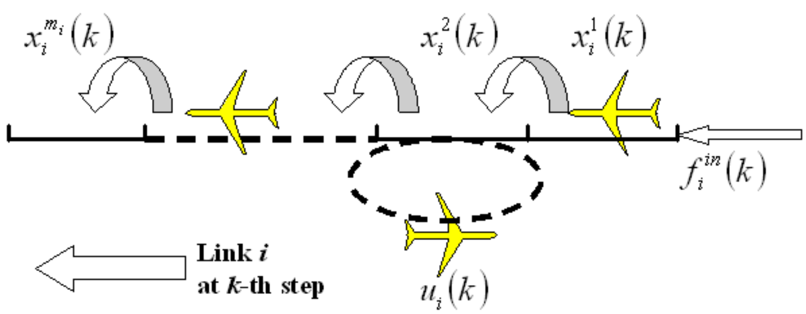

Figure 2.11: Illustration of delay-based control to the model (which can, for example, model holding pattern control, vector for spacing, or ATC prescribed deceleration), where $x_i^3(k+1) = x_i^2(k) - u_i(k)$, and $x_i^2(k+1) = x_i^1(k) + u_i(k)$, unless another control action is applied.

already airborne. The p-th element denotes the number of aircraft under delay control in the p-th cell of link i at time instant k. In this model, the cycle of increment delay is one minute. In Figure 2.11, one type of delay control, a holding pattern control, is taken as an example, where $u_i(k) = [0, \cdots, 0, 1, 0]^T$, because there is only one aircraft under holding pattern control in the second cell of link i at time k.

(f) The output, $y_i(k)$, is the aircraft count in link i in a user-specified set of cells at time step k, e.g. the total number of aircraft in all cells of this link at time step k. For example, in Figure 2.9, $y_i(k) = 4$.

3. *Model Description*. A *deterministic*, *Linear Time Invariant* (LTI) model for link i is developed in state space form as follows:

$$x_i(k+1) = A_i x_i(k) + B_i^{\text{in}} f_i^{\text{in}}(k) + B_i^{\text{desc}} f_i^{\text{desc}}(k) + B_i^{\text{climb}} f_i^{\text{climb}}(k) + B_i^u u_i(k) \quad (2.1)$$

$$y_i(k) = C_i x_i(k), \quad (2.2)$$

where A_i is called a system matrix, and is a m_i by m_i nilpotent matrix with 1's on its super-diagonal. The forcing input matrix, $B_i^{\text{in}} = [0, \cdots, 0, 1]^T$, is a $m_i \times 1$ vector. The descent input matrices, B_i^{desc}, and the climb input matrix, B_i^{climb}, are both $m_i \times 1$ vectors, in which 1's mean that aircraft will leave from the p-th cell of link i for descent or enter the q-th cell of the same link because of climbing. The controlled input matrix, B_i^u has a dimension of $m_i \times m_i$, containing all 0's elements except with 1's on its diagonal and -1's on its super-diagonal. The non-zero elements of the $m_i \times 1$ vector C_i correspond to the cells in the user-specified set, and are equal to 1's.

In fact, three inputs, $f_i^{\text{in}}(k)$, $f_i^{\text{desc}}(k)$ and $f_i^{\text{climb}}(k)$, can be incorporated into one vector. Then, equation (2.1) can be rewritten in a more compact form:

$$x_i(k+1) = A_i x_i(k) + B_i^f f_i(k) + B_i^u u_i(k), \tag{2.3}$$

where $B_i^f = [B_i^{\text{desc}}, B_i^{\text{climb}}, B_i^{\text{in}}]$ is the forcing matrix with a dimension of $m_i \times 3$, and the forcing input $f_i(k) = [f_i^{\text{desc}}(k), f_i^{\text{climb}}(k), f_i^{\text{in}}(k)]^T$, is a column vector with three elements. It is also noted that, when implementing delay control, the link level model must satisfy the following two assumptions:

(a) The delay control always takes place at the beginning of a time step.

(b) When an aircraft is under delay control, it is in one time increment units.

Therefore, if there are n aircraft under delay control in the m-th cell of link i at time instant k, and the control action lasts for $p+1$ time units, then the controlled input vector will be $u_i(k) = u_i(k+1) = \cdots = u_i(k+p) = [0, \cdots, 0, n, 0, \cdots, 0]^T$, where the m-th element of these vectors is equal to n. Because the input for delay control is linear, the superposition principle is satisfied. This means, for multiple delay controls taking place at the same time, the gross controlled input vector is the summation of each controlled input for each corresponding delay control.

Sector level model

Extending this modeling technique to set up a sector level model is fairly straightforward, because there is no interconnection (neither inputs, nor states) between different links in one sector. For example, to obtain the sector count, all link counts are added in this sector. Suppose that there are n links in the considered sector, then the state space equations for the model at the sector level can be written as:

$$x(k+1) = Ax(k) + B^{\text{in}} f^{\text{in}}(k) + B^{\text{desc}} f^{\text{desc}}(k) + B^{\text{climb}} f^{\text{climb}}(k) + B^{u} u(k) \tag{2.4}$$

$$y(k) = Cx(k), \tag{2.5}$$

where $x(k) = [x_n(k)^T, \cdots, x_1(k)^T]^T$ denotes the state, and $f^{\text{in}}(k) = [f_n^{\text{in}}(k)^T, \cdots, f_1^{\text{in}}(k)^T]^T$ is the forcing input vector, i.e. the entry count into the considered sector during a unit time interval from k to $k+1$. The descent input vector $f^{\text{desc}}(k) = [f_n^{\text{desc}}(k)^T, \cdots, f_1^{\text{desc}}(k)^T]^T$ and the climb input vector $f^{\text{climb}}(k) = [f_n^{\text{climb}}(k)^T, \cdots, f_1^{\text{climb}}(k)^T]^T$ are both column vectors with n elements. The controlled input vector, $u(k) = [u_n(k)^T, \cdots, u_1(k)^T]^T$ and the output $y(k)$ still represents the total aircraft count in the user-specified set of cells at time step k. Note that matrices

$$A = \mathbf{diag}(A_n, A_{n-1}, \cdots, A_2, A_1),$$

$$B^{\text{in}} = \begin{bmatrix} B_n^{\text{in}} & 0 & \dots & 0 & 0 \\ 0 & B_{n-1}^{\text{in}} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & B_2^{\text{in}} & 0 \\ 0 & 0 & \dots & 0 & B_1^{\text{in}} \end{bmatrix},$$

$$B^{\text{desc}} = \begin{bmatrix} B_n^{\text{desc}} & 0 & \dots & 0 & 0 \\ 0 & B_{n-1}^{\text{desc}} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & B_2^{\text{desc}} & 0 \\ 0 & 0 & \dots & 0 & B_1^{\text{desc}} \end{bmatrix},$$

$$B^{\text{climb}} = \begin{bmatrix} B_n^{\text{climb}} & 0 & \dots & 0 & 0 \\ 0 & B_{n-1}^{\text{climb}} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & B_2^{\text{climb}} & 0 \\ 0 & 0 & \dots & 0 & B_1^{\text{climb}} \end{bmatrix},$$

$$B^{u} = \begin{bmatrix} B_n^{u} & 0 & \dots & 0 & 0 \\ 0 & B_{n-1}^{u} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & B_2^{u} & 0 \\ 0 & 0 & \dots & 0 & B_1^{u} \end{bmatrix}$$

are all block matrices because states and inputs in this sector level model are all decoupled. C is given by $[C_n, C_{n-1}, \cdots, C_2, C_1]$.

In this sector level model, three inputs, $f^{\text{in}}(k)$, $f^{\text{desc}}(k)$ and $f^{\text{climb}}(k)$ can also be incorporated into one vector. Then, equation (2.4) can be rewritten as:

$$x(k+1) = Ax(k) + B^f f(k) + B^u u(k) \tag{2.6}$$

where

$$B^f = \begin{bmatrix} B_n^f & 0 & \dots & 0 & 0 \\ 0 & B_{n-1}^f & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & B_2^f & 0 \\ 0 & 0 & \dots & 0 & B_1^f \end{bmatrix},$$

$$f(k) = [f_n(k)^T, f_{n-1}(k)^T, \cdots, f_2(k)^T, f_1(k)^T]^T,$$

whose elements have been defined by equation (2.3).

The dimension of the state space for each sector depends on the number of total cells in the sector. For example, for sector ZOA33 of Oakland ARTCC, using one minute of flight time as the size of a cell, there are 84 cells in it, therefore the dimension of the state space is 84 and A is a 84×84 matrix for the ZOA33 sector level model. Figure 2.12 shows the hierarchical structure of the CTM(L).

ARTCC or multicommodity network level model

When an ARTCC level model is created, it is necessary to include *merge/diverge* nodes in the network [49; 66; 70; 9]. Merge nodes are straightforward: flows are added as streams of aircraft merge (see Figure 2.3 for an illustration of decoupled multicommodity network).

For diverge nodes, the corresponding routing choices must in general rely on knowledge of aircraft destination. Several approaches have been proposed to solve this problem, in particular *split coefficients* [49], which is inspired by the highway transportation literature [58; 23]. In the present work, an alternate way of modeling the problem is proposed based on *a priori* knowledge of the aircraft destination (provided by the ASDI/ETMS data), which is available in the form of filed

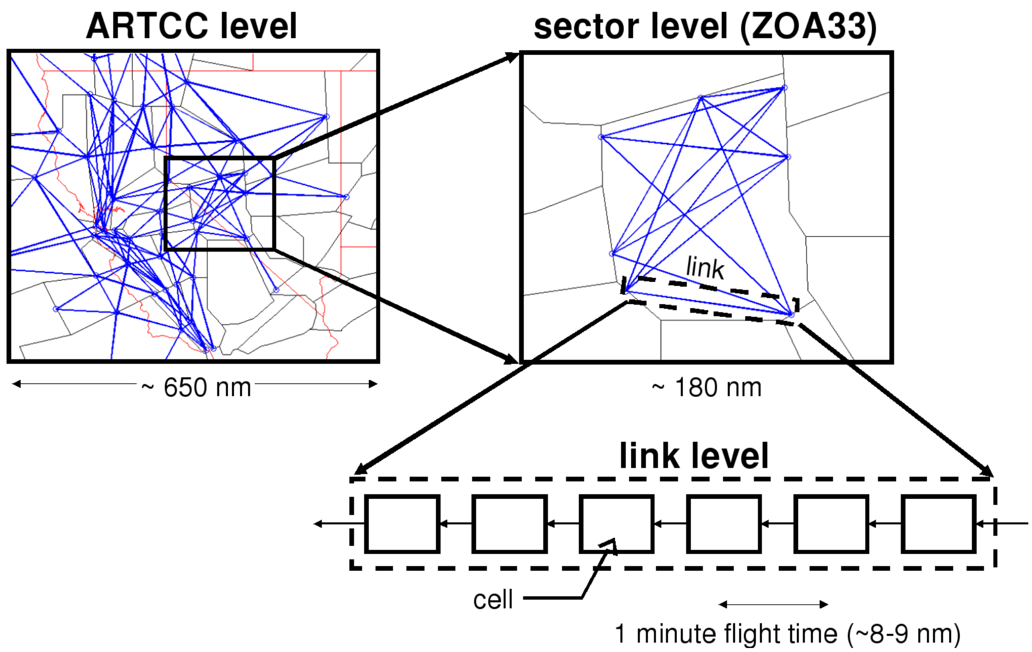

Figure 2.12: The hierarchical structure of CTM(L).

flight plans, designated long before the aircraft depart. One significant contribution of this thesis is thus to incorporate this knowledge into the model, which previous Eulerian models do not [49; 50; 66; 70; 9]. First, flights are clustered based on their entry-exit node pairs in the network. Each pair corresponds to a *path* consisting of links between these nodes. If two or more paths have one link in common, this link will be duplicated, using a multicommodity flow structure. Therefore, the NAS-wide model can also be cast in the framework of (2.4)–(2.5), where the matrices A, B^f, B^u and C now include all links of all sectors, and the corresponding $x(k)$ includes all cells of the complete network. The forcing input, $f(k)$, is now the entry count onto the NAS. The output, $y(k)$, denotes the aircraft count in a user-specified set of cells at time step k. The equations can be writen as follows:

$$x(k+1) = Ax(k) + B^f f(k) + B^u u(k) \tag{2.7}$$

$$y(k) = Cx(k), \tag{2.8}$$

where

$$A = \mathbf{diag}(A_n, A_{n-1}, \cdots, A_2, A_1),$$

$$B^f = \begin{bmatrix} B_n^f & 0 & \dots & 0 & 0 \\ 0 & B_{n-1}^f & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & B_2^f & 0 \\ 0 & 0 & \dots & 0 & B_1^f \end{bmatrix},$$

$$f(k) = [f_n(k)^T, f_{n-1}(k)^T, \cdots, f_2(k)^T, f_1(k)^T]^T$$

are all block matrices whose elements are similar to those defined by equation (2.3): the states now include all the cells along a *path* instead of a *link*.

The dimension of the state space for each ARTCC depends on the number of cells in the ARTCC and the number of merge/diverge nodes. For example, for the Oakland ARTCC, using one minute of flight time as the size of a cell, the dimension of the state space is 1,096 and A is a 1096×1096 matrix. For the network level model (the full continental US airspace), the dimension of the state space is 27,104. Since the sizes of the matrices A, B and C of the network level model are very large and the matrices are very sparse, the model is not directly implemented in the compact form as in (2.7) and (2.8). Instead, it is implemented as follows:

$$\begin{aligned} &x_{k+1,p,i} = x_{k,p,i-1} + u_{k,p,i} - u_{k,p,i-1}, && k \in \{0, \cdots, N-1\},\ p \in P,\ i \in \{2, \cdots, n_p\} \\ &x_{k,p,1} = f_{k,p} + u_{k,p,1}, && k \in \{0, \cdots, N\}, p \in P, \end{aligned} \tag{2.9}$$

where N is the time horizon, P is a set of paths, n_p is the number of cells in path p. k, p and i represent the time step, path index and cell index, respectively.

For different applications, the size of the cell can be changed to any value. For validation purposes, each cell is designated as one minute of flight time in this dissertation, which corresponds to the sampling rate of recorded ASDI/ETMS data. This framework requires more space and computational time than existing models. For example, the dimension of the Menon model [49] is the number of control volumes, which is five in the example model; the dimension of the dynamic stochastic model [66] is 23, which is the number of ARTCCs including one for the international region. However, the scalability of the CTM(L) model greatly facilitates the network model. A full comparison between this model, the PDE model [9], the one dimensional Menon model [49], and the two dimensional Menon model [50] is discussed in Chapter 3.

Controllability and observability

Controllability and observability play very important roles in control theory [54]. On one hand, if a discrete-time linear dynamical system is controllable, then for any initial state $x_0 \in \mathbb{R}^n$ and final state $x(k) \in \mathbb{R}^n$, there always exists an input sequence u, such that $x(k)$ will be reached from x_0 by the time $t = n$. On the other hand, if a dynamical system is observable, then for any $k \geq 0$, the initial state x_0 can be determined from the time history of the input $u(k)$ and the output $y(k)$ in the interval of $[0, k]$.

Controllability and observability are equally important in CTM(L) for TFM. If the system (2.7)–(2.8) is not controllable and observable, one cannot guarantee the existence of a feasible solution for TFM problems using CTM(L) (an example is shown in Chapter 3 with a TFM problem that is formulated with equation (3.21)). If a desired target flow pattern is not in the controllable subspace, this method provides an infeasibility certificate for the corresponding TFM policy.

Several algebraic or geometric criteria enable the verification of a dynamical system's controllability and observability (either continuous time or discrete time). For example, if the controllability matrix has full-row rank, then the system is controllable, or dually, if the observability matrix has full-column rank, then the system is observable [54]. For the system (2.7)–(2.8), it is easy to show that the controllability and observability matrices have full rank. This means that if there were no constraints on the inputs and the states (in particular, components of x are restricted to be integer and cannot be negative), the system would be controllable and observable. It is also an issue of interest to verify the validation of those algebraic or geometric criteria for an integer-valued system in the future. In fact, the verification of controllability and observability is related to the feasibility checks of a linear (integer) program, which is potentially NP-hard in the present case [10; 12; 29].

2.2.2 Flight routing

The multicommodity flow model makes it straightforward to incorporate different graph topologies into the CTM(L). In this section, three major connections of links will be introduced. Based on the control of these three connections, routing flights can be achieved. Inter-cell flows can be determined using a set of laws for different types of inter-cell connections, as shown in Figures 2.13-2.15.

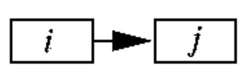

Figure 2.13: Simple connection.

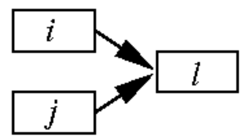

Figure 2.14: Merge, situation between cells i and j.

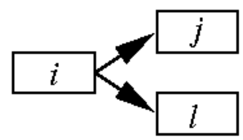

Figure 2.15: Diverge, situation at cell i into cells j and l.

Definitions

- i (j, l, etc.): cell number (integer).
- $x^i(k)$: aircraft count in cell i at time k.
- $\mathcal{N}(i)$: immediate downstream cells of cell i.
- $\mathcal{P}(i)$: immediate upstream cells of cell i.
- $u^i(k)$: delay control of traffic flow in cell i at time k.
- $u^{i \to j}(k)$: traffic flow from cell i to cell j at time k by prescribed *routing control*.

Simple connection

Two cells are said to be *simply connected* when they are directly connected without any intervening merging or diverging cells. Let i and j denote the upstream and downstream cells. The traffic flow is determined by the following law:

$$x^j(k+1) = x^i(k) - u^i(k) + u^j(k), \tag{2.10}$$

where

$$0 \leq u^p(k) \leq x^p(k), \quad p = i, j. \tag{2.11}$$

Equation (2.10) is a simple mass balance, while equation (2.11) encodes the fact that one cannot control more aircraft than actually present in a cell at a given time. Note that this is very close to the approach taken by Daganzo in his definition of the original CTM [22; 23].

Merge connection

Merge connection represents the configuration in which two cells i and j merge into one downstream cell k. The traffic flow is governed by the following laws:

$$x^l(k+1) = x^i(k) + x^j(k) - u^i(k) - u^j(k) + u^l(k), \tag{2.12}$$

where

$$0 \leq u^p(k) \leq x^p(k), \quad p = i, j, l. \tag{2.13}$$

In the general case, multiple incoming links merging laws can be represented as

$$x^l(k+1) = \sum_{p \in \mathcal{P}(l)} [x^p(k) - u^p(k)] + u^l(k), \tag{2.14}$$

where

$$0 \leq u^p(k) \leq x^p(k), \quad p \in \mathcal{P}(l) \cup \{l\}. \tag{2.15}$$

Diverge connection

Diverge connection is the configuration in which the upstream cell i diverges into two cells j and k. The diverge laws are

$$
\begin{aligned}
x^j(k+1) &= u^{i\to j}(k) + u^j(k), \\
x^l(k+1) &= u^{i\to l}(k) + u^l(k),
\end{aligned} \tag{2.16}
$$

where

$$
\begin{aligned}
u^{i\to j}(k) + u^{i\to l}(k) + u^i(k) &= x^i(k), \\
0 \le u^p(k) \le x^p(k), &\quad p = i, j, l \\
u^{i\to p}(k) \ge 0, &\quad p = j, l.
\end{aligned} \tag{2.17}
$$

In the general case, the diverging laws can be represented as

$$
x^p(k+1) = u^{i\to p}(k) + u^p(k), \quad p \in \mathcal{N}(i), \tag{2.18}
$$

where

$$
\begin{aligned}
\sum_{p\in\mathcal{N}(i)} u^{i\to p}(k) + u^i(k) &= x^i(k), \\
0 \le u^p(k) \le x^p(k), &\quad p \in \mathcal{N}(i) \cup \{i\}, \\
u^{i\to p}(k) \ge 0, &\quad p \in \mathcal{N}(i).
\end{aligned} \tag{2.19}
$$

For example, if some link, say the link starting with cell l, is completely closed because of weather, *Special Use Airspace* (SUA), congestion, etc., the situation can be modeled by imposing one additional constraint as follows to the constraints in (2.19):

$$
u^{i\to l}(k) = 0. \tag{2.20}
$$

The mathematical formulation of the three major connections of links is linear (in fact, integer linear).

2.3 Validation

Validation is the process of testing a model on a data set to demonstrate that the model performs as desired. Demonstration of the accuracy of flow models is obviously key in the process of incorporating them into decision support tools. The specific aspect of the model which we want to validate is its predictive capability, i.e., the capability to forecast traffic from a given demand (requested departure times and routes).

In the present case, validation consists of using OD input (i.e., for each aircraft, a departure airport, a destination airport, and a departure time) and showing that the model accurately produces sector counts for the period of interests. The counts are then compared with the ASDI/ETMS counts. In general form, it means that the model is able to predict flows of aircraft accurately based on OD demand information available in ASDI/ETMS data. Validations are performed using data from 8:00 a.m. GMT on January 24th, 2005 to 8:00 a.m. GMT on January 25th. The input to the models is the number of aircraft entering the considered region from airports through climb inputs (284 high altitude continental sectors of the United States). The predicted states and sector counts are computed from the model and compared with the recorded ASDI/ETMS data.

2.3.1 Sector counts

Sector counts predicted by the CTM(L) are first compared with sector counts obtained from the recorded ASDI/ETMS data. This study shows that the sector counts predicted by the model and the ASDI/ETMS data have the same trends for all the sectors in the model, and differ by an error of a small magnitude (mean errors are less than one for most sectors). This can be explained as follows: the travel time on a link in the network is computed as the aggregated travel

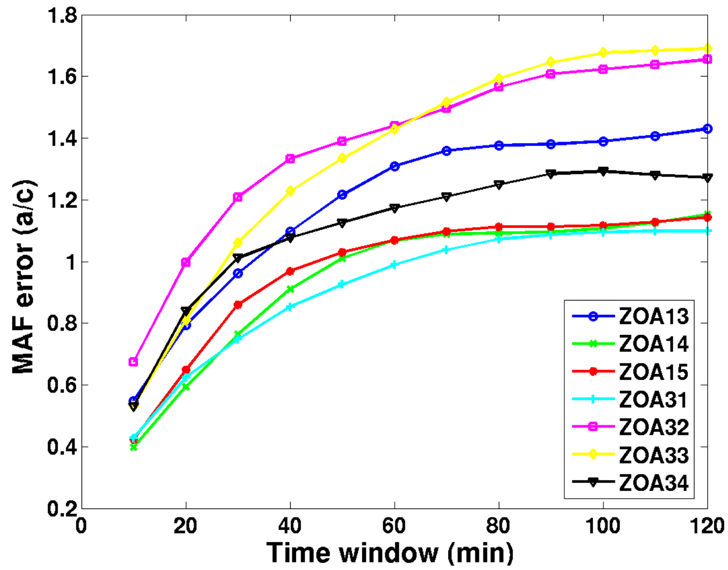

Figure 2.16: Average error between the filtered and unfiltered ASDI/ETMS data (sector counts). The mean error increases as the time window increases.

time for all flights in the data set used for the identification (one year in the present work), which is not necessariy equal to the travel time for the particular flights on a particular day.

To avoid small amplitude, high frequency fluctuations in the data caused by the sampling time and boundary crossing, a *moving average filter* (MAF) technique [69] is used to filter the sector counts for both the recorded ASDI/ETMS data and the model's simulation. Applying a MAF to the data requires an appropriate number of data points (time window) in the average. A small time window captures errors in the dynamics of the flow but loses the "filtering" benefits, while a large time window filters variations but loses the dynamics of the flow errors. To determine a proper size of the MAF time window, an experiment involving the average sector count error is performed. The average count error is the mean error, computed as the absolute difference between the MAF-filtered data and the raw unfiltered data, over the course of a simulation. Figure 2.16 shows the results obtained. It shows how the mean error increases as the time window (number of data points

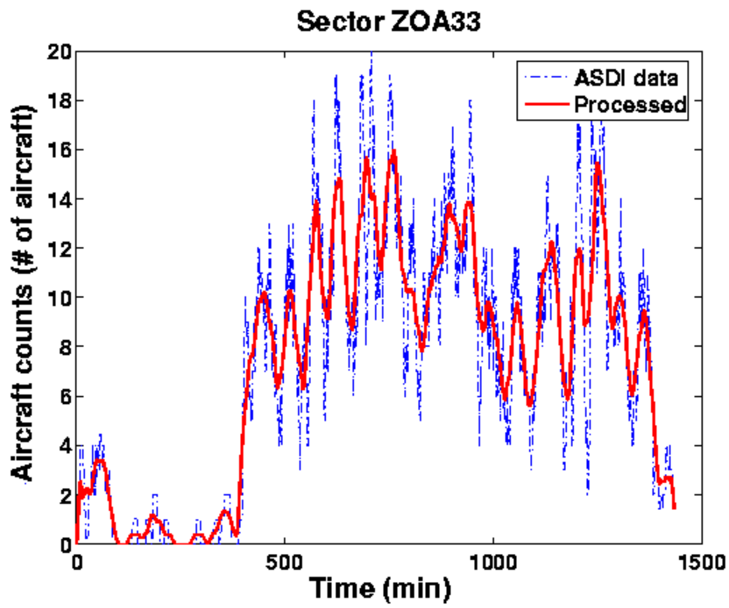

Figure 2.17: Moving Average Filter (MAF) data processing: the dotted curve represents the unfiltered sector counts of sector ZOA33, and the solid curve represents the filtered data using a time window of 20 minutes.

in the average) increases. Note that for most sectors, the mean errors are below one aircraft per sector, when the time window is 20 minutes. For this reason, 20 is chosen as the number of data points in the average (the time window, or time span). For this problem, removing variation makes physical sense. Very often, sector count exceeds legal values for a few minutes (if aircraft are about to exit a sector), which is tolerated in practice because such flights usually do not pose significant problems to air traffic controllers.

Figure 2.17 shows an example of the unfiltered raw data overlayed with the filtered data using MAF, which is more useful for flow pattern analysis. As can be seen, a significant portion of the undesired variation in the data can be removed by performing a MAF of the data, which makes it more suitable for analysis and comparison.

Figures 2.18–2.21 show the predicted and actual sector counts as a function of time in four

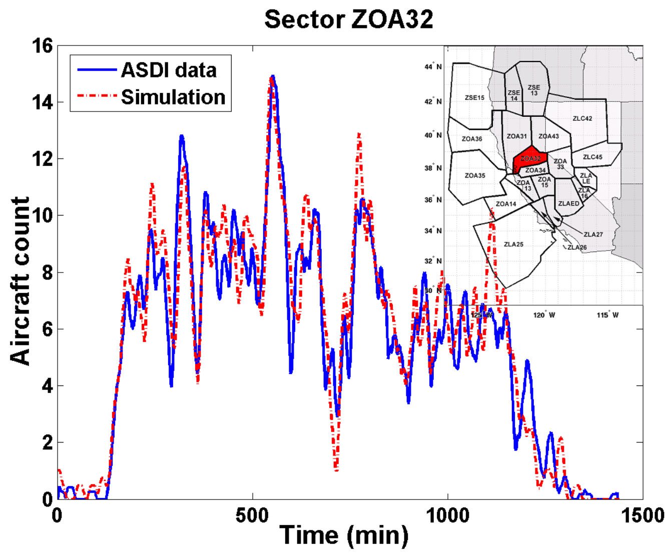

Figure 2.18: Sector ZOA32: comparison of the predictions of aircraft sector counts with the CTM(L) and ASDI/ETMS. Curves represent the processed sector counts after filtering. The map in the figure illustrates the location of the corresponding sector (shaded).

sectors: medium loaded sectors ZOA32 and ZOA34, highly loaded sector ZOA33, and low traffic sector ZOA35. The data shown in the figures is filtered by MAF. The figures show qualitatively that the model correctly predicts the trends of sector counts.

2.3.2 Quantitative error analysis

The sector count error analysis involves two comparisons: the sum of the error breach S, and the instantaneous error. S is defined as the summation of time intervals under the condition that the difference of sector counts between the simulation and the ASDI/ETMS data is greater than or equal to a user-specified capacity limitation, within a certain time window. This is summarized in equation (2.21):

$$S = \sum_{k=1}^{T} \mathbb{I}_{\{|y_{\text{sim}}(k) - y_{\text{ASDI/ETMS}}(k)| \geq C_s\}} \tag{2.21}$$

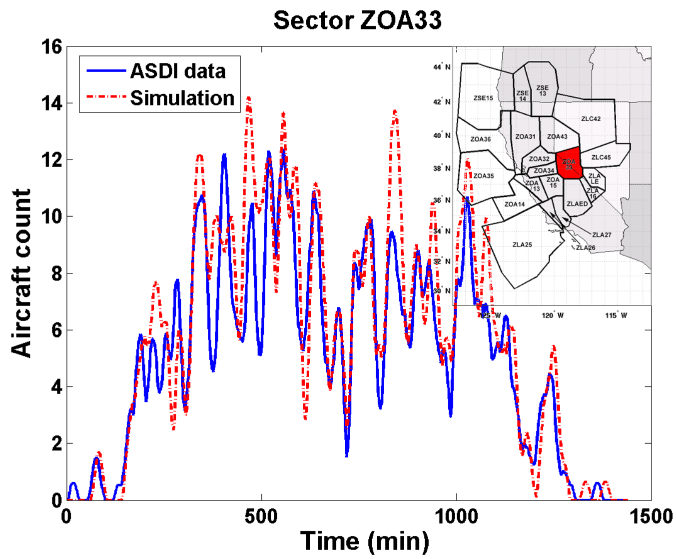

Figure 2.19: Sector ZOA33: comparison of the predictions of aircraft sector counts with the CTM(L) and ASDI/ETMS. Curves represent the processed sector counts after filtering. The map in the figure illustrates the location of the corresponding sector (shaded).

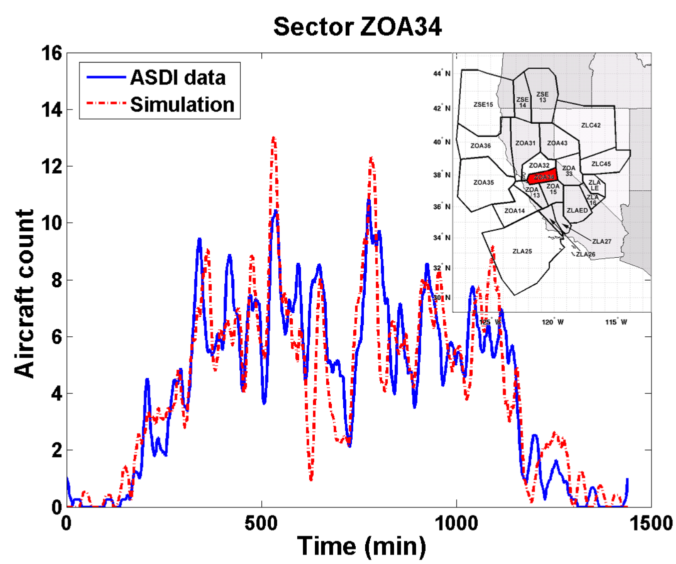

Figure 2.20: Sector ZOA34: comparison of the predictions of aircraft sector counts with the CTM(L) and ASDI/ETMS. Curves represent the processed sector counts after filtering. The map in the figure illustrates the location of the corresponding sector (shaded).

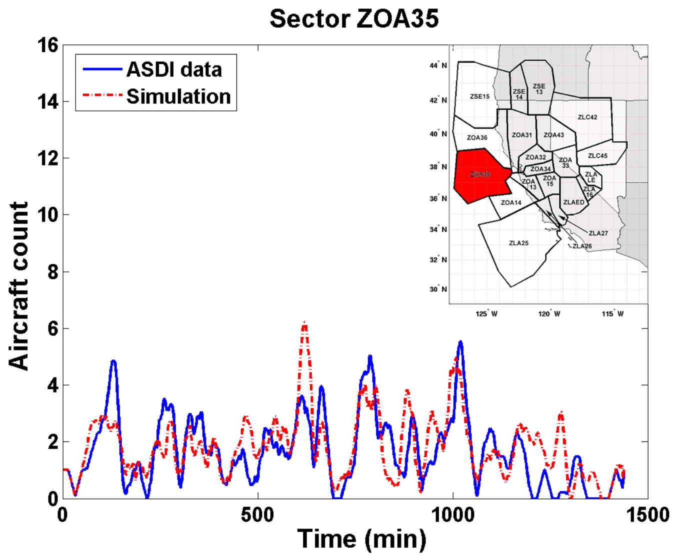

Figure 2.21: Sector ZOA35: comparison of the predictions of aircraft sector counts with the CTM(L) and ASDI/ETMS. Curves represent the processed sector counts after filtering. The map in the figure illustrates the location of the corresponding sector (shaded).

where $\mathbb{I}$ represents the indicator function. The sector count is denoted by $y(k)$, ASDI/ETMS and sim (or simulated). The constant C_s is a user-defined threshold. The time window chosen in the simulation is $T = 1440$ minutes (24 hours). To measure the similarity in the simulation and the ASDI/ETMS data, different values of C_s are used, and plots of percentage of breaches versus C_s are shown in Figure 2.22. For example, if $C_s = 3$, the percentage of breaches in sector ZOA35 is 7%, which means the predicted sector counts in ZOA35 by the model differ from the ASDI/ETMS data by at least three aircraft for 7% of the time. As the value of C_s increases, the breach length for each model tends to zero: the larger the aircraft count error, the shorter the breach length is.

An instantaneous sector count error analysis is performed as well. This error is the difference between the model's predicted aircraft count and the actual aircraft count for each sector, computed from the recorded ASDI/ETMS data at each time step in the simulation. The corresponding relative error is the ratio between the absolute instantaneous error and the actual count. The

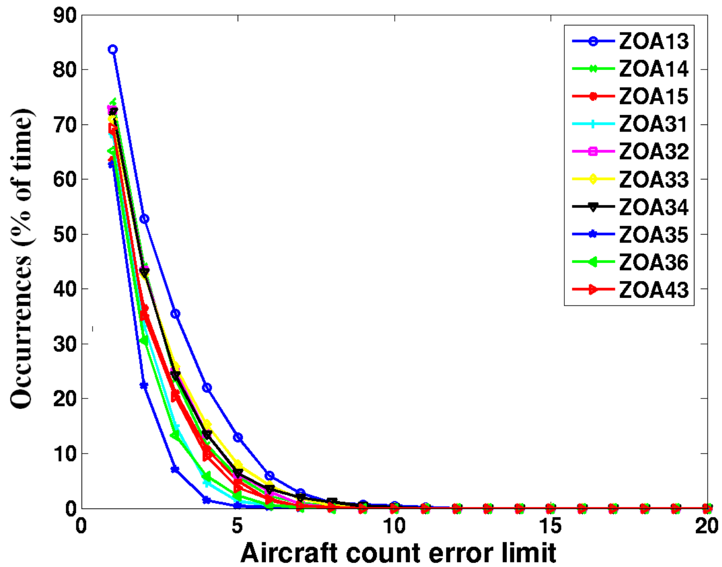

Figure 2.22: Occurrences of breach of sector count error for ten sectors in the Oakland Center.

instantaneous error and relative error are shown for sectors ZOA32, ZOA33 and ZOA34 in Figure 2.23. From Figure 2.23, it can be seen that for each of the three sectors (ZOA32, ZOA33 and ZOA34), the means of the instantaneous error are between 1.19 and 1.33, with a standard deviation between 1.51 and 1.96. For the relative errors, the largest error is four (for a very short period of time), but in general, the relative errors are less than one.

A summary of the prediction errors on July 2nd, 2005 for all the sectors in the study is presented in Tables 2.1–2.10. The error table shows that the CTM(L) works very well. For absolute errors, 51% of the sectors have mean errors less than one; for 99.65% of the sectors, the mean errors are below two. The maximum of the mean errors of all sectors is about two (sector ZTL15), which also has the largest standard deviation. This is because the traffic in Center ZTL was greatly influenced by the weather, and on a particular day (July 1st, 2005), there were more flights routed through ZTL15 than usual. In terms of signed errors, for 91.55% of all sectors, the mean errors are

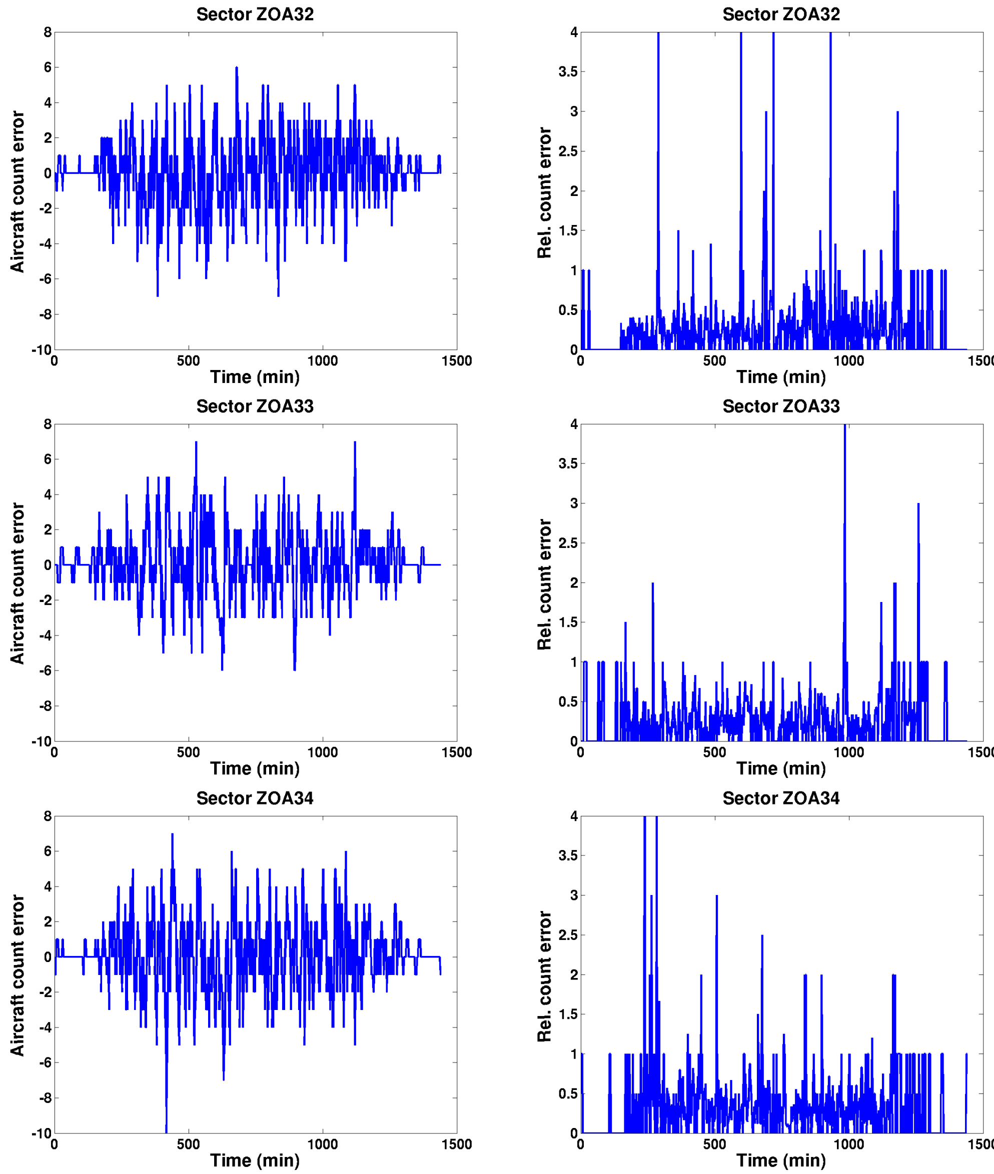

Figure 2.23: **Left:** Instantaneous errors for three high altitude sectors: ZOA32, ZOA33 and ZOA34. **Right:** Relative error for the three sectors.

between -1 and 1, which shows a good predictive performance of the CTM(L).

Name	Mean	Std. Dev.	Name	Mean	Std. Dev.	Name	Mean	Std. Dev.
ZOA13	2.18	1.77	ZOA14	1.63	1.17	ZOA15	1.38	1.12
ZOA31	1.23	0.72	ZOA32	1.64	1.26	ZOA33	1.69	1.50
ZOA34	1.66	1.44	ZOA35	0.94	0.46	ZOA36	1.18	0.77
ZOA43	1.40	1.03	ZLA16	0.91	0.56	ZLA25	2.00	1.79
ZLA26	1.78	1.42	ZLA27	1.71	1.77	ZLAED	1.29	1.21
ZLALE	0.23	0.14	ZLA30	1.60	1.40	ZLA31	1.93	1.91
ZLA32	1.54	1.34	ZLA34	1.17	0.95	ZLA36	1.63	1.32
ZLA37	1.32	0.99	ZLA38	1.37	1.17	ZLA39	1.14	0.94
ZLA40	1.32	1.02	ZLA60	1.11	0.80	ZSE01	0.51	0.25
ZSE02	1.16	0.65	ZSE03	0.97	0.40	ZSE07	1.07	0.61
ZSE11	1.04	0.63	ZSE12	1.10	0.63	ZSE13	0.95	0.63
ZSE14	1.21	0.72	ZSE15	0.86	0.47	ZSE16	1.01	0.74
ZSE31	0.83	0.41	ZSE32	0.85	0.50	ZSE42	1.18	0.64
ZSE46	1.41	1.03	ZSE47	0.77	0.38	ZSE48	0.62	0.32
ZLC03	0.96	0.60	ZLC04	1.67	1.38	ZLC05	1.14	0.67
ZLC06	0.98	0.54	ZLC07	1.25	0.77	ZLC08	0.98	0.64
ZLC15	0.90	0.50	ZLC16	0.82	0.53	ZLC17	1.25	0.64
ZLC20	1.12	0.75	ZLC33	1.31	1.08	ZLC34	1.60	1.48

Table 2.1: Validation results: absolute errors (1). Means and standard deviations of the absolute errors between the predicted and recorded sector counts.

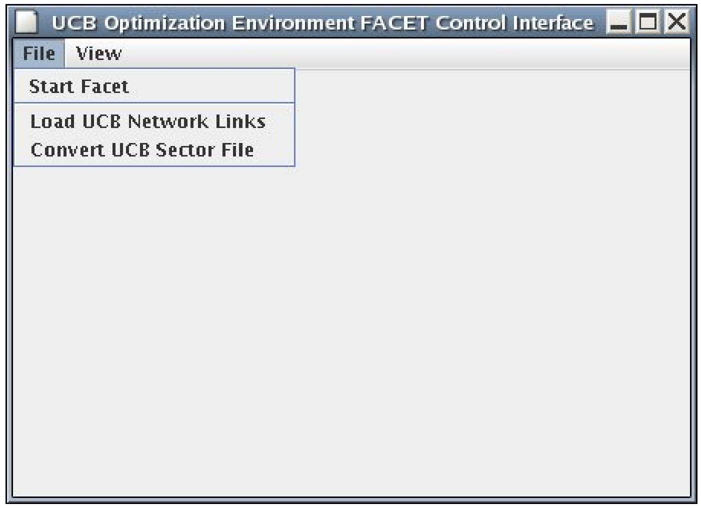

Figure 2.24: A screenshot of the interface that launches and controls FACET via the FACET *Application Programming Interface* (API).

2.4 Implementation

The CTM(L) has been implemented in C++ during this thesis and incorporated in FACET [14] by Metron Aviation [2]. Table 2.11 and Figures 2.24 – 2.28 show some snapshots of the implementation of the CTM(L) in FACET.

2.5 Conclusion

A new Eulerian-Lagrangian Large-capacity Cell Transmission Model of airspace was created and implemented using a full year of air traffic data and applied to high altitude traffic for all continental Air Traffic Control Centers of the National Airspace System in the United States. The Eulerian-Lagrangian model was reduced to a linear time invariant dynamical system, in which the state is a vector of aggregate aircraft counts. The model was validated against recorded air traffic data for the whole National Airspace System based on a full day of traffic.

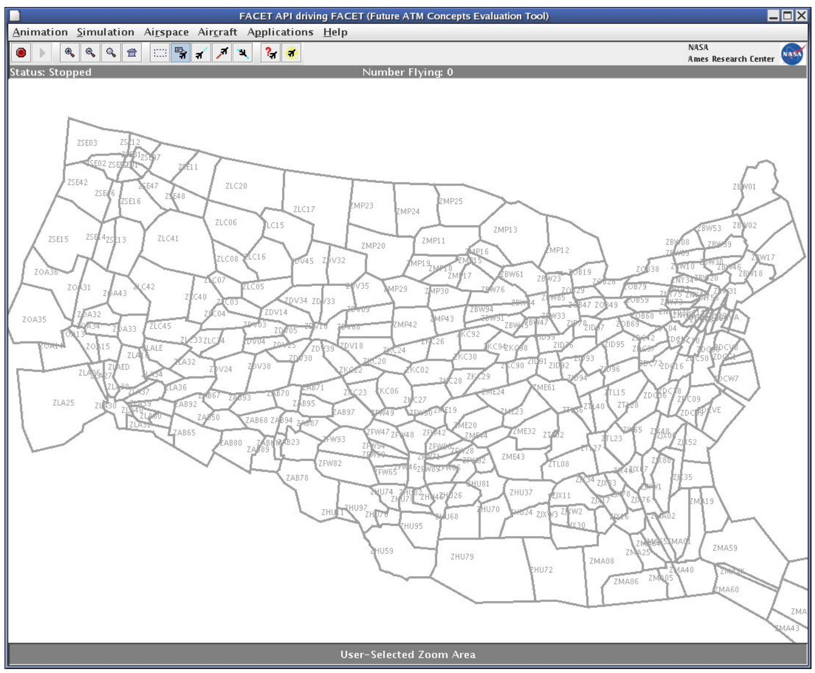

Figure 2.25: High altitude sectors in the model running in FACET.

Name	Mean	Std. Dev.	Name	Mean	Std. Dev.	Name	Mean	Std. Dev.
ZLC40	1.20	0.71	ZLC41	1.14	0.73	ZLC42	1.41	1.07
ZLC45	1.72	1.41	ZABHM	0.61	0.37	ZAB23	1.60	1.36
ZAB50	1.28	0.87	ZAB65	1.32	1.00	ZAB67	1.01	0.66
ZAB68	1.25	0.87	ZAB70	1.02	0.66	ZAB71	1.17	0.84
ZAB78	2.79	3.61	ZAB80	1.40	1.15	ZAB87	1.07	0.72
ZAB89	1.55	1.54	ZAB92	1.25	0.80	ZAB93	1.13	0.72
ZAB94	1.10	0.70	ZAB95	1.06	0.79	ZAB97	1.40	1.25
ZTL02	2.41	2.82	ZTL08	2.21	2.35	ZTL15	2.85	4.46
ZTL23	2.42	3.18	ZTL27	1.60	1.61	ZTL28	2.34	3.13
ZTL36	1.70	1.64	ZTL40	2.39	2.82	ZBW01	0.76	0.53
ZBW02	0.96	0.45	ZBW08	0.55	0.29	ZBW09	0.93	0.53
ZBW10	1.67	1.32	ZBW17	0.76	0.44	ZBW18	1.14	0.87
ZBW20	1.77	1.37	ZBW31	2.20	2.58	ZBW38	1.82	1.79
ZBW39	1.40	1.06	ZBW46	1.32	0.74	ZBW53	1.03	0.78
ZBW23	1.79	1.31	ZBW33	2.04	1.94	ZBW45	1.10	0.60
ZBW47	1.29	0.85	ZBW61	1.87	1.65	ZBW76	2.26	2.02
ZBW84	1.79	1.44	ZBW85	1.17	0.64	ZBW91	1.97	1.62
ZBW94	2.50	2.64	ZOB19	1.63	1.33	ZOB26	1.65	1.53
ZOB29	1.65	1.73	ZOB38	2.48	2.73	ZOB47	1.80	1.66

Table 2.2: Validation results: absolute errors (2). Means and standard deviations of the absolute errors between the predicted and recorded sector counts.

Name	Mean	Std. Dev.	Name	Mean	Std. Dev.	Name	Mean	Std. Dev.
ZOB49	1.78	1.68	ZOB59	1.90	2.07	ZOB68	1.49	1.22
ZOB69	1.85	1.89	ZOB79	2.50	3.22	ZDV03	1.44	1.27
ZDV04	1.33	1.21	ZDV05	1.34	1.00	ZDV08	1.84	1.84
ZDV09	1.31	0.91	ZDV14	1.71	1.29	ZDV16	1.49	1.13
ZDV18	1.87	1.60	ZDV24	1.80	1.92	ZDV25	1.65	1.41
ZDV30	1.93	1.56	ZDV32	0.87	0.47	ZDV33	1.46	0.96
ZDV34	1.59	1.15	ZDV35	1.54	1.00	ZDV38	1.60	1.33
ZDV39	2.04	1.95	ZDV45	1.00	0.58	ZFW28	1.03	0.72
ZFW39	1.05	0.62	ZFW42	1.39	1.32	ZFW46	1.15	0.80
ZFW47	1.52	1.51	ZFW48	1.62	1.74	ZFW49	1.02	0.71
ZFW50	1.30	1.18	ZFW65	1.63	1.34	ZFW71	0.79	0.45
ZFW82	1.63	1.44	ZFW86	1.07	0.74	ZFW89	1.25	0.81
ZFW90	0.88	0.48	ZFW92	0.85	0.52	ZFW93	1.43	1.34
ZFW94	1.06	0.84	ZHU11	0.01	0.01	ZHU24	1.28	1.02
ZHU26	0.96	0.48	ZHU37	1.62	1.61	ZHU46	1.02	0.69
ZHU59	1.23	1.05	ZHU68	1.13	1.02	ZHU70	1.24	0.88
ZHU72	0.04	0.02	ZHU74	1.45	1.10	ZHU76	0.70	0.43
ZHU78	1.01	0.64	ZHU79	0.20	0.11	ZHU81	1.06	0.73
ZHU82	1.08	0.77	ZHU95	1.25	1.03	ZHU97	1.15	0.82

Table 2.3: Validation results: absolute errors (3). Means and standard deviations of the absolute errors between the predicted and recorded sector counts.

Name	Mean	Std. Dev.	Name	Mean	Std. Dev.	Name	Mean	Std. Dev.
ZID76	1.62	1.45	ZID78	1.86	1.61	ZID91	1.73	1.52
ZID92	1.47	1.17	ZID93	1.72	1.77	ZID94	1.93	2.27
ZID95	2.16	2.50	ZID96	1.81	1.78	ZID97	1.75	1.67
ZID99	1.96	2.02	ZJXW1	3.27	4.80	ZJXW2	0.47	0.37
ZJXW3	0.32	0.18	ZJX00	0.28	0.15	ZJX11	1.16	0.93
ZJX16	2.57	4.20	ZJX17	1.94	2.20	ZJX30	1.26	1.09
ZJX33	1.28	1.10	ZJX34	1.89	2.10	ZJX35	1.72	1.93
ZJX48	1.92	2.48	ZJX49	2.24	2.75	ZJX52	1.97	2.60
ZJX65	2.31	3.29	ZJX67	2.70	4.24	ZJX76	2.17	2.96
ZJX78	1.77	1.95	ZJX88	0.72	0.48	ZKC02	1.03	0.60
ZKC06	1.04	0.64	ZKC20	1.17	0.88	ZKC22	1.16	0.79
ZKC23	1.28	0.96	ZKC24	1.78	1.36	ZKC26	1.74	1.57
ZKC27	1.50	1.20	ZKC28	1.20	0.86	ZKC29	1.48	1.20
ZKC30	1.49	0.95	ZKC90	1.64	1.52	ZKC92	1.74	1.52
ZKC94	2.06	1.96	ZKC98	1.39	0.96	ZME19	1.45	0.97
ZME20	1.70	1.53	ZME23	1.84	1.72	ZME24	1.88	1.73
ZME32	2.21	2.36	ZME43	1.77	1.76	ZME44	1.12	0.95
ZME61	2.02	2.33	ZMAXX	0.36	0.20	ZMA01	1.63	1.77
ZMA02	2.69	4.09	ZMA05	1.51	1.18	ZMA06	0.88	0.63

Table 2.4: Validation results: absolute errors (4). Means and standard deviations of the absolute errors between the predicted and recorded sector counts.

Name	Mean	Std. Dev.	Name	Mean	Std. Dev.	Name	Mean	Std. Dev.
ZMA08	1.09	0.85	ZMA19	0.68	0.52	ZMA25	1.87	2.07
ZMA30	0.00	0.00	ZMA40	1.49	1.29	ZMA43	0.00	0.00
ZMA59	0.98	1.13	ZMA60	0.62	0.38	ZMA62	0.00	0.00
ZMA63	0.08	0.04	ZMA64	1.84	1.77	ZMA65	1.95	2.74
ZMP11	1.91	2.28	ZMP12	1.15	0.76	ZMP13	1.07	0.63
ZMP15	0.94	0.54	ZMP16	1.09	0.70	ZMP17	1.76	1.61
ZMP18	1.55	1.08	ZMP19	1.29	0.79	ZMP20	1.51	1.11
ZMP23	0.87	0.44	ZMP24	1.07	0.59	ZMP25	0.92	0.49
ZMP29	1.54	0.83	ZMP30	1.84	1.25	ZMP42	2.47	2.30
ZMP43	2.03	1.79	ZNY09	1.73	1.65	ZNY10	1.61	1.43
ZNY34	1.30	0.92	ZNY42	2.44	3.08	ZNY49	1.35	1.03
ZNY56	1.45	1.14	ZNY73	1.24	0.90	ZNY75	1.38	0.97
ZDC04	1.52	1.42	ZDC09	2.26	3.03	ZDC10	1.52	1.32
ZDC12	1.57	1.46	ZDC16	2.03	2.19	ZDC18	0.85	0.55
ZDC19	1.55	1.41	ZDC36	1.59	1.52	ZDC37	1.58	1.32
ZDC38	1.61	1.50	ZDC42	1.57	1.42	ZDC50	3.15	5.46
ZDC58	1.81	1.94	ZDC59	1.77	1.85	ZDC72	2.61	3.47
ZDC97	0.26	0.15	ZDC98	0.93	0.52	ZDC99	1.33	1.15
ZDCG1	0.01	0.00	ZDCVA	0.16	0.08	ZDCVB	0.00	0.00

Table 2.5: Validation results: absolute errors (5). Means and standard deviations of the absolute errors between the predicted and recorded sector counts.

Name	Mean	Std. Dev.	Name	Mean	Std. Dev.	Name	Mean	Std. Dev.
ZOA13	-0.67	3.91	ZOA14	-0.71	2.24	ZOA15	0.69	1.84
ZOA31	-0.17	1.46	ZOA32	-0.20	2.60	ZOA33	-0.63	2.74
ZOA34	0.01	2.82	ZOA35	-0.18	0.89	ZOA36	-0.02	1.46
ZOA43	-0.28	1.98	ZLA16	-0.41	0.89	ZLA25	-1.43	2.75
ZLA26	-0.40	2.93	ZLA27	0.96	2.77	ZLAED	1.24	1.28
ZLALE	0.23	0.14	ZLA30	-0.94	2.24	ZLA31	-1.21	3.04
ZLA32	-0.71	2.27	ZLA34	-0.29	1.59	ZLA36	0.34	2.58
ZLA37	0.40	1.78	ZLA38	-0.71	1.86	ZLA39	0.87	1.21
ZLA40	0.76	1.61	ZLA60	0.39	1.34	ZSE01	0.04	0.38
ZSE02	-0.30	1.28	ZSE03	0.28	0.83	ZSE07	-0.19	1.17
ZSE11	-0.15	1.16	ZSE12	-0.45	1.14	ZSE13	0.17	1.06
ZSE14	0.07	1.44	ZSE15	-0.39	0.76	ZSE16	-0.24	1.23
ZSE31	-0.25	0.72	ZSE32	-0.06	0.86	ZSE42	-0.56	1.18
ZSE46	0.71	1.77	ZSE47	-0.05	0.67	ZSE48	-0.16	0.50
ZLC03	0.45	0.96	ZLC04	-0.23	2.75	ZLC05	-0.14	1.31
ZLC06	-0.23	1.00	ZLC07	-0.32	1.50	ZLC08	-0.22	1.09
ZLC15	-0.11	0.90	ZLC16	-0.24	0.84	ZLC17	-0.23	1.40
ZLC20	-0.33	1.32	ZLC33	0.01	1.93	ZLC34	-0.75	2.47

Table 2.6: Validation results: signed errors (1). Means and standard deviations of the signed errors of (recorded sector counts − predicted counts).

Name	Mean	Std. Dev.	Name	Mean	Std. Dev.	Name	Mean	Std. Dev.
ZLC40	0.56	1.28	ZLC41	0.11	1.37	ZLC42	0.12	2.06
ZLC45	-0.56	2.73	ZABHM	0.58	0.39	ZAB23	1.53	1.47
ZAB50	0.22	1.66	ZAB65	-0.22	1.85	ZAB67	-0.11	1.16
ZAB68	0.40	1.58	ZAB70	0.17	1.16	ZAB71	0.12	1.51
ZAB78	-2.65	4.01	ZAB80	-0.06	2.12	ZAB87	0.20	1.27
ZAB89	-1.04	2.19	ZAB92	-0.48	1.46	ZAB93	0.04	1.36
ZAB94	0.14	1.29	ZAB95	-0.03	1.36	ZAB97	-0.10	2.22
ZTL02	-0.26	5.69	ZTL08	-0.92	4.35	ZTL15	-1.02	8.01
ZTL23	-0.80	5.78	ZTL27	0.17	2.88	ZTL28	-1.08	5.29
ZTL36	-0.87	2.70	ZTL40	-0.24	5.65	ZBW01	0.14	0.81
ZBW02	-0.28	0.87	ZBW08	-0.17	0.43	ZBW09	-0.45	0.86
ZBW10	0.20	2.69	ZBW17	0.26	0.70	ZBW18	-0.43	1.43
ZBW20	-0.10	2.94	ZBW31	1.07	4.43	ZBW38	0.36	3.39
ZBW39	-0.51	1.91	ZBW46	-0.02	1.61	ZBW53	-0.55	1.16
ZBW23	-0.81	2.59	ZBW33	-0.66	3.80	ZBW45	0.25	1.17
ZBW47	0.34	1.63	ZBW61	-1.03	2.88	ZBW76	0.42	4.50
ZBW84	-0.73	2.77	ZBW85	-0.06	1.33	ZBW91	-0.29	3.52
ZBW94	-1.34	4.87	ZOB19	-0.35	2.61	ZOB26	-0.10	2.88
ZOB29	-0.54	2.95	ZOB38	-1.67	4.43	ZOB47	-0.31	3.24

Table 2.7: Validation results: signed errors (2). Means and standard deviations of the signed errors of (recorded sector counts − predicted counts).

Name	Mean	Std. Dev.	Name	Mean	Std. Dev.	Name	Mean	Std. Dev.
ZOB49	-0.51	3.14	ZOB59	-0.52	3.75	ZOB68	-0.10	2.33
ZOB69	-0.03	3.59	ZOB79	-1.15	5.68	ZDV03	-0.45	2.21
ZDV04	-0.10	2.09	ZDV05	0.27	1.87	ZDV08	-0.24	3.51
ZDV09	0.11	1.76	ZDV14	-0.05	2.74	ZDV16	0.19	2.22
ZDV18	0.34	3.28	ZDV24	0.80	3.21	ZDV25	0.19	2.75
ZDV30	0.11	3.42	ZDV32	-0.14	0.84	ZDV33	-0.21	2.01
ZDV34	-0.49	2.30	ZDV35	-0.27	2.15	ZDV38	-0.02	2.61
ZDV39	-0.15	4.02	ZDV45	-0.24	1.05	ZFW28	-0.18	1.24
ZFW39	0.26	1.14	ZFW42	0.22	2.25	ZFW46	-0.05	1.46
ZFW47	-0.10	2.66	ZFW48	-0.09	3.05	ZFW49	0.23	1.20
ZFW50	-0.51	1.90	ZFW65	-0.55	2.52	ZFW71	-0.43	0.67
ZFW82	-0.30	2.73	ZFW86	0.14	1.30	ZFW89	-0.05	1.59
ZFW90	-0.12	0.86	ZFW92	-0.05	0.88	ZFW93	1.06	1.80
ZFW94	-0.32	1.35	ZHU11	-0.00	0.01	ZHU24	-0.56	1.68
ZHU26	0.06	0.94	ZHU37	-0.02	2.93	ZHU46	0.07	1.22
ZHU59	-0.58	1.63	ZHU68	-0.35	1.59	ZHU70	-0.50	1.53
ZHU72	0.02	0.02	ZHU74	-0.27	2.12	ZHU76	-0.29	0.63
ZHU78	0.23	1.13	ZHU79	0.18	0.11	ZHU81	-0.25	1.26
ZHU82	0.04	1.35	ZHU95	-0.33	1.76	ZHU97	-0.56	1.33

Table 2.8: Validation results: signed errors (3). Means and standard deviations of the signed errors of (recorded sector counts − predicted counts).

Name	Mean	Std. Dev.	Name	Mean	Std. Dev.	Name	Mean	Std. Dev.
ZID76	-0.13	2.76	ZID78	0.26	3.31	ZID91	-0.19	3.00
ZID92	-0.49	2.13	ZID93	0.04	3.24	ZID94	-0.54	4.00
ZID95	-0.86	4.46	ZID96	-0.50	3.30	ZID97	0.03	3.21
ZID99	-0.16	3.92	ZJXW1	3.25	4.86	ZJXW2	0.43	0.39
ZJXW3	0.25	0.20	ZJX00	0.08	0.18	ZJX11	0.57	1.44
ZJX16	1.02	7.00	ZJX17	-0.72	3.81	ZJX30	-0.29	1.85
ZJX33	0.04	1.93	ZJX34	-0.43	3.78	ZJX35	-0.87	3.03
ZJX48	0.33	4.27	ZJX49	0.17	5.24	ZJX52	-1.29	3.70
ZJX65	-0.19	5.95	ZJX67	-1.77	6.30	ZJX76	-0.81	5.00
ZJX78	-0.19	3.49	ZJX88	0.64	0.53	ZKC02	0.00	1.14
ZKC06	0.14	1.18	ZKC20	0.24	1.53	ZKC22	0.06	1.46
ZKC23	-0.13	1.77	ZKC24	0.50	2.83	ZKC26	0.48	2.97
ZKC27	-0.02	2.33	ZKC28	-0.22	1.55	ZKC29	-0.27	2.25
ZKC30	0.20	2.04	ZKC90	-0.43	2.77	ZKC92	0.55	2.88
ZKC94	-0.06	4.09	ZKC98	-0.14	1.92	ZME19	0.27	1.98
ZME20	-0.20	2.95	ZME23	-0.22	3.39	ZME24	-0.46	3.38
ZME32	0.19	4.78	ZME43	-0.09	3.33	ZME44	-0.35	1.51
ZME61	0.32	4.32	ZMAXX	0.02	0.27	ZMA01	-0.87	2.72
ZMA02	1.72	6.23	ZMA05	-0.19	2.30	ZMA06	-0.49	0.90

Table 2.9: Validation results: signed errors (4). Means and standard deviations of the signed errors of (recorded sector counts − predicted counts).

Name	Mean	Std. Dev.	Name	Mean	Std. Dev.	Name	Mean	Std. Dev.
ZMA08	-0.29	1.40	ZMA19	-0.14	0.74	ZMA25	0.48	3.71
ZMA30	0.00	0.00	ZMA40	-0.07	2.40	ZMA43	0.00	0.00
ZMA59	-0.09	1.61	ZMA60	-0.01	0.58	ZMA62	-0.00	0.00
ZMA63	-0.03	0.04	ZMA64	0.41	3.38	ZMA65	1.39	3.66
ZMP11	1.09	3.52	ZMP12	-0.30	1.39	ZMP13	-0.27	1.18
ZMP15	-0.15	0.97	ZMP16	-0.17	1.28	ZMP17	0.01	3.15
ZMP18	-0.26	2.25	ZMP19	-0.06	1.61	ZMP20	-0.45	2.15
ZMP23	0.04	0.82	ZMP24	0.13	1.16	ZMP25	0.15	0.91
ZMP29	-0.24	1.98	ZMP30	0.09	2.94	ZMP42	-0.72	5.09
ZMP43	-0.85	3.49	ZNY09	0.65	2.94	ZNY10	-0.31	2.68
ZNY34	0.28	1.73	ZNY42	-0.79	5.74	ZNY49	-0.05	1.95
ZNY56	0.20	2.17	ZNY73	-0.37	1.61	ZNY75	-0.01	1.92
ZDC04	0.07	2.57	ZDC09	-1.56	4.36	ZDC10	0.16	2.47
ZDC12	0.41	2.61	ZDC16	-0.09	4.24	ZDC18	0.46	0.81
ZDC19	-0.60	2.42	ZDC36	0.02	2.79	ZDC37	-0.17	2.56
ZDC38	0.40	2.71	ZDC42	-0.36	2.59	ZDC50	-2.08	8.25
ZDC58	-1.32	2.70	ZDC59	0.67	3.19	ZDC72	0.20	6.85
ZDC97	0.14	0.17	ZDC98	0.89	0.56	ZDC99	1.31	1.18
ZDCG1	-0.00	0.00	ZDCVA	0.00	0.09	ZDCVB	0.00	0.00

Table 2.10: Validation results: signed errors (5). Means and standard deviations of the signed errors of (recorded sector counts − predicted counts).

Figure 2.24	The interface that launches and controls CTM(L) in FACET.
Figure 2.26	Network model of the NAS.
Figure 2.27	Network of the northwest part in the NAS.
Figure 2.28	Aggregate traffic flow in the NAS.

Table 2.11: List of snapshots of CTM(L) implementation in FACET.

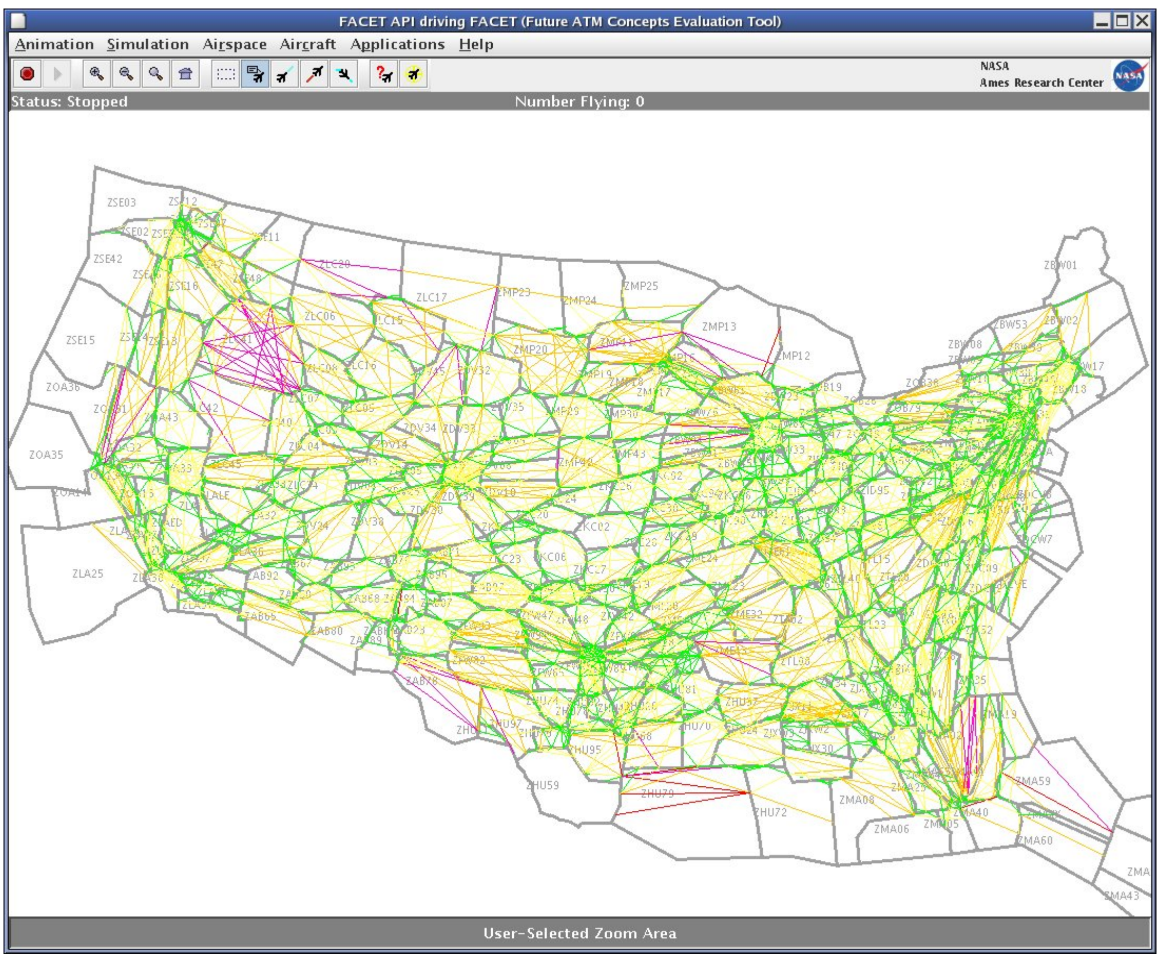

Figure 2.26: Visualization of the network model of the NAS in FACET in the Metron implementation.

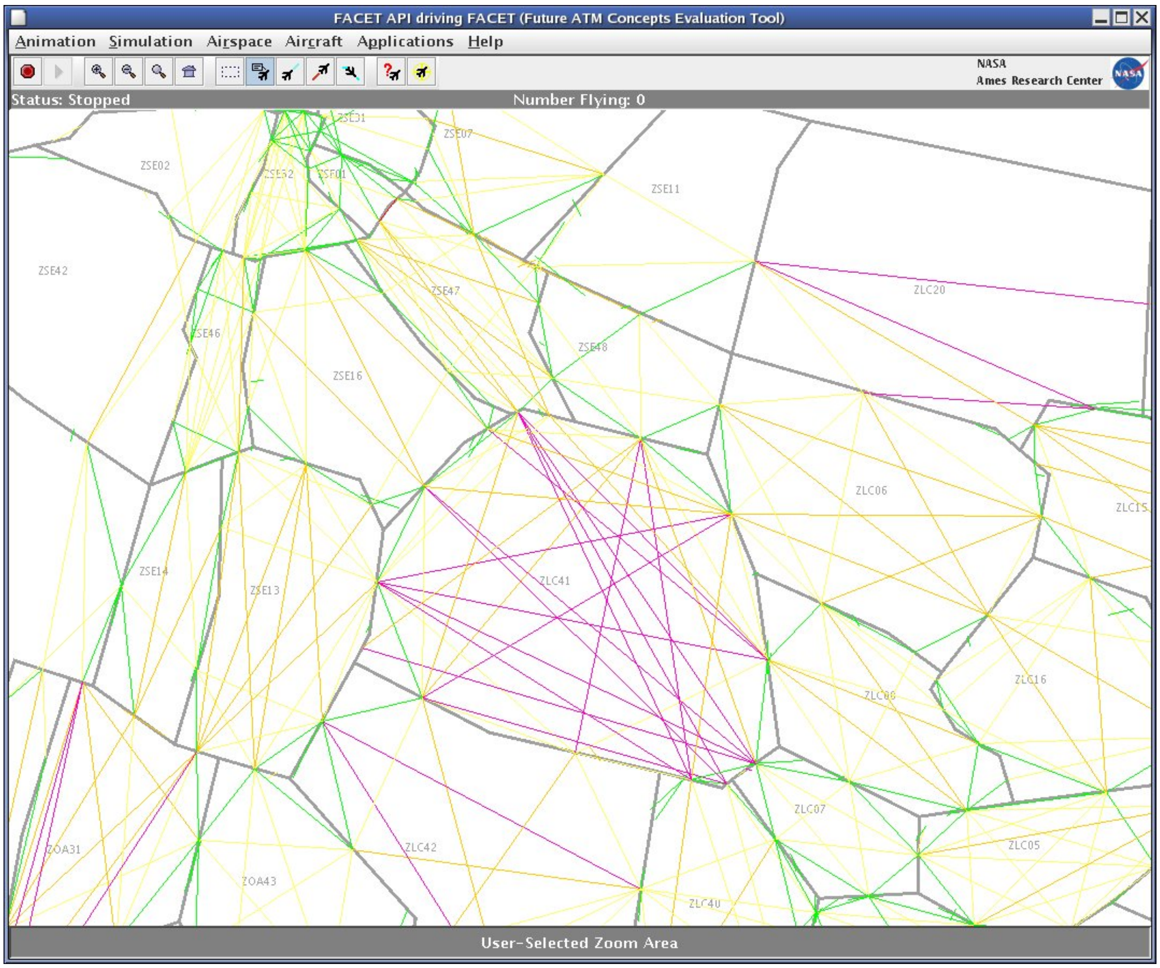

Figure 2.27: Magnified subsection of the network for the northwest part in the NAS.

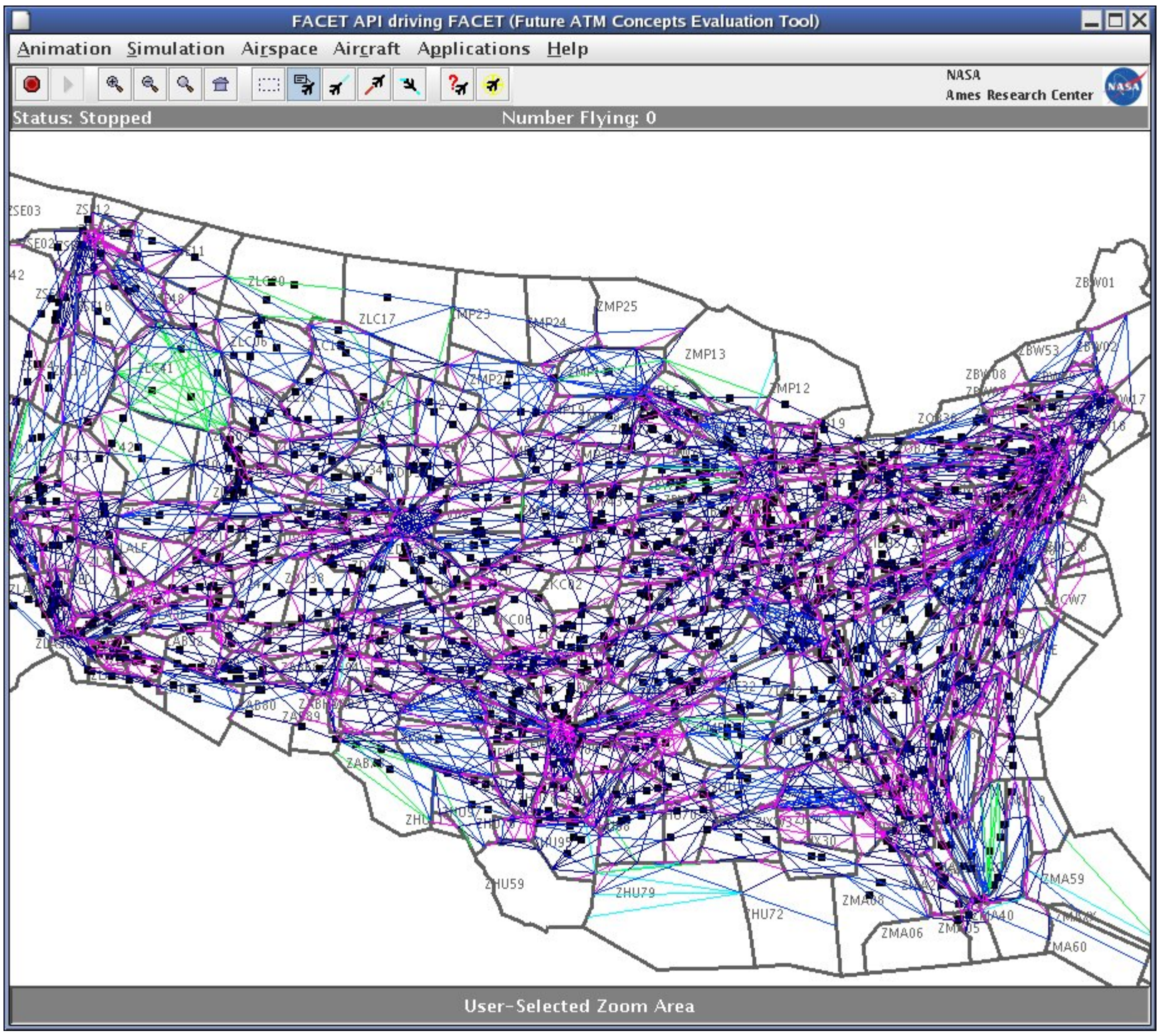

Figure 2.28: Aggregate traffic flow in the NAS. Volumes of air traffic evolve on top of the NAS network.

Chapter 3

Assessment of the performance of the CTM(L) model

This chapter compares the performance of four aggregate flow models including the CTM(L) developed in this thesis. The predictive capabilities of each model are compared through a careful validation against recorded air traffic data (ASDI/ETMS), following the procedure outlined in the previous section. The performances of the different models are compared, in particular, the accuracy of their predictive capabilities, computational time, and memory requirements. A discussion follows that highlights the structural differences between the four models and explains why one model may outperform another. Finally this chapter also presents the framework which needs to be used for the respective models in order to perform optimal control.

3.1 Models

This section presents a short summary of each of the four models used for this chapter. A detailed description of each model is available in the corresponding references, in particular for the models which are developed as part of this thesis. For completeness of this chapter, the CTM(L) model proposed in Chapter 2 is also briefly summarized.

3.1.1 The Large-capacity Cell Transmission Model (CTM(L))

The *Large-capacity Cell Transmission Model* (CTM(L)) was developed in Chapter 2, and appeared in [72; 65]. It uses a graph-theoretic representation of traffic flow. Air traffic flow on this graph is modeled as a discrete time dynamical system evolving on a network. To formulate the model, the following assumptions are made:

1. Each link of the network is modeled as a directional edge, which can be used for some of the other models in this chapter as well.

2. All aircraft in a given link fly at an aggregate speed. This speed can be obtained by aggregating the speed (obtained from the ASDI/ETMS data) of all aircraft following this link. To a certain extent, this aggregated speed can also be used for the other models presented here.

3. The number of cells in one link is given by the number of steps of expected travel time. In this implementation, one minute is taken as a unit time step. For example, if it takes around 12 minutes for an aircraft to fly across a sector, following a particular link, then this link would be divided into 12 segments, called *cells*. The choice of the cell length (time discretization) is arbitrary. In the model, a link indexed by i has m_i cells. As the time step decreases, the

model becomes more accurate, but at the expense of increased computational complexity.

4. At the link level, only high altitude traffic (above 24,000 feet) is taken into account for the calculation of aircraft count. This choice is also arbitrary and can be adjusted to any user-defined level. The same assumption is also made for the other models for fairness of the comparison.

5. The control strategy (based on the application of delays to aircraft) is mainly used as the controlled input to the model, which can be implemented in many forms: speed change, *vector for spacing* (VFS), *holding pattern* (HP), etc. It is applied in time increments corresponding to the unit time step.

6. The model is *deterministic*. No statistical factor, such as weather impact, is taken into consideration at this stage. Note that it can be added later using a stochastic framework [66; 70]. Such comparison will be the objective of future work.

7. In this model, all states, inputs and outputs are integer valued. This might increase the complexity of computation or analysis, in particular, the computational complexity for optimization which is integer program, but this provides higher accuracy.

Under the assumption that air traffic flow can be accurately represented by an aggregated travel time, the behavior of aircraft flow on a single link can be modeled by a deterministic linear dynamical system with a unit time delay, presented in Chapter 2:

$$x_i(k+1) = A_i x_i(k) + B_i^f f_i(k) + B_i^u u_i(k), \tag{3.1}$$

$$y(k) = C_i x_i(k), \tag{3.2}$$

where $x_i(k) = [x_i^{m_i}(k), \cdots, x_i^1(k)]^T$ is the state vector, whose elements represent the corresponding aircraft counts in each cell of link i at time step k, and m_i is the number of cells in the link. The forcing input, $f_i(k)$, is a scalar that denotes the entry count onto link i during a unit time interval from k to $k+1$, and the control input, $u_i(k)$, is an $m_i \times 1$ vector, representing holding pattern control. The output, $y(k)$, is the aircraft count in a user-specified set of cells at time step k. The nonzero elements of the $m_i \times 1$ vector C_i correspond to the cells in the user-specified set, and are equal to one. A_i is an $m_i \times m_i$ nilpotent matrix with 1's on its super-diagonal. $B_i^f = [0, \cdots, 0, 1]^T$ is the forcing vector with m_i elements, and B_i^u is the $m_i \times m_i$ holding pattern matrix, in which all nonzero elements are 1 on the diagonal and -1 on the super-diagonal.

The extension to sectors was presented in Chapter 2 and is quickly summarized here. Suppose there are n links in a sector, then the state space equations for the model at the sector level can be described as:

$$x(k+1) = Ax(k) + B^f f(k) + B^u u(k), \tag{3.3}$$

$$y(k) = Cx(k), \tag{3.4}$$

where $x(k) = [x_n(k), \cdots, x_1(k)]^T$ denotes the state, and $f(k) = [f_n(k), \cdots, f_1(k)]^T$ is the forcing input vector (the entry count onto the sector). The control input vector $u(k) = [u_n(k), \cdots, u_1(k)]^T$. The vector $y(k)$ represents the aircraft count in a user-specified set of cells at time step k. The matrices A, B^f and B^u are block diagonal, with block elements associated with each link in the sector. For example, $A = \text{diag}(A_n, \cdots, A_1)$ with A_i's defined by equation (3.1).

When an ARTCC level model is created, it is necessary to include *merge/diverge* nodes in the network [49; 66; 9; 71]. Merge nodes are straightforward: flows are added as streams of aircraft merge. For diverge nodes, the corresponding routing choices must rely on knowledge of the aircraft destination. The modeling of the problem is proposed based on *a priori* knowledge of the aircraft destination (provided by ASDI/ETMS data): knowledge of each aircraft destination is available long before its departure in the form of filed flight plans. One significant contribution of the CTM(L) is to incorporate this knowledge into the model with use of the notion of multicommodity flow, which previous Eulerian models do not use [49; 50; 66; 70; 9].

3.1.2 The Modified Menon Model (MMM)

This section is based on the work presented in [49]. The model has been modified to fit the structure of the graph model that will be discussed in Section 3.1.5, which provides a good comparison framework with the model developed for this thesis.

The original Menon model is an Eulerian traffic flow model in which the air traffic is spatially aggregated into control volumes, which are line elements [49]. This model was historically one of the first models to pose the problem of traffic flow modeling using an Eulerian framework. It is based on the Daganzo *Cell Transmission Model* (CTM) [22; 23] in which the traffic flowing into a control volume changes the density of aircraft in that control volume and, hence, changes the outflow of the control volume. Several modifications of the original Menon model are made and outlined at the end of this subsection; we will thus refer to the modified version of the model as the *Modified Menon Model* (MMM). The model also incorporates ATC management, and handle merging and diverging air traffic flows. The model consists of two parts, the one-dimensional control volume model and the merge and diverge routing structure.

The one-dimensional control volume model models air traffic flow as a network of inter-connected control cells through which the air traffic flows. Aircraft counts in the network can be described by the discrete-time difference equation:

$$x_j(i+1) = x_j(i) + \tau_j[y_{j-1}(i) - y_j(i)]. \tag{3.5}$$

In the above equation, $x_j(i+1)$ is the aircraft count of control volume j at time $i+1$. The flow into j is $y_{j-1}(i)$, and $y_j(i)$ is equal to the flow out of j. The time step, τ_j, is computed by dividing the cell dimension, Ω_j, by the aircraft speed in the cell, v_j ($\tau_j = \Omega_j/v_j$). In other words, τ_j is the time an aircraft takes to travel through the cell.

The effects of delaying aircraft due to ATC action is accounted for by recirculating some of the air traffic flow in a control volume. The recirculated air traffic flow in control volume j is defined as u_j. The physical constraint on u_j is that at time i, it can not be greater than the existing flow in the cell or less than 0,

$$0 \leq \tau_j u_j(i) \leq x_j(i). \tag{3.6}$$

By including u_j and writing down the equation for y_j, the model can be written in the form of a linear, discrete-time dynamical system:

$$x_j(i+1) = a_j x_j(i) + \tau_j u_j(i) + \tau_j y_{j-1}(i), \tag{3.7}$$

$$y_j(i) = b_j x_j(i) - u_j(i). \tag{3.8}$$

The coefficients, a_j, b_j and τ_j handle the conversion between the air traffic flow, y_j, and the aircraft count, x_j. In other words, at a given time step, a_j is the portion of aircraft remaining in the volume, and b_j is the portion of air traffic flow leaving the volume. As was noted earlier, τ_j is the length of time needed for the aircraft to travel the length of the control volume. The coefficients are defined in terms of Ω_j, the control volume length, and v_j, the aircraft speed.

$$a_j = (1 - v_j \tau_j / \Omega_j), \quad b_j = v_j / \Omega_j, \quad \tau_j = \Omega_j / v_j. \tag{3.9}$$

The original Menon model assumes that velocity is constant within a given control volume. This means that a_j is always zero (see equation (13) in the original article [49]). That is, if there is no control from u_j, then all the aircraft in the volume travel to the subsequent volume on the next time step.

Intuitively, what is happening in equations (3.7) and (3.8) is that the aircraft count in a given control volume at time $i + 1$ depends on the number of aircraft in the volume at time i, the number of aircraft that flow into the volume, the number of aircraft that are recirculated and the number of aircraft that flow out of the volume. Over multiple time steps, aircraft will move through successive cells.

In a network of inter-connected control volumes, there may be points where air traffic coming from different directions merge into a single flow. This type of situation is referred to as a merge node. Furthermore, there may be points where the air traffic in one direction diverges into multiple flows. This type of situation is referred to as a diverge node. Because the nodes do not retain any aircraft, the conservation principle implies that for merge nodes, the resulting air traffic flow is the sum of all air traffic flows into that node. For example, if the air traffic flows q_{k-1} and q_{k-2} merge into q_k,

$$q_k = q_{k-1} + q_{k-2}. \tag{3.10}$$

Likewise, diverge nodes make use of the same conservation principle and the flow along a path from a diverge node is some proportion of the total flow coming into the diverge node. The proportion is defined as the divergence parameter, β, and is the ratio of aircraft traveling out of the diverge node along a given path over the aircraft traveling into the diverge node. In the following example, the air traffic flow diverges from the q_k to q_{k+1} and q_{k+2},

$$q_{k+1} = \beta q_k, \quad q_{k+2} = (1-\beta)q_k. \tag{3.11}$$

As mentioned earlier, since the MMM is implemented on a graph model of traffic flow constructed in the articles [64; 72] and discussed in Section 3.1.5, a number of modifications are made to improve the original Menon model described in the article [49].

1. The flights in the MMM are aggregated according to the links of the graph structure defined in Section 2.1 of Chapter 2, instead of the graph model presented in their original article [49]. This will ensure fairness of the comparison with the other models.

2. A link length (physical distance) is determined from flights in the data: flights in the data are aggregated according to the links in the graph. A link's entry and exit locations are determined by those flights' link entries and exits. The entry and exit locations are used in computing each link's length.

3. The MMM contains merge-diverge nodes. A merge-diverge node is one that has both merging and diverging flows at the same time. The original Menon model does not have such nodes.

4. A merge-diverge node can have n ($n \geq 2$) outflows, whose β values are determined from the data, whereas in the original Menon model n is limited to $n = 2$.

3.1.3 The PDE model

This section is based on the joint work in the article [73]. It presents the implementation of a PDE model based on earlier work available in the literature [9]. This model divides the airspace into line elements. These line elements are called paths and in practice often coincide with jetways. In practice, the same structure as before can be used. We represent a link on a path as a segment $[0, L]$ and we define $C(x, t)$ as the cumulated number of aircraft between distances 0 and x at time t. In particular, $C(0, t) = 0$, and $C(L, t)$ is the total number of aircraft in the path modeled by $[0, L]$ at time t. We make the additional assumption of a steady velocity profile $v(x) > 0$, which depicts the average velocity of aircraft flow at position x and time t. This velocity can be extracted from the travel time study performed in Section 2.1.4 in Chapter 2 for the CTM(L). Applying the conservation of mass to a control volume comprised between positions x and $x + h$, and letting h tend to 0, one easily finds the following relation between the spatial and temporal derivatives of $C(x, t)$ [9]:

$$\begin{cases} \frac{\partial C(x,t)}{\partial t} + v(x)\frac{\partial C(x,t)}{\partial x} = q(t) & (x, t) \in (0, L) \times (0, T] \\ C(x, 0) = C_0(x) & x \in [0, L] \\ C(0, t) = 0 & t \in [0, T], \end{cases} \tag{3.12}$$

where $q(t)$ represents the inflow at the entrance of the link ($x = 0$). Alternatively, $q(t)$ can be defined in terms of the density as $q(t) = \rho(0, t)v(0)$.

We can define the density of aircraft as the weak derivative of $C(x, t)$ with respect to x: $\rho(x, t) = \frac{\partial C(x,t)}{\partial x}$. The aircraft density is a solution to the partial differential equation:

$$\begin{cases} \frac{\partial \rho(x,t)}{\partial t} + \frac{\partial(\rho(x,t)v(x,t))}{\partial x} = 0 & (x, t) \in (0, L) \times (0, T] \\ \rho(x, 0) = \rho_0(x) & x \in [0, L] \\ \rho(0, t) = \frac{q(t)}{v(0)} & t \in [0, T]. \end{cases} \tag{3.13}$$

or in a nonconservative form:

$$\begin{cases} \frac{\partial \rho(x,t)}{\partial t} + v(x)\frac{\partial \rho(x,t)}{\partial x} + v'(x)\rho(x,t) = 0 & (x,t) \in (0,L) \times (0,T] \\ \rho(x,0) = \rho_0(x) & x \in [0,L] \\ \rho(0,t) = \frac{q(t)}{v(0)} & t \in [0,T]. \end{cases} \tag{3.14}$$

This is a linear advection equation with positive velocity $v(x)$ and a source term:

$$v'(x)\rho(x,t).$$

Clearly, these two partial differential equations are equivalent and model the same physical phenomenon. In this chapter, we will use the latter for control, as it enables us to impose constraints in terms of aircraft density. We will use the former for simulation and comparison because the aircraft count is more readily available from experimental data.

Now that the model has been defined on one link, we will extend it to a network. We consider a junction with m incoming links numbered from 1 to m and n outgoing links numbered from $m+1$ to $m+n$; each link k is represented by an interval $[0, L_k]$ (Figure 3.1). One can see that any network is composed of a number of such junctions. We define an allocation matrix $M(t) = (m_{ij}(t))$ for $1 \leq i \leq m,\ m+1 \leq j \leq m+n$, where $0 \leq m_{ij}(t) \leq 1$ denotes the proportion of aircraft from incoming link i going to the outgoing link j. We also require $\sum_{j=m+1}^{m+n} m_{ij}(t) = 1$ for $1 \leq i \leq m$. The system of partial differential equations on the network can be written as:

$$\begin{cases} \frac{\partial \rho_k(x,t)}{\partial t} + v_k(x)\frac{\partial \rho_k(x,t)}{\partial x} + v_k'(x)\rho_k(x,t) = 0 & 1 \leq k \leq m+n, (x,t) \in (0,L_k) \times (0,T] \\ \rho_k(x,0) = \rho_{0,k}(x) & x \in [0,L_k] \\ \rho_i(0,t) = \frac{q_i(t)}{v_i(0)} & 1 \leq i \leq m, t \in [0,T] \\ \rho_j(0,t) = \frac{\sum_{i=1}^{m} m_{ij}(t)\rho_i(L_i,t)v_i(L_i)}{v_j(0)} & m+1 \leq j \leq m+n, t \in [0,T]. \end{cases} \tag{3.15}$$

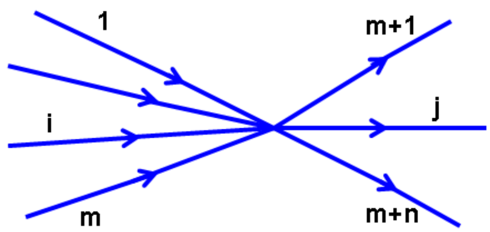

Figure 3.1: A junction with m incoming links ($1 \leq i \leq m$) and n outgoing links ($m+1 \leq j \leq m+n$).

We will now show that on such a network, the preceding system of partial differential equations admits a unique solution, hence that the problem is well-posed.

First, we consider the case of a single link $[0, L]$. Since the velocity is always positive, a boundary condition shall be set on the left ($x = 0$) but not on the right ($x = L$). Using classical partial differential equations techniques, more precisely the theory of characteristics, to compute the solution and prove the existence and energy methods for the uniqueness, it can be shown that the advection equation will have a unique solution on this interval (for example, see [43] or [9] for a proof). On a network, this ensures the existence and uniqueness of a solution on the incoming links. For the outgoing links, we need to impose a boundary condition on the left, that is, immediately after the junction. This is done using the coefficients of the allocation matrix. Indeed for the j-th outgoing link, the density at the origin will be related to the densities at the right extremity of the incoming links by:

$$\rho_j(0,t) = \frac{\sum_{i=1}^{m} m_{ij}(t)\rho_i(L_i,t)v_i(L_i)}{v_j(0)}.$$

Now the advection equation on each outgoing link has only one solution, thus uniquely defining a density on both the incoming and outgoing links. Therefore, the problem for any network, which is made of several such junctions, is well-posed.

The Lax-Wendroff scheme is applied to the preceding partial differential equation. A discrete grid is used on the domain $[0, L] \times [0, T]$:

$$x_a = \frac{aL}{M}, \quad 0 \leq a \leq M \text{ and } t_b = \frac{bT}{N}, \quad 0 \leq b \leq N$$

and

$$\Delta x = \frac{L}{M}, \Delta t = \frac{T}{N}.$$

The Lax-Wendroff scheme (see [44]) is based on the second order Taylor series expansion of $C(x, t)$,

$$C(x, t_{b+1}) = C(x, t_b) + (\Delta t)C_t(x, t_b) + \frac{1}{2}(\Delta t)^2 C_{tt}(x, t_b) + \dots$$

Given that $C(x, t)$ is a solution of the partial differential equation above, we have:

$$C_t(x, t) = -v(x)C_x(x, t) - v'(x)C(x, t),$$

$$C_{tt}(x, t) = -v(x)C_{xt}(x, t) - v'(x)C_t(x, t).$$

If we differentiate the expression of $C_t(x, t)$ with respect to x, we obtain:

$$C_{xt}(x, t) = -v(x)C_{xx}(x, t) - v''(x)C(x, t) - 2v'(x)C_x(x, t),$$

which yields:

$$C_{tt}(x, t) = v^2(x)C_{xx}(x, t) + 3v(x)v'(x)C_x(x, t) + (v(x)v''(x) + (v'(x))^2)C(x, t).$$

Using the preceding expressions of $C_t(x, t)$ and $C_{tt}(x, t)$ in the Taylor series expansion, we find:

$$
\begin{aligned}
C(x,t_{b+1}) = C(x,t_b) &- (\Delta t)v(x)C_x(x,t_b) - (\Delta t)v'(x)C(x,t_b) \\
&+ \frac{1}{2}(\Delta t)^2(v^2(x)C_{xx}(x,t_b) + 3v(x)v'(x)C_x(x,t_b) \\
&+ (v(x)v''(x) + (v'(x))^2)C(x,t_b)) + \dots
\end{aligned}
$$

Then we replace the spatial derivatives by central finite difference approximations:

$$
\begin{aligned}
C_x(x,t) &\leftrightarrow \frac{C^{a+1,b} - C^{a-1,b}}{2\Delta x} \\
C_{xx}(x,t) &\leftrightarrow \frac{C^{a-1,b} - 2C^{a,b} + C^{a+1,b}}{(\Delta x)^2}.
\end{aligned}
$$

We eventually obtain the Lax-Wendroff scheme:

$$
\begin{aligned}
C^{a,b+1} = &\left(1 - (\Delta t)v'(x_a) + \frac{(\Delta t)^2}{2}(v'(x_a))^2\right) C^{a,b} \\
&+ \frac{\Delta t}{2\Delta x} v(x_a)\left(\frac{3}{2}(\Delta t)v'(x_a) - 1\right)(C^{a+1,b} - C^{a-1,b}) \\
&+ \frac{1}{2}\left(\frac{\Delta t}{\Delta x}\right)^2 v^2(x_a)(C^{a-1,b} - 2C^{a,b} + C^{a+1,b}).
\end{aligned}
\tag{3.16}
$$

The initial condition implies:

$$
C^{a,0} = \frac{1}{2\Delta x}\int_{x_{a-1}}^{x_{a+1}} C_0(x)dx \text{ for } 0 \le a \le M. \tag{3.17}
$$

The boundary conditions are implemented using 2 ghost-cells on the left and right of the spatial domain. Given that the velocity is always positive, the boundary conditions can only be prescribed on the left; we use zero-order extrapolation for the right boundary condition:

$$
C^{-1,b} = \frac{1}{\Delta t}\int_{t_b}^{t_{b+1}} \frac{q(t)}{v(0)}dt \text{ and } C^{M+1,b} = C^{M,b} \text{ for } 1 \le b \le N. \tag{3.18}
$$

Finally, when choosing the space and time steps, the *Courant-Friedrichs-Lewy* (CFL) condition has to be verified:

$$
\left|\frac{v(x)\Delta t}{\Delta x}\right| \le 1 \text{ for } x \in [0,L].
$$

where $\left|\frac{v(x)\Delta t}{\Delta x}\right|$ is called the *Courant* number. Since the Lax-Wendroff scheme is increasingly accurate as the Courant number approaches to 1, the time and space steps should be chosen so that:

$$\frac{\Delta t}{\Delta x} \text{ is slightly smaller than } \frac{1}{\sup_{x\in[0,L]} v(x)} \quad .$$

The update equation (3.16) with initial condition (3.17) and boundary condition (3.18) can thus be used as a constitutive model for traffic flow, which is a discretization of the system (3.15).

3.1.4 The 2D Menon Model (MM2D)

The section is based on the article [50]. The *2D Menon Model* (MM2D) first partitions the airspace into sectors called *Control Volumes* (CVs) or *Surface Elements* (SELs). These SELs are formed by equal increments in latitudes and longitudes. To simplify the analysis, all the SELs are treated as equal squares. Each SEL is then partitioned into eight streams, which discretize the notion of directions within a SEL (Figure 3.2; Source: [50]). Conceptually, a ninth stream will constitute the input/output of a SEL from an underlying airport or from altitudes above or below the desired range of study. Each SEL will have three, five or eight entry/exit points with its neighboring SELs, depending on its location (corner, border or center), in addition to an eventual airport beneath it.

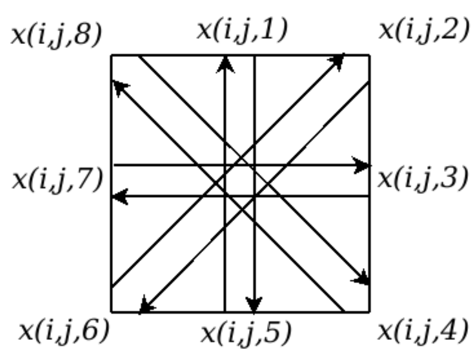

Figure 3.2: Traffic flow directions in an SEL (i, j).

The MM2D discretizes the time into steps of increments τ. In each time step, a number of aircraft exit from the streams of each SEL. The flow divergence parameter β is a five-dimensional variable. It is time dependent, and determines the percentage of aircraft that switched from streams m to n in the SEL (i, j). The inertia parameter a is four-dimensional. It characterizes the proportion of aircraft that will stay in a certain stream s of a SEL (i, j) in from time k to $k + 1$. The dynamics of the model can be represented as follows:

$$x_{(i,j,1)}(k+1) = a_{(i,j,1)} \sum_{m=1}^{8} \beta_{(i,j,1,m)} x_{(i,j,m)}(k) + \tau y_{(i-1,j,1)}(k) + \tau q^{\text{depart}}_{(i,j,1)}(k), \tag{3.19}$$

where $x_{(i,j,1)}(k + 1)$ represents the predicted number of aircraft in stream 1 of SEL(i, j) at time step $k + 1$; $a_{(i,j,1)}(k)$ represents the fraction of aircraft that will stay in stream 1 of SEL(i, j) after time step k; $\beta_{(i,j,1,m)}(k)$ represents the portion of aircraft that switched from streams 1 to stream m at time step k before leaving SEL(i, j) at time step $k + 1$; $y_{(i-1,j,1)}(k)$ represents the flow at time step k of aircraft into stream 1 of SEL(i, j) coming from SEL$(i - 1, j)$; $q^{\text{depart}}_{(i,j,1)}(k)$ represents the flow at time step k of aircraft into stream 1 of SEL(i, j) coming from an airport located beneath it.

The dynamics of other streams in the SEL can be expressed in a similar way, simply by replacing the number 1 in the index with other numbers (2, 3, $\cdots$, 8). The output flow y is computed as follows:

$$y_{(i,j,m)}(k) = \big(1 - a_{(i,j,m)}\big) \sum_{n=1}^{8} \beta_{(i,j,m,n)} x_{(i,j,n)}(k).$$

The implementation of the MM2D relies on a two-dimensional geometric partition of the airspace, which is different from the other three models described in the previous sections. The original work of the authors of the MM2D [50] did not mention how to identify the parameters (a and β in equation (3.19) of the model). We use one year of ASDI/ETMS data for this identification as follows: from the recorded data, we compute a and β for each day in a full year of data, and take the mean of the a's and the normalized (by the rule of conservation of flows) mean of the β's as the parameters a and β, respectively.

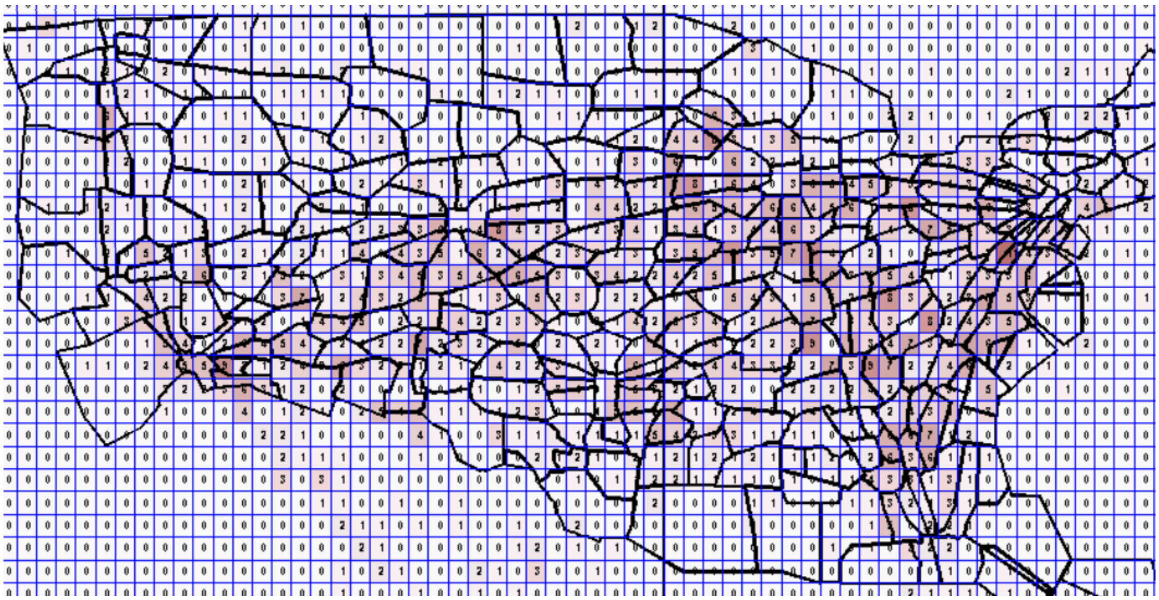

Figure 3.3: An implementation of the MM2D model. Numbers represent the amount of flights in the corresponding SELs.

Figure 3.3 shows the implementation of the MM2D model, which was performed as part of this thesis. As can be seen from the figure, the two dimensionality of the model is captured by the rectangular tessilation of the airspace.

3.1.5 A benchmark scenario for comparison of the models

For the comparison, three of the four models described above (MMM, the CTM(L) and the PDE model) are implemented on the same aggregate traffic flow graph model depicted in Figure 3.5. The construction of the graph is outlined in Section 2.1 of Chapter 2. The MM2D must be implemented on its own flow structure because of the two-dimensional nature of the model (see Figure 3.3). The portion of airspace studied for this comparison is depicted in Figure 3.4 (FACET courtesy of NASA Ames [14]), and consists of 75 sectors of the Oakland, Los Angeles, Seattle, Salt Lake City, Denver, Albuquerque and Oakland Oceanic Centers. The graph identification procedure relies on the notion of a *path*, illustrated in Figure 2.2. We use a full year of ASDI/ETMS data for this identification. The complete resulting graph of the entire NAS is shown in Figure 3.5 which covers the region in Figure 3.4. For the MMM, we will use β splits at the nodes where traffic is diverging, following the procedure outlined in the original article [49] and modified according to Section 3.1.2. For the two other models, we will use the notion of paths, linking any origin to any destination in the graph. This idea is sometimes referred to as the *colored flow paradigm*, which is an example of multicommodity flows in the network flow and combinatorial optimization literature [4]. This enables us to avoid the identification of the β split parameters and the resulting inaccuracies of this model, and most importantly, this uses the fact that the destinations of the aircraft are $known$ before take off.

Using the terminology presented before and illustrated in Figure 2.2, the graph used for this study has 648 paths, 437 links, 12,574 cells (MMM), 39,776 cells (CTM(L)), and 128,500 grid points (PDE model).

The parameter identification used for the MMM is straightforward: following the work in [49], we average all the velocities of all aircraft over one year for the airspace of interest. For the CTM(L) and the PDE model, we do it path by path. An example of velocity fit for one path is shown in Figure 3.6. The β split coefficients used for the MMM are computed by dividing the number of aircraft on a branch from a split by the total number of aircraft exiting the split. The cell dimension in the MMM is computed as the distance traveled by an aircraft in one minute (the time step set for the simulation). Since the average velocity is 480 knots, this gives a cell dimension of about 15 km.

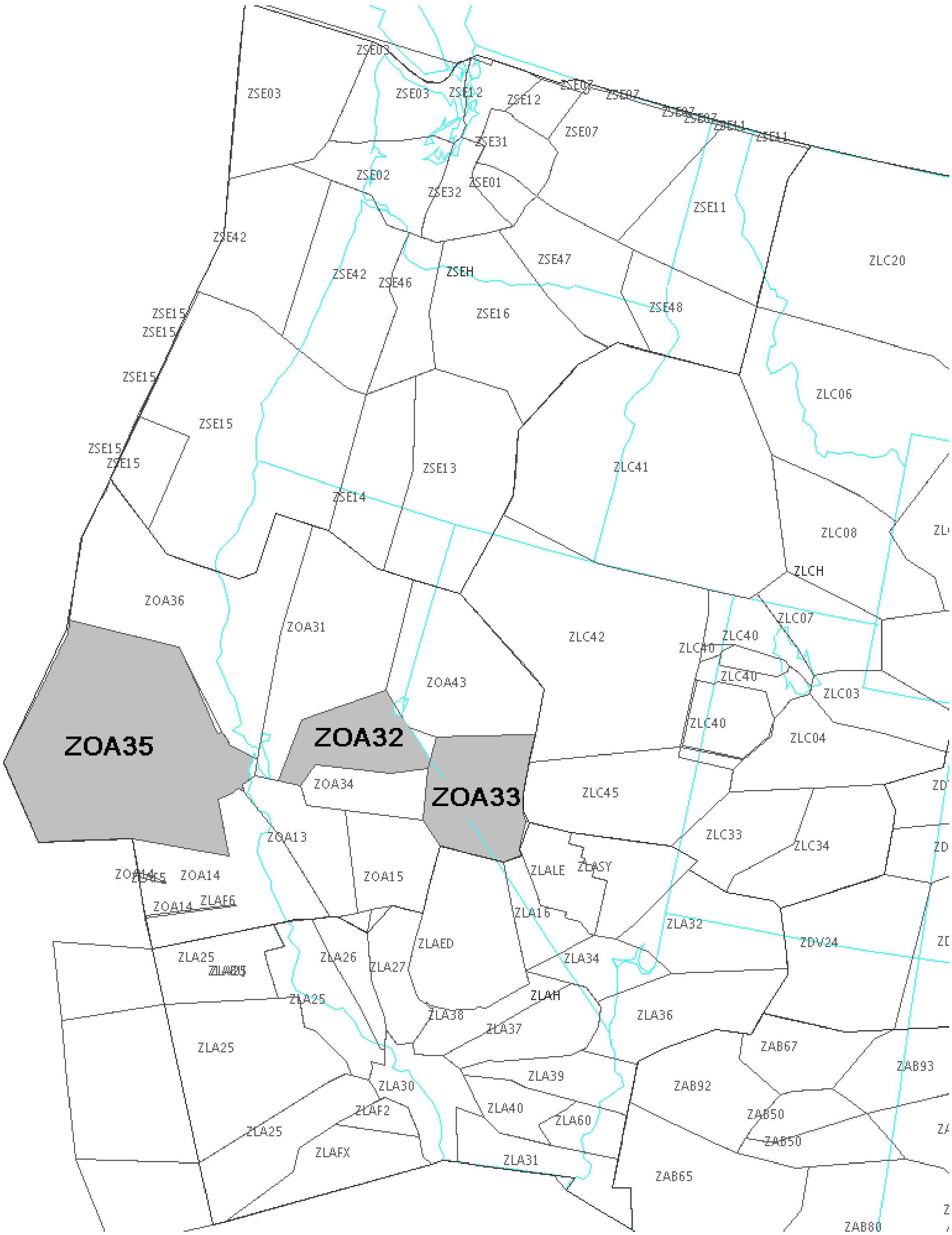

Figure 3.4: Map of the portion of airspace considered in this study: Oakland ARTCC (ZOA), Los Angeles ARTCC (ZLA), Salt Lake City ARTCC (ZLC), Seattle ARTCC (ZSE), a portion of Denver ARTCC (ZDV), a portion of Albuquerque ARTCC (ZAB), and a portion of Oakland Oceanic ARTCC. Map obtained using the software FACET.

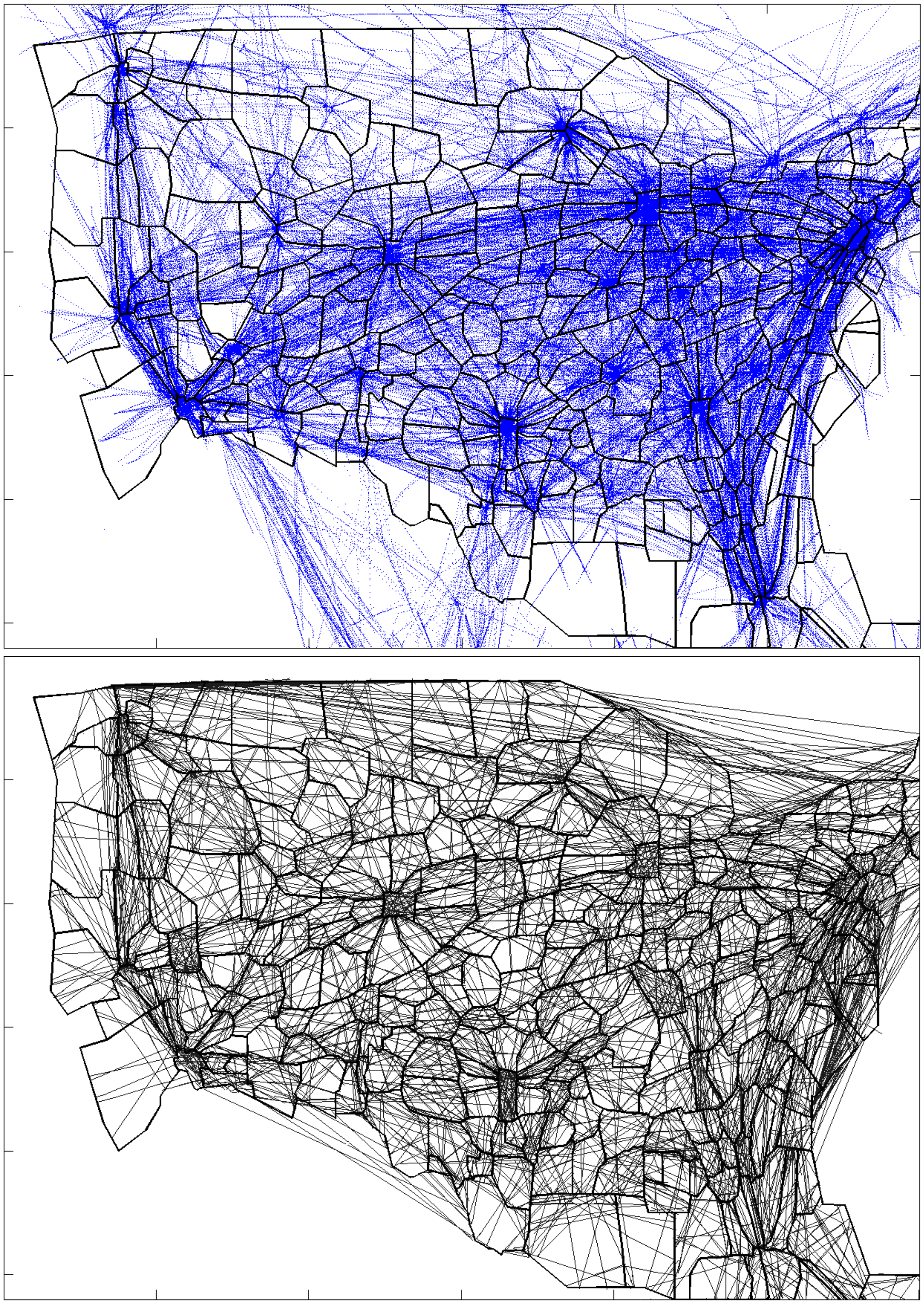

Figure 3.5: **Top:** An example of flight tracks for the full NAS. **Bottom**: Graph model representing the flow patterns above.

For the CTM(L), the cell dimension is time-based and is one minute in length.

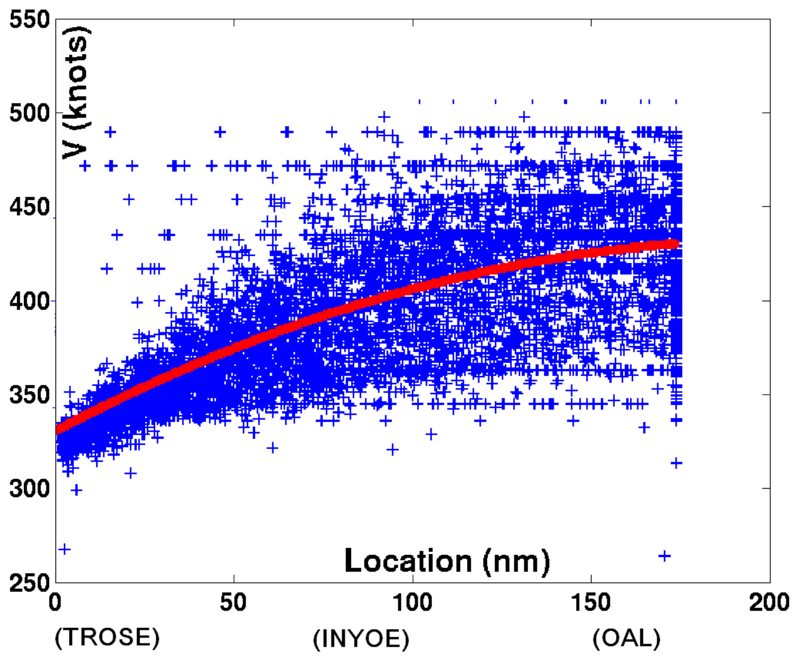

Figure 3.6: Aggregation of velocities along a path. x-**axis**: positions from the starting point of a path in the model; y-**axis**: velocities in knots. A third order curve fit is used for the velocity profile. Typically, flights going through this path (passing through way-points TROSE-INYOE-OAL) pop-up from low altitude airspace and climb up to high altitudes.

3.2 Comparison of the respective performance of the models

Following the procedure of the previous chapter, the models are validated against ASDI/ETMS data, and their respective performances are compared. The validation procedure consists in taking filed flight plans (origin-destination and schedule for each aircraft) as inputs, performing a forward simulation of traffic for the full NAS (with the four models), and comparing the corresponding results with the recorded data. The input to each model is the number of aircraft entering the considered region (Figure 3.4). The predicted states (for example, sector counts) are computed from each model and compared with the recorded data. Simulations are performed from 8:00 a.m. GMT on January 24th, 2005 to 8:00 a.m. GMT on January 25th, 2005.

Sector counts predicted by the four models are compared with the recorded ASDI/ETMS data. Our study shows that all the sector counts predicted by the four models and ASDI/ETMS data differ from the true counts by noise of a non-negligible magnitude for the following reasons: (i) for the CTM(L), the travel time on a link in the network is computed as the average travel time for all flights in the data set used for the identification; (ii) for the PDE model, the velocity profile of each path is filtered from sampled velocities and only several modes are preserved; (iii) for the MMM, the split ratios are computed from historical data, which usually does not match the instantaneous ratios for a specific day, and also the MMM assumes a uniform velocity across the whole network; (iv) for the MM2D, the parameters a and β are computed from historical data, which differ from the actual a and β for a specific day.

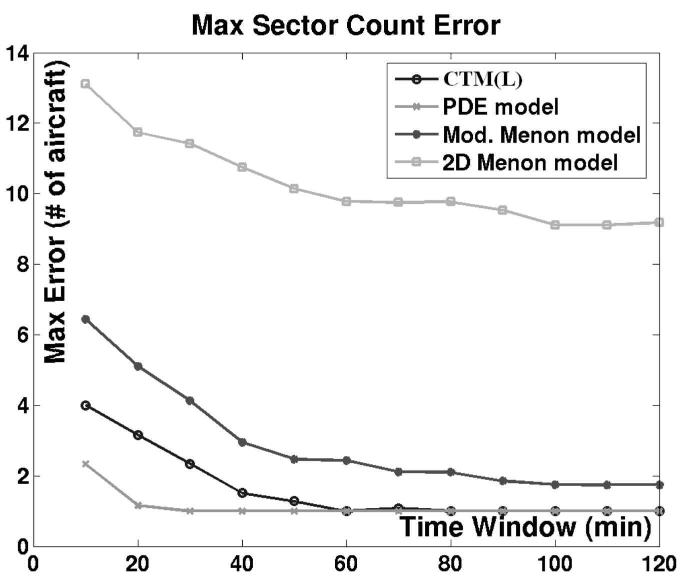

Figure 3.7: Maximum sector count error of ZOA33, between simulation of the models and ASDI/ETMS data (after filtering). The maximum error decreases as the time window increases.

A *moving average filter* (MAF) technique is used to filter the sector counts for both the recorded ASDI/ETMS data and the models' simulation data, similar as in Section 2.3.1 in Chapter 2. Figure 3.7 shows the results obtained when using different sizes of time window for the four models in sector ZOA33. The maximum sector count error between the filtered ASDI/ETMS data and the filtered simulation results of the models decreases when the size of the time window increases. In the extreme case, in which the time window is 24 hours (the simulation time span for our study), the error between the filtered recorded data and the filtered simulation results tends to be zero. This occurs because the error is the difference between the average sector count from the ASDI/ETMS data and the simulation for a full day, which is very small in general for this case. For example, for the PDE model, the maximum errors are below two when the time window is 20 minutes. Above 20 minutes, increasing the time window does not help to significantly decrease the maximum error, and does not make sense for the problem of interest as well.

Figure 3.8 shows the predicted and actual sector counts as a function of time in three sectors: heavily loaded sector ZOA33, medium loaded sector ZOA32 and low loaded sector ZOA35. The data shown in the figure is filtered by MAF. From the figures we can see that all the models correctly predict the trends of sector counts.

3.3 Controller design

One of the contributions of this thesis is the development and implementation of optimal control schemes for the CTM(L) developed in Chapter 2. We also outline for completeness the strategies which would have to be applied for the other models (presented in the previous section). Their implementation and analysis are outside the scope of this work.

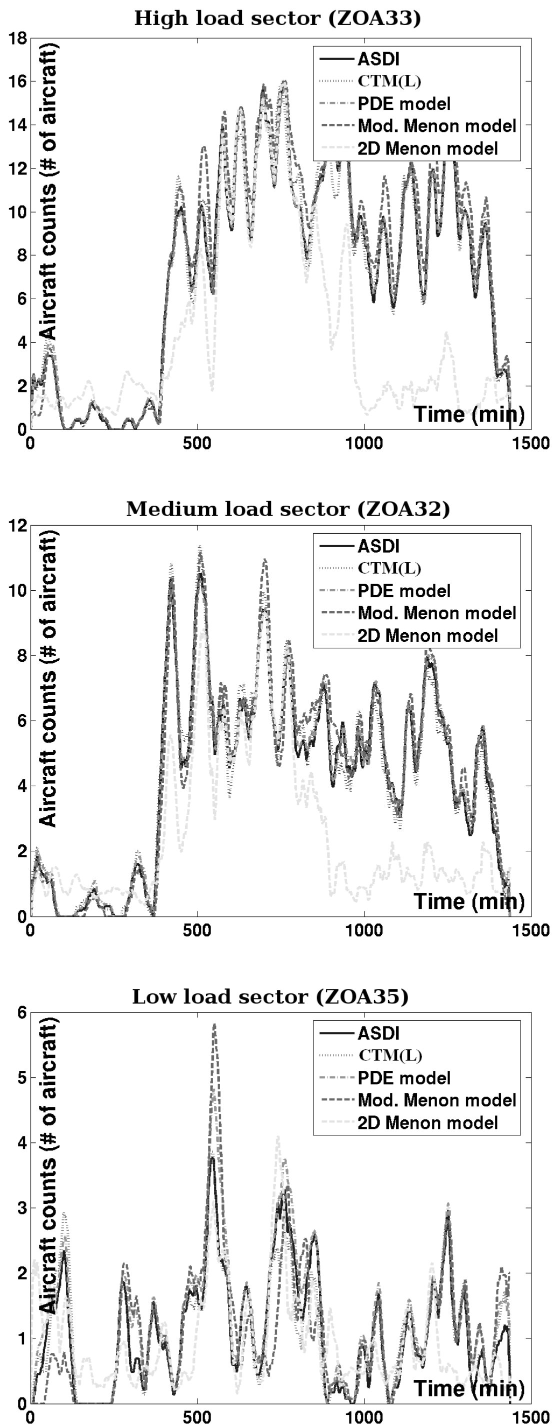

Figure 3.8: Comparison between the predictions of aircraft sector counts predicted by the four models and the actual ASDI/ETMS data counts. Curves represent the processed sector counts after filtering. 0 min corresponds to midnight.

3.3.1 The Large-capacity Cell Transmission Model

The present section formulates the problem of regulating the aircraft count in different sectors under a legal threshold so that high level TFM can be applied to comply with FAA standards.

The time horizon of the problem (order of magnitude of two hours) is discretized in N time steps of length τ. Therefore, τ is the time spent by one aircraft in one cell in absence of ATC action. The state of the system at time step $k \in \{0, \cdots, N\}$ is characterized by the number of aircraft in each cell and represented by the vector $x_k \in \mathbb{R}^n$, where n is the number of cells in the network. The control variables are denoted $u_k \in \mathbb{R}^n$ for $k \in \{0, \cdots, N\}$, where u_k represents the number of aircraft held in each cell at time step k. The input to the system at time step $k \in \{0, \cdots, N\}$ consists of the aircraft entering the network, and the number of aircraft entering each cell at time step k is represented by the vector $f_k \in \mathbb{R}^n$. Note that, unlike in a standard control framework terminology, we do not have control over the input, f_k, which is an "exogenous forcing" from outside the system.

Using a standard optimal control framework such as in [16], the dynamics (3.3)-(3.4) becomes part of the constraints of the *mixed integer linear program* (MILP) formulation:

$$
\begin{array}{lll}
\mathbf{min} & \sum_{k=0}^{N} c^T x_k & \\
\mathbf{s.t.} & E x_k + L u_k \leq M, & k \in \{0, \cdots, N-1\} \\
 & x_N \in \chi_f & \\
 & x_{k+1} = A x_k + B^f f_k + B^u u_k, & k \in \{0, \cdots, N\} \\
 & x_0 = B^f f_0, &
\end{array}
\tag{3.20}
$$

where $\chi_f \subseteq \mathbb{R}^n$ is a terminal polyhedron region, and the matrices E, L and M represent the constraints on the system: the sector counts must remain under a legal threshold, and the number of aircraft held in a cell cannot be greater than the number of aircraft in that cell. The objective of the problem is to minimize the total travel time; therefore, $c \in \mathbb{R}^n$ is the vector $[\tau, \tau, \ldots, \tau]^T$. The formulation of (3.20) is a MILP because (i) the objective function and the constraints are all linear and (ii) the state variable x and control variable u are actually integer in practice.

Implementation

In order to solve (3.20) in practice, we need to encode it in a computationally efficient manner, which is now presented. Flights are clustered on *paths*, as explained in section 3.2. The set P of paths is determined from the data, as well as the number n_p of cells along path $p \in P$. Within each path, cells are indexed so that flights go through cells of increasing index numbers. The notation for the state of the system, the input and the control variables are adapted to take the paths into account. The state is reindexed, such that $x_{k,p,i}$ now denotes the number of aircraft in cell $i \in \{1, \cdots, n_p\}$ of path $p \in P$ at time step $k \in \{0, \cdots, N\}$. The corresponding control variables are denoted as $u_{k,p,i}$ for $k \in \{0, \cdots, N\}$, $p \in P$ and $i \in \{1, \cdots, n_p\}$, where $u_{k,p,i}$ represents the number of aircraft held in cell i of path p at time step k. The forcing inputs to the system are denoted as $f_{k,p}$, for $k \in \{0, \cdots, N\}$ and $p \in P$, where $f_{k,p}$ represents the number of aircraft entering the network on path p at time step k.

The sector capacity (i.e. the maximum number of aircraft allowed in the sector) is enforced independently for a set S of different sectors. These sectors, referred to as sector-capacity-constrained sectors, have capacities $C_s,\ s \in S$. The adapted MILP formulation of the problem is as follows:

$$
\begin{array}{lll}
\textbf{min} & \tau \sum_{k=0}^{N} \sum_{p\in P} \sum_{i=1}^{n_p} x_{k,p,i} & \\
\textbf{s.t.} & \sum_{(p,i)\in I_s} x_{k,p,i} \le C_s, & k\in\{0,\cdots,N\},\ s\in S \\
 & 0 \le u_{k,p,i} \le x_{k,p,i}, & k\in\{0,\cdots,N\},\ p\in P,\ i\in\{1,\cdots,n_p\} \\
 & x_{k+1,p,i} = x_{k,p,i-1} + u_{k,p,i} - u_{k,p,i-1}, & k\in\{0,\cdots,N-1\},\ p\in P,\ i\in\{2,\cdots,n_p\} \\
 & x_{k,p,1} = f_{k,p} + u_{k,p,1}, & k\in\{0,\cdots,N\}, p\in P \\
 & x_{0,p,i} = 0, & p\in P,\ i\in\{2,\cdots,n_p\} \\
 & x_{k,p,i} \in \mathbb{Z}, & k\in\{0,\cdots,N\},\ p\in P,\ i\in\{1,\cdots,n_p\},
\end{array}
\tag{3.21}
$$

where I_s is the set of cells (represented by a path p and a cell number along path p) physically present in sector $s \in S$. The integrality of the number of aircraft in each cell ensures the integrality of the number of aircraft held in each cell, since the input of the system is assumed to be integer.

LP relaxation of the MILP formulation

Because problem (3.21) cannot be solved in polynominal time deterministically, it is relaxed to a *linear program* (LP), which is faster to solve in practice, and theoretically polynomial time solvable. [1]

The relaxed MILP (i.e. the LP) was solved on a statistical sample of 1,000 different sets of input parameters. 85 percent of the runs led to an integer solution. For the remaining 15 percent, the optimal solution of the LP (OPT_{LP}) was compared to the optimal solution of the corresponding MILP (OPT_{MILP}). The integrality gap α, (i.e. $OPT_{\text{MILP}} = \alpha \cdot OPT_{\text{LP}}$), was always smaller then 1.0015. However, the corresponding solutions are fractional, thus impractical. Several techniques

[1] We did not assess the usefulness of the guaranteed computational complexity of LP explicitly in the present case. Indeed, the fact that LPs are polynomial time solvable can only be used with a thorough analysis of the constant mutiplying of the corresponding higher order monomial.

might apply in the future to alleviate this difficulty, in particular LP rounding, which would yield to a suboptimal but integer solution.

On one hand, there is no guarantee of integrality of the LP solution, but on the other hand, the running time of computing the MILPs solution is not guaranteed. Despite the limitations of these two approaches, one conclusion can still be guaranteed from the LP approach: when it returns no solution, it provides a certificate of infeasibility of the corresponding TFM problem with guaranteed running time. Also, given the structure of the problem, minimizing the total travel time is equivalent to minimizing the amount of delay assigned to the aircraft. Therefore, the number of holding patterns provided by the LP solution is the lower bound of the number of holding patterns for which there may exist a physical solution. In other words, no air traffic control can enforce the sector count limitations with less holding patterns than the number of holding patterns provided by the LP relaxation.

3.3.2 The PDE model

In this section, we study an optimal flow control problem for a network using the PDE model derived in Section 3.1.3. This follows the earlier work performed on control of PDE models of the NAS, in particular [9]. A similar case for highway networks was studied in [30]. We try to mitigate congestion on the network by acting on the coefficients of the allocation matrix. To evaluate the gradient of the objective function, we implement a continuous adjoint method to evaluate its performance. We consider the following problem:

$$
\begin{array}{lll}
\textbf{min} & H(m_{ij}) = \sum_{k=1}^{m+n} \int_0^T \int_0^{L_k} \rho_k(x,t)dxdt & \\
\textbf{s.t.} & \frac{\partial \rho_k(x,t)}{\partial t} + v_k(x)\frac{\partial \rho_k(x,t))}{\partial x} + v_k'(x)\rho_k(x,t) = 0, & 1 \le k \le m+n, (x,t) \in (0,L_k) \times (0,T] \\
 & \rho_k(x,0) = \rho_{0,k}(x), & x \in [0,L_k] \\
 & \rho_i(0,t) = \frac{q_i(t)}{v_i(0)}, & 1 \le i \le m, t \in [0,T] \\
 & \rho_j(0,t) = \frac{\sum_{i=1}^m m_{ij}(t)\rho_i(L_i,t)v_i(L_i)}{v_j(0)}, & m+1 \le j \le m+n, t \in [0,T] \\
 & 0 \le m_{ij}(t) \le 1, & 1 \le i \le m, \ m+1 \le j \le m+n \\
 & \sum_{j=m+1}^{m+n} m_{ij}(t) = 1, & 1 \le i \le m \\
 & \rho_k(x,t) \le \rho_k^{\max}, & 1 \le k \le m+n.
\end{array}
\tag{3.22}
$$

Minimizing this functional is equivalent to maximizing the outflow of the network; indeed the value of H represents the total amount of time aircraft spent in the network. The control variables are the coefficients of the allocation matrix $(m_{ij}(t))$. This is, in fact, a case of boundary control since as explained earlier, the density at the left of an outgoing link is directly related to the value of $(m_{ij}(t))$ by:

$$\rho_j(0,t) = \frac{\sum_{i=1}^m m_{ij}(t)\rho_i(L_i,t)v_i(L_i)}{v_j(0)}, \ m+1 \le j \le m+n \text{ and } t \in [0,T].$$

The first two constraints are used to make sure that the model is realistic; all the aircraft have to leave an incoming link and enter an outgoing link. The third constraint implements a maximum density that can not be exceeded for each link.

Adjoint methods were first introduced in the late 1980s as a tool for shape optimization, in particular aircraft design [37]. The direct approach, which consists of calculating the gradient of the cost functional using finite differences, is only possible when the number of control variables

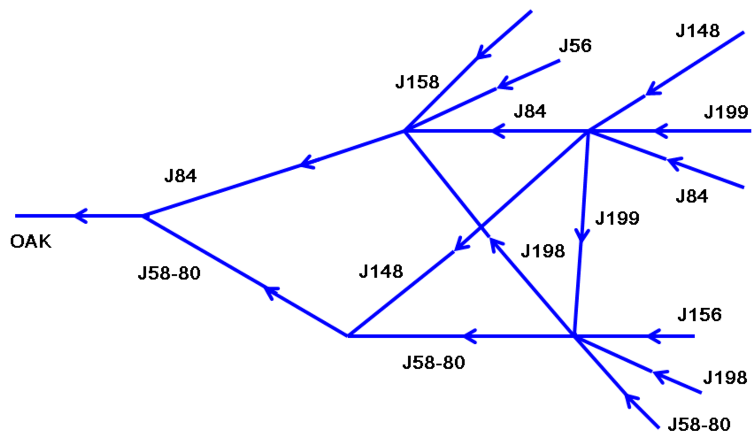

Figure 3.9: Network used for the optimization containing 16 links and 5 junctions. The links are numbered according to the jetways they represeComparison between the predictions of aircraft sector countsnt which are part of the ZOA ARTCC.

is small. In most real life problems, this number is too large, making this approach unfeasible. A more efficient way of calculating gradients is to use the adjoint equations and boundary conditions, which can be solved using numerical schemes to yield the gradient of the cost functions.

We will use this technique to determine the gradient of the functional H. We consider links of length L_k, which in our example will be equal to the actual length of the corresponding air traffic network links considered. We bring the reader's attention to the fact that the following results can be applied to any functional

$$H(m_{ij}) = \sum_{k=1}^{m+n} \int_0^T \int_0^{L_k} h_k(\rho_k(x,t))dxdt,$$

for any functions $h_k(x)$. Note that m_{ij} does not appear explicitly in the functional, but implicitly, through the constraints of (3.22).

Continuous Adjoint Method

We will present the continuous adjoint method in this section. We start by forming the variation in the cost function:

$$\begin{aligned}\delta H &= H(m_{ij}+\delta m_{ij}) - H(m_{ij})\\ &= \sum_{i=1}^{m}\int_0^{L_i}\int_0^T \rho_i(x,t)dxdt + \sum_{j=m+1}^{m+n}\int_0^{L_j}\int_0^T (\rho_j(x,t)+\delta\rho_j(x,t))dxdt\\ &\quad -\sum_{i=1}^{m}\int_0^{L_i}\int_0^T \rho_i(x,t)dxdt - \sum_{j=m+1}^{m+n}\int_0^{L_j}\int_0^T \rho_j(x,t)dxdt\\ &= \sum_{j=m+1}^{m+n}\int_0^{L_j}\int_0^T \delta\rho_j(x,t)dxdt.\end{aligned}$$

We then compute the variation of the constraint equation, in our case the partial differential equation verified by the density, which yields for the outgoing links (the incoming links not being affected by the control):

$$\frac{\partial\delta\rho_j(x,t)}{\partial t} + v_j(x)\frac{\partial\delta\rho_j(x,t)}{\partial x} + v_j'(x)\delta\rho_j(x,t) = 0 \text{ for } m+1 \le j \le m+n,$$

with the initial condition:

$$\delta\rho_j(x,0) = 0,$$

and the boundary condition:

$$\delta\rho_j(0,t) = \frac{\sum_{i=1}^{m}\delta m_{ij}(t)\rho_i(L_i,t)v_i(L_i)}{v_j(0)}.$$

Since the variation of the cost function depends on $\delta\rho_j$, we need to add a term to the variation of the cost function to eliminate this dependence. If $\lambda_j(x,t)$ is an arbitrary function, we can add the scalar product of $\lambda_j(x,t)$ with the previous equation since it is equal to zero:

$$\delta H = \delta H + \sum_{j=m+1}^{m+n}\int_0^{L_j}\int_0^T \lambda_j(x,t)\left(\frac{\partial\delta\rho_j(x,t)}{\partial t} + v_j(x)\frac{\partial\delta\rho_j(x,t)}{\partial x} + v_j'(x)\delta\rho_j(x,t)\right)dxdt.$$

After integrating by parts:

$$\delta H = \delta H + \sum_{j=m+1}^{m+n} \left\{ \int_0^{L_j} \int_0^T \left(-\frac{\partial \lambda_j(x,t)}{\partial t} - \frac{\partial (v_j(x)\lambda_j(x,t))}{\partial x} + v_j'(x)\lambda_j(x,t) \right) \delta\rho_j(x,t) dx dt \right.$$
$$\left. + \int_0^{L_j} [\lambda_j(x,t)\delta\rho_j(x,t)]_0^T dx + \int_0^T [\delta\rho_j(x,t) v_j(x)\lambda_j(x,t)]_0^{L_j} dt \right\}.$$

Assembling the terms, we obtain:

$$\delta H = \sum_{j=m+1}^{m+n} \int_0^{L_j} \int_0^T \left(-\frac{\partial \lambda_j(x,t)}{\partial t} - \frac{\partial (v_j(x)\lambda_j(x,t))}{\partial x} + v_j'(x)\lambda_j(x,t) + 1 \right) \delta\rho_j(x,t) dx dt$$
$$+ \sum_{j=m+1}^{m+n} \left(\int_0^{L_j} [\lambda_j(x,t)\delta\rho_j(x,t)]_0^T dx + \int_0^T [\delta\rho_j(x,t) v_j(x)\lambda_j(x,t)]_0^{L_j} dt \right).$$

In order to eliminate the dependence of δH on $\delta\rho_j$, we make the following choice for λ_j:

$$\begin{cases} \frac{\partial \lambda_j(x,t)}{\partial t} + v_j(x)\frac{\partial \lambda_j(x,t)}{\partial x} = 1 & m+1 \leq j \leq m+n, (x,t) \in [0, L_j) \times [0, T) \\ \lambda_j(x,T) = 0 & x \in [0, L_j] \\ \lambda_j(L_j, t) = 0 & t \in [0, T]. \end{cases} \tag{3.23}$$

This is the adjoint equation that will be solved to obtain λ_j. Using the boundary and initial conditions for $\delta\rho_j$ and λ_j, we can now compute the gradient of the cost function:

$$\nabla_{m_{ij}} H = -\sum_{j=m+1}^{m+n} \sum_{i=1}^{m} v_i(L_i)\rho_i(L_i, \cdot)\lambda_j(0, \cdot).$$

At each iteration, we solve the original and adjoint equations using an upwind finite difference scheme and modify the descent direction accordingly, using the gradient computed above, which gives the increment in $m_{ij}(\cdot)$. The gradient of the cost function is used as input in a nonlinear optimization method. A number of nonlinear optimization software are available, for example, KNITRO, MINOS, NPSOL and SNOPT, the first of which is used for this dissertation.

The following algorithm was implemented and converged to a minimum of the optimization program:

1. Solve the partial differential equations for the density on each link.

2. Solve the adjoint equations.

3. Evaluate the gradient of the cost functional.

4. Use this result in a nonlinear optimization method.

5. Return to step 1 until numerical convergence.

Algorithm 1: Continuous adjoint method.

The method is shown for completeness. However, it is joint work, and the results are not shown in this thesis. This optimization method was implemented on the network represented in Figure 3.9; the links are taken from the high altitude en route jetways between Salt Lake City and Oakland International Airport. We use jetways J56, J58-80, J84, J148, J156, J158, J198 and J199. The input is constructed using ASDI/ETMS data. We note that without control, the flows are often above the desirable threshold, whereas we manage to maintain the flow under the limit at all times by applying the optimal control strategy. The method used here consists in finding an optimal routing through the coefficients of the allocation matrix that will prevent sudden jumps in aircraft density. These coefficients are automatically adjusted in order to allow the best repartition of aircraft on the network. If a given link is becoming congested, the allocation coefficient that regulates the inflow on this link will decrease and the other coefficients at this junction will correspondingly increase, thus redirecting the aircraft to less congested links. Thus, we are able to maintain a regular spacing between the aircraft even if sudden increases in aircraft density are registered at the entrance of the network. In the absence of control, these jumps in aircraft density are not mitigated and eventually allow the aircraft flow to exceed the limit. Full details of the results are shown in [73]. Note that following the publication of [73], the problem (3.22) was shown to be a convex problem when discretized with a specific linear discretization scheme [77]. It is a major improvement with respect to this formulation, which still leaves computational issues unanswered. These issues are still the focus of ongoing research [77].

3.4 Comparison of the four models' performance

When we compare the predictive capabilities of the four models, it can be seen that the four models differ in accuracy. From the validation performed in Section 3.2 (see Figure 3.8 in particular), we can see that the PDE model displays the best prediction capabilities among all the models. As can be seen in Figure 3.8, the sector count prediction of the PDE model is closer to the recorded ASDI/ETMS data, compared with the other three. In comparing the four models, we will quantify each model's error as well as its computational efficiency. The computational cost of optimally solving problems with the four models presented here is out of the scope of this thesis. The next chapter of the thesis will present the specific computational issues of optimal control of the CTM(L).

3.4.1 Error analysis

Two comparisons are performed, similar to in the validation process of CTM(L) in Section 2.3.2: cumulated occurrence of sector count error breach (S) and the instantaneous sector count error, where S is defined as the summation of time intervals under the condition that difference of sector counts between the simulation and ASDI/ETMS data is greater than or equal to a user-specified capacity limitation within a certain time window. This is summarized in equation (2.21) in Section 2.3.2.

The time window we choose in the simulation is 1440 minutes (24 hours), i.e. $T = 1440$. Different values of C_s are used, and plots of the percentage of breaches versus C_s are shown in Figure 3.10. For example, if we choose $C_s = 3$, the percentage of breaches of the MMM in sector ZOA32 is 15%, which means the predicted sector counts in ZOA32 by the MMM differ from the ASDI/ETMS data by at least three aircraft for 15% of the time. As the value of C_s increases, the breach length for each model tends to zero. This is because C_s is the aircraft count error limit. The PDE model is close to zero breach when the aircraft count error limit is less than five, which has the best predictive performance. The aircraft count error limits that bring the CTM(L) and the MMM to zero breach are higher than the PDE model, while the MM2D requires the largest aircraft count error to bring zero breach.

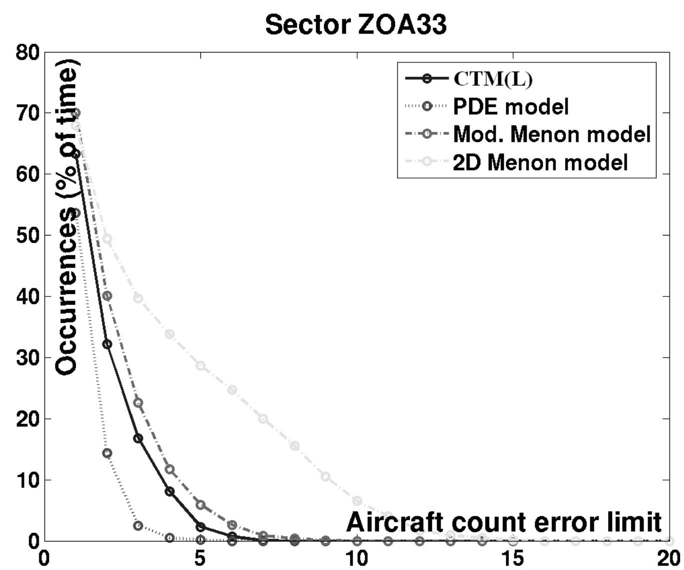

Figure 3.10: Cumulative distribution of breach of sector count error for high load sector ZOA33 (unit is $\%$ of the time).

The instantaneous sector count error analysis is performed as well. This error is the difference between the models' aircraft count and the actual aircraft count for each sector, computed from the recorded ASDI/ETMS data at each time step in the simulation. The corresponding relative error is the ratio between the absolute instantaneous error and the actual count. Statistics of the absolute instantaneous error and the relative instantaneous error for sector ZOA33 as a function of time for a day are presented in Table 3.2.

Number	1	2	3	4	5
Name	ZOA13	ZOA14	ZOA15	ZOA31	ZOA32
Number	6	7	8	9	10
Name	ZOA33	ZOA34	ZOA35	ZOA36	ZOA43

Table 3.1: Indices for a portion of the considered sectors in Oakland Center (numbers refer to Figure 3.11).

Absolute error (aircraft)	Mean	Max	Variance
MMM	1.5456	9	2.4695
MM2D	3.5241	15.8750	12.4788
CTM(L)	1.2373	7	1.7758
PDE	0.7119	5	0.6328
Relative error	Mean	Max	Variance
MMM	0.2706	2.5000	0.0933
MM2D	0.5204	2.3105	0.1579
CTM(L)	0.2000	3	0.0669
PDE	0.1160	2	0.0288

Table 3.2: Instantaneous error (absolute error and relative error) statistics for high load sector ZOA33 on January 1st, 2005.

Figure 3.11 shows a summary of the max/mean error of the sector counts, and the error variance as well. From Figure 3.11, we can see that the PDE model exhibits less error and less variance than the other three.

The MM2D model has the largest predictive errors among the four for two major reasons: (i) The fineness of MM2D depends on the SEL size, a $1^\circ \times 1^\circ$ latitude-longitude tessellation in [50], which is coarse compared with other models. With smaller SEL size, MM2D has more states which increases the computational complexity of the model. (ii) The parameters of MM2D (a and β) are assumed to be constant, which usually differ from the actual parameters in the time of interest in the real system.

3.4.2 Computational efficiency

The respective performance of the models are compared (forward simulation). This enables us to assess their computational tractability. For models based on the network graph (the CTM(L), the MMM and the PDE model), it takes approximately 45 minutes to convert the aggregate traffic flow graph model referred to in Section 3.1.5 according to each model's specifications [2], while the MM2D model needs approximately three days to identify the system parameters (a and β) using a full year of ASDI/ETMS data. Table 3.3 lists the CPU time and memory usage for the four models to predict sector counts. The analysis is done for 75 high altitude sectors in Figure 3.4. The computations are performed on a 1.6 GHz CPU, 2 GB RAM PC running Linux, using the C++ programming language. The CTM(L) has the fastest running time (20 minutes), which is about 10 times faster than the PDE model and 15 times faster than the MMM. The running time of the MM2D is relatively faster than the PDE model and the MMM. The difference between the CTM(L)

[2] Constructing the graph model alone needs four days, using a full year of ASDI/ETMS data.

and the PDE model is that the time increments required for a PDE model simulation are smaller than the delay unit used in the CTM(L). The reason why the MMM has the largest running time is because the MMM must keep track of all the merge/diverge nodes in the system, for which a number of matrix multiplications are needed for all merge and diverge nodes at each time step. For the PDE model and the CTM(L), the aircraft count updates are based only on the previous counts and the path length (see Section 3.1.5). Since the MM2D is based on a different modeling structure, i.e., by partitioning the airspace into small blocks (see Section 3.1.4; in this study, a $1^\circ \times 1^\circ$ latitude-longitude tessellation is applied), the number of states of the MM2D is smaller than those of the PDE model and the MMM, but larger than that of the CTM(L). This is why the MM2D has comparable computational efficiency to the CTM(L).

Models	CTM(L)	PDE	MMM	MM2D
CPU time (minutes)	20	224	310	45
Max RAM usage (MB)	759	1262	697	732

Table 3.3: Computational efficiency (runs performed on a 1.6 GHz CPU, 2 GB RAM PC running Linux, using C++).

3.5 Conclusion

Four Eulerian models were implemented and compared in this thesis. We started with the Large-capacity Cell Transmission Model, and then presented a modified version of the Menon model adapted to fit a general network topology. We also presented a new application of the Lax Wendroff scheme to a partial differential equation representing air traffic flow. Finally, we implemented the two-dimensional Menon model. The models were applied to high altitude traffic for six

Air Route Traffic Control Centers in the National Airspace System. Each model was used for simulation over an entire day. Compared to flight data, the models show accurate predictive capabilities. The models were also compared in terms of their computational time and memory requirements. Control strategies were designed and implemented on similar benchmark scenarios for two of the models that were also compared.

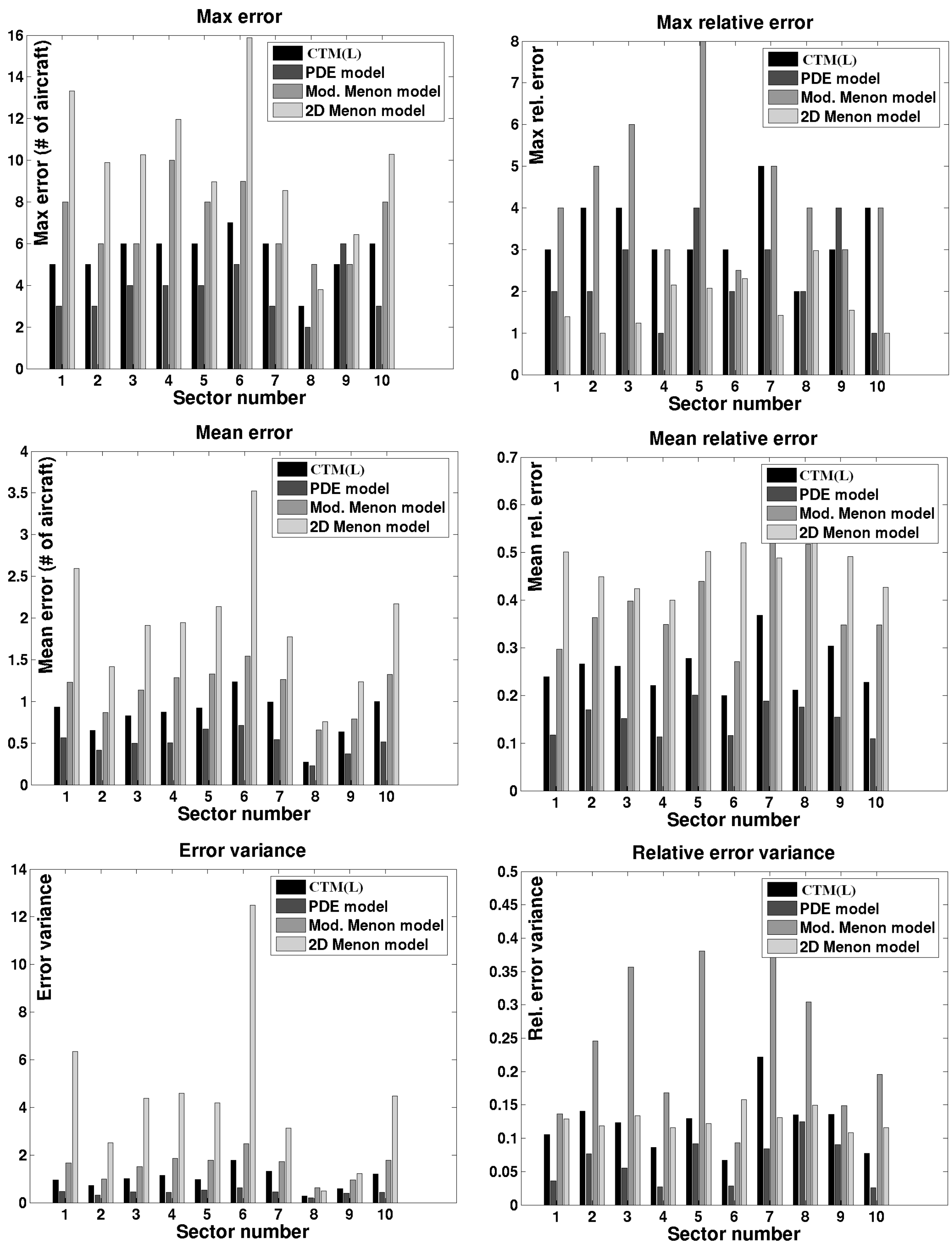

Figure 3.11: **Left:** Summary of the absolute instantaneous error of aircraft sector count. **Right:** The relative error summary. Numbers on the y-axis correspond to the sectors listed in Table 3.1.

Chapter 4

Optimization-based Traffic Flow Management

In this chapter, a dual decomposition method is developed to solve the problem of minimizing the total travel time of flights in the National Airspace System of the United States. The method uses the flow model developed in Chapter 2 of this thesis. Given flight departures and destinations, we can solve the problem of finding optimal en route delay control actions which follow sector capacity constraints while minimizing the total flight time for all the flights in the NAS. The problem is formulated as an *integer program* (IP) with billions of variables and constraints, which is relaxed to a *linear program* (LP) for computational tractability. Solving the global LP directly is prohibitively complex due to the high dimensions of the state space variables and the constraints. A method based on dual decomposition is developed to solve the *large scale linear program* (LSLP) [10] resulting from this problem formulation.

The method of dual decomposition has been used since the 1960's with the historical work of Dantzig and Wolfe [24]. A good, modern reference of dual decomposition is Chapter 6 in the book of Bertsekas [10]. The dual decomposition method has been applied in engineering, such as in rate control for communication networks [38], and to networking problems for simultaneous routing and resource allocation [78]. Recently, the dual decomposition method was presented in survey [19] in an effort towards a systematic understanding of "layering" as "optimization decomposition," where the overall communication network is modeled by a generalized network utility maximization problem, and each layer corresponds to a decomposed subproblem. Alternative decomposition methods were also applied in network utility maximization problems to obtain different distributed algorithms [74; 57]. The dual decomposition method was also recently used to solve large computationally intractable problems for formation flight with multiple cooperative agents, which resulted in an algorithm that is easily implementable in a decentralized manner [60].

In this chapter, it will be shown that the dual decomposition method is particularly well suited to the network structure of the aggregate traffic flow model, CTM(L), which is described in Chapter 2. It breaks the LSLP into a sequence of small LP problems (subproblems), which are much more tractable and can be solved very efficiently in real-time. It consists of an iterative algorithm, in which the subproblems are solved involving their own local variables as well as the variables of subproblems they are coupled with.

4.1 Customization of CTM(L) for dual decomposition

The CTM(L) model is introduced in Chapter 2. To simplify the notations for the optimization problem in this chapter, the dynamics of CTM(L) are revisited in this section. The notational changes are required to make the dual decomposition method clear.

The behavior of air traffic flow on a single link can be modeled by a deterministic linear dynamical system with a unit time delay, following Chapter 2:

$$x_i(k+1) = A_i x_i(k) + B_1^i u_i(k) + B_2^i f_i(k), \tag{4.1}$$

$$y(k) = \tilde{C}_i x_i(k), \tag{4.2}$$

where $x_i(k) = [x_i^{m_i}(k), \cdots, x_i^1(k)]^T$ is the state vector whose elements represent the corresponding aircraft counts in each cell of link i at time step k, and m_i is the number of cells in the link. The forcing input, $f_i(k)$, is a scalar that denotes the entry count onto link i during a unit time interval from k to $k+1$, and the control input, $u_i(k)$, is an $m_i \times 1$ vector, representing delay control. The output, $y(k)$, is the aircraft count in a user-specified set of cells at time step k. The nonzero elements of the $m_i \times 1$ vector $\tilde{C}_i$ correspond to the cells in the user-specified set, and are equal to one. A_i is an $m_i \times m_i$ nilpotent matrix with 1's on its super-diagonal. $B_2^i = [0, \cdots, 0, 1]^T$ is the forcing vector with m_i elements, and B_1^i is the $m_i \times m_i$ holding pattern matrix, in which all nonzero elements are 1 on the diagonal and -1 on the super-diagonal. [1]

Based on the link level model, it is straightforward to build a sector level model using the same technique. Suppose there are n links in a sector, then the dynamics for the sector level model can be described as:

$$x(k+1) = Ax(k) + B_1 u(k) + B_2 f(k), \tag{4.3}$$

$$y(k) = \tilde{C} x(k), \tag{4.4}$$

where $x(k) = [x_n(k), \cdots, x_1(k)]^T$ denotes the state, and $f(k) = [f_n(k), \cdots, f_1(k)]^T$ is the forcing input vector (the entry count onto the sector). The control input vector is $u(k) = [u_n(k), \cdots, u_1(k)]^T$.

[1] B_1^i and B_2^i are defined as B_u^i and B_f^i in Chapter 2, respectively.

The vector $y(k)$ represents the aircraft count in a user-specified set of cells at time step k. The matrices A, B_1 and B_2 are block diagonal, with block elements associated with each link in the sector. For example, $A = \text{diag}(A_n, \cdots, A_1)$ with A_i's defined by equation (4.1). In the above model, the matrices A, B_1 and B_2 are sparse and highly structured, which can be exploited to develop efficient algorithms for optimization using this model [2].

When building a NAS-wide model (at the ARTCC level) flights are first clustered based on their origin-destination (source-sink) pairs in the network. Each pair corresponds to a *path* consisting of links between these nodes. If two or more paths have one link in common, this link will be duplicated. Therefore, the NAS-wide model can also be cast in the same framework of (4.3-4.4) and the corresponding $x(k)$ includes all cells of the complete network. The forcing input, $f(k)$, is now the entry count into the NAS. The output, $y(k)$, denotes the aircraft count in a user-specified set of cells at time step k.

4.2 Problem formulation

4.2.1 Notations, nomenclature

The following notations are used to formulate the optimization problem presented in the rest of the chapter.

- $S = \{s_1, s_2, \cdots, s_{|S|}\}$: set of sectors in the NAS. $|S|$ denotes total number of sectors in the NAS.
- $V = \{v_1, v_2, \cdots, v_{|V|}\}$: set of vertices, defined in Section 2.1.2. $|V|$ denotes total number of vertices in the NAS model constructed earlier.

[2] B_1 and B_2 are defined as B^u and B^f in Chapter 2, respectively.

- $E = \{e_1, e_2, \cdots, e_{|E|}\}$: set of links (Section 2.1.2). Link $e_m = (v_i, v_j)$, or simply $e_m = (i, j)$, corresponds to an ordered pair of vertices.

- $K = \left\{k_{o_1,d_1}, k_{o_2,d_2}, \cdots, k_{o_{|K|},d_{|K|}}\right\}$, or simply $K = \{k_1, k_2, \cdots, k_{|K|}\}$: set of origin-destination (OD) pairs: origin (source) o_k, destination (sink) d_k. $|K|$ denotes total number of OD pairs. In this chapter, OD pairs can also be understood as "paths," because given an OD pair, a path is uniquely defined: no multiple paths exist between a single OD pair.[3]

- T: time horizon of the optimization.

- $Q_s \subset E$: set of cells in sector $s \in S$. $(j, k) \in Q_s$ means the j-th cell on path k is in Q_s (therefore in sector s).

- $x(i, j, t)$ or $x_t^{j,i}$: state of cells in the dynamical system. $x_t^{j,i}$ represents the number of aircraft in the j-th cell on path i (namely cell (i, j); defined in Section 2.1.2) at time t. The vector x_t^k or $x^k(t)$ is used to denote the aggregation of states on path k at time t: $x_t^k = [x(k, 1, t), x(k, 2, t), \cdots, x(k, n(k), t)]^T$, where $n(k)$ is the total number of cells on path k. The vector x^k is the aggregation of states on path k: $x^k = [x_1^k; x_2^k; \cdots; x_T^k]$. The vector $x(t)$ or x_t is the aggregation of states at time t: $x_t = \left[x_t^1; x_t^2; \cdots; x_t^k; \cdots; x_t^{|K|}\right]$. x represents the vector of all the states: $x = [x(1); x(2); \cdots; x(T)]$.

- $u(i, j, t)$ or $u_t^{j,i}$: delay control in cell (i, j) at time t, representing number of delay controlled aircraft in the j-th cell on path i at time t. The vector u_t^k or $u^k(t)$ is used to denote the aggregation of controls on path k at time t: $u_t^k = [u(k, 1, t), u(k, 2, t), \cdots, u(k, n(k), t)]^T$, where $n(k)$ is total number of cells on path k. The vector u^k is the aggregation of the controls

[3]This feature can easily adapt to cases of multiple paths between an OD pair. It is omitted in this chapter for simplicity of the description.

on path k: $u^k = \left[u_1^k; u_2^k; \cdots; u_T^k\right]$. The vector $u(t)$ or u_t is the aggregation of controls at time t: $u_t = \left[u_t^1; u_t^2; \cdots; u_t^k; \cdots; u_t^{|K|}\right]$. u represents the vector of all the controls: $u = [u(1); u(2); \cdots; u(T)]$.

- $c(i,j)$, $i = 1, \cdots, |K|$, $j = 1, \cdots, n_i$ (n_i is the number of cells on path i), means the cost associated with flying through cell (i,j), which is the travel time of a flight through cell (i,j). $c(i,j) = 1$ represents one minute travel time in the present CTM(L). Let c^k represent an aggregation of costs on path k: $c^k = [c(k,1), c(k,2), \cdots, c(k,n(k))]^T$, and let $c = \left[c^1; c^2; \cdots; c^{|K|}\right]$.

- $C_s(t)$, $s = 1, \cdots, |S|$: sector capacity for sector s at time t. The sector capacities are time dependent because the usage of airspace is dynamic: capacity can change due to weather or operations. Denote $C(t)$ = $\left[C_1(t), C_2(t), \cdots, C_{|S|}(t)\right]^T$, was the aggregation of capacity constraints at time t, and $C = [C(1); C(2); \cdots; C(T)]$ as the aggregation of all sector capacities at all times.

- Slack variables $Z_s^k(t)$, $s = 1, \cdots, |S|$, $k = 1, \cdots, |K|$, represent the number of aircraft in sector s on path k at time t: $Z_s^k(t) = \sum_{(j,k)\in Q_s} x(k,j,t)$, where (j,k) is the j-th cell on path k and Q_s is the set of links in sector s. Let $Z^k(t)$ denote the aggregation of slack variables $Z_s^k(t)$ on path k at time t: $Z^k(t) = \left[Z_1^k(t), Z_2^k(t), \cdots, Z_{|S|}^k(t)\right]^T$, and $Z^k = [Z^k(1); Z^k(2); \cdots; Z^k(T)]$ are slack variables associated with path k. Let $Z(t)$ denote the aggregation of all slack variables at time t: $Z(t) = \left[Z^1(t); Z^2(t); \cdots; Z^{|K|}(t)\right]$. Let $Z = [Z(1); Z(2); \cdots; Z(T)]$ be the aggregation of all slack variables.

- $\mathbb{T}_0 = \{0, \cdots, T-1\}$: set of time indices from 0 to $T-1$.

- $\mathbb{T}_1 = \{1, \cdots, T\}$: set of time indices from 1 to T.
- $\mathbb{T} = \{0, \cdots, T\}$: set of time indices from 0 to T.
- $\mathbb{S} = \{1, \cdots, |S|\}$: set of sector indices.
- $\mathbb{K} = \{1, \cdots, |K|\}$: set of path indices.
- $\mathbb{Z}_+$: the non-negative integer set.

4.2.2 Formulation

As was explained in the previous chapter of the thesis, the problem of minimizing the total travel time of the flights in the NAS can be formulated as follows:

$$\min_{x,u} \quad \sum_{t=0}^{T} c^T x_t \tag{4.5}$$

$$\textbf{s.t.} \quad x_0 = B_2 f_0 \tag{4.6}$$

$$x_{t+1} = Ax_t + B_1 u_t + B_2 f_t, \qquad t \in \mathbb{T}_0 \tag{4.7}$$

$$\sum_{(i,j)\in Q_s} x_t^{i,j} \leq C_s(t), \qquad s \in \mathbb{S}, t \in \mathbb{T} \tag{4.8}$$

$$u \leq x \tag{4.9}$$

$$x \subset \mathbb{Z}_+ \tag{4.10}$$

$$u \subset \mathbb{Z}_+. \tag{4.11}$$

The objective function (4.5) encodes the minimization of the total travel time for all the flights in the NAS for the time horizon of interest. The constraint (4.6) represents the initial condition, i.e., the airborne flights at the beginning of optimization. The constraint (4.7) encodes the dynamics of the system. The constraint (4.8) enforces the capacity constraint for every sector, meaning that the number of aircraft in the sector cannot exceed the sector capacity. The constraint (4.9) is the control constraint for every cell: the number of delay controlled aircraft cannot exceed the total number of aircraft in the cell. Constraints (4.10) and (4.11) represent the non-negativity integer constraint on the states and controls, respectively.

4.2.3 Comments on this optimization formulation

Using the framework above, different objective functions can be used for other optimization purposes. In particular, as long as the objective function is convex, there exist efficient algorithms to solve the optimization problem [17; 10]. Moreover, when the terms in the objective function are seperable path by path, the dual decomposition method described in the rest of this chapter can be applied following the algorithm described in Section 4.3. This is one of the main contributions of the dissertation, which makes this large scale optimization computationally tractable.

4.3 Dual decomposition

In the model, there are approximately 100,000 paths, and every path usually has hundreds of cells. In order to perform two-hour TFM ($t = 1, \cdots, 120$), the problem consists of about five billion states and controls ($x(i,j,t)$ and $u(i,j,t)$); the number of constraints is of the same order (billions). The formulation in Section 4.2.2 is an *integer program* (IP), which is computationally challenging to solve efficiently. It is very unlikely that at the time this dissertation is written, any computational software could solve a MILP of this size in reasonable time on a reasonable computational platform (maybe with the exception of computer clusters in National Labs). The cost of the relaxation of this MILP itself is also enormous, and solving this relaxation on a regular computer is most likely an impossible task with today's computing capabilities. This chapter solves this problem using the algorithmic approach of dual decomposition. To make the problem computationally tractable, we relax the last two constraints (4.10) and (4.11) to $u \geq 0$, and as a consequence, $x \geq u \geq 0$, by constraint (4.9). The formulation is now a *linear program* (LP):

$$\min_{x,u} \quad \sum_{t=0}^{T} c^T x_t \tag{4.12}$$

$$\text{s.t.} \quad x_0 = B_2 f_0 \tag{4.13}$$

$$x_{t+1} = Ax_t + B_1 u_t + B_2 f_t, \qquad t \in \mathbb{T}_0 \tag{4.14}$$

$$\sum_{(i,j)\in Q_s} x_t^{i,j} \leq C_s(t), \qquad s \in \mathbb{S}, t \in \mathbb{T} \tag{4.15}$$

$$0 \leq u_t \leq x_t, \qquad t \in \mathbb{T}. \tag{4.16}$$

However, LP relaxation does not change the size of the problem: we still have the same number of variables and constraints as in the IP formulation. Now, we apply the dual decomposition method [10] to solve the large scale LP.

Step 1 Decompose the terms path by path. The objective function can be rewritten as a summation of the total travel time of flights along each path, where the path index is denoted by k. Each constraint can also be written path by path, which is also indexed by k for the k-th path.

$$
\begin{array}{lll}
\min\limits_{x,u} & \sum\limits_{k=1}^{|K|}\left(\sum\limits_{t=0}^{T} c^{k^T} x_t^k\right) & \\
\text{s.t.} & x_0^k = B_2^k f_0^k, & k \in \mathbb{K} \\
& x_{t+1}^k = A^k x_t^k + B_1^k u_t^k + B_2^k f_t^k, & t \in \mathbb{T}_0, k \in \mathbb{K} \\
& 0 \leq u_t^k \leq x_t^k, & t \in \mathbb{T}, k \in \mathbb{K} \\
& \sum\limits_{k=1}^{|K|} \sum\limits_{(i,k)\in Q_s} x_t^{i,k} \leq C_s(t), & s \in \mathbb{S}, t \in \mathbb{T}.
\end{array}
$$

$\boxed{\textbf{Step 2}}$ Introduce slack variables $Z_s^k(t)$, $Z(t)$ and Z, as defined in Section 4.2.1.

$$
\begin{array}{lll}
\min\limits_{x,u,Z} & \sum\limits_{k=1}^{|K|}\left(\sum\limits_{t=0}^{T} c^{k^T} x_t^k\right) & \\
\text{s.t.} & x_0^k = B_2^k f_0^k, & k \in \mathbb{K} \\
& x_{t+1}^k = A^k x_t^k + B_1^k u_t^k + B_2^k f_t^k, & t \in \mathbb{T}_0, k \in \mathbb{K} \\
& 0 \leq u_t^k \leq x_t^k, & t \in \mathbb{T}, k \in \mathbb{K} \\
& \sum\limits_{(i,k)\in Q_s} x_t^{i,k} = Z_s^k(t), & s \in \mathbb{S}, k \in \mathbb{K}, t \in \mathbb{T} \\
& \sum\limits_{k=1}^{|K|} Z_s^k(t) \leq C_s(t), & s \in \mathbb{S}, t \in \mathbb{T}.
\end{array}
$$

$\boxed{\textbf{Step 3}}$ Form the partial Lagrangian for the last constraints,

$$
\sum_{k=1}^{|K|} Z_s^k(t) \leq C_s(t), \quad s \in \mathbb{S}, t \in \mathbb{T},
$$

and express the problem in an equivalent partial-Lagrangian form as:

$$p^* := \min_{x,u,Z} \max_{\lambda \geq 0} \quad \sum_{k=1}^{|K|} \left(\sum_{t=0}^{T} {c^k}^T x_t^k \right) + \sum_{t=0}^{T} \sum_{s=1}^{|S|} \lambda_s(t) \left(\sum_{k=1}^{|K|} Z_s^k(t) - C_s(t) \right) \tag{4.17}$$

$$\begin{aligned}
\textbf{s.t.} \quad & x_0^k = B_2^k f_0^k, && k \in \mathbb{K} \\
& x_{t+1}^k = A^k x_t^k + B_1^k u_t^k + B_2^k f_t^k, && t \in \mathbb{T}_0, k \in \mathbb{K} \\
& 0 \leq u_t^k \leq x_t^k, && t \in \mathbb{T}, k \in \mathbb{K} \\
& \sum_{(i,k) \in Q_s} x_t^{i,k} = Z_s^k(t), && s \in \mathbb{S}, k \in \mathbb{K}, t \in \mathbb{T}.
\end{aligned}$$

where p^* denotes the primal optimal value of the problem. The $\lambda_s(t)$, $s = 1, \cdots, |S|, t = 0, \cdots, T$, are Lagrange multipliers. The aggregation of all Lagrange multipliers (over space and time) is denoted by $\lambda = \{\lambda_s(t) \,|\, s = 1, \cdots, |S|,\ t = 0, \cdots, T\}$.

Step 4 Switch the min and max operators, and obtain the dual problem:

$$d^* := \max_{\lambda \geq 0} \min_{x,u,Z} \quad \sum_{k=1}^{|K|} \left(\sum_{t=0}^{T} {c^k}^T x_t^k \right) + \sum_{t=0}^{T} \sum_{s=1}^{|S|} \lambda_s(t) \left(\sum_{k=1}^{|K|} Z_s^k(t) - C_s(t) \right) \tag{4.18}$$

$$\begin{aligned}
\textbf{s.t.} \quad & x_0^k = B_2^k f_0^k, && k \in \mathbb{K} \\
& x_{t+1}^k = A^k x_t^k + B_1^k u_t^k + B_2^k f_t^k, && t \in \mathbb{T}_0, k \in \mathbb{K} \\
& 0 \leq u_t^k \leq x_t^k, && t \in \mathbb{T}, k \in \mathbb{K} \\
& \sum_{(i,k) \in Q_s} x_t^{i,k} = Z_s^k(t), && s \in \mathbb{S}, k \in \mathbb{K}, t \in \mathbb{T}.
\end{aligned}$$

where d^* denotes the optimal value of the dual problem. Notice that the primal and dual functions are both linear. We assume that Slater's condition for constraint qualifications [17; 10] is satisfied, i.e., there exists a feasible solution x, u and Z such that the capacity constraints hold with strict inequality:

$$\sum_{k=1}^{|K|} Z_s^k(t) < C_s(t), \quad s \in \mathbb{S}, t \in \mathbb{T}.$$

This is always true in practice: when the initial airborne aircraft strictly satisfy the capacity constaints, i.e., $\sum_{k=1}^{|K|} Z_s^k(0) < C_s(t)$ for all $s = 1, \cdots, |S|$, $t = 0, \cdots, T$, we can hold all of them in their sectors by delay controls, and apply "do-not-fly" to all other departure flights by holding them on the ground. While this is not a feasible "physical" solution, it provides the mathematical requirements of the Slater's condition. With this assumption, the optimal values of the dual problem (4.18) and the primal problem (4.17) are equal [17; 10]. This allows us to solve the primal (4.17) via the dual (4.18).

$\boxed{\textbf{Step 5}}$ Re-arrange the terms in the objective function of the dual problem (4.18) to group the terms path by path:

$$\begin{aligned}
&\max_{\lambda \geq 0} \min_{x,u,Z} \sum_{k=1}^{|K|} \left(\sum_{t=0}^{T} c^{k^T} x_t^k \right) + \sum_{t=0}^{T} \sum_{s=1}^{|S|} \lambda_s(t) \left(\sum_{k=1}^{|K|} Z_s^k(t) - C_s(t) \right) \\
= \; &\max_{\lambda \geq 0} \min_{x,u,Z} \sum_{k=1}^{|K|} \left(\sum_{t=0}^{T} c^{k^T} x_t^k \right) + \sum_{t=0}^{T} \sum_{s=1}^{|S|} \sum_{k=1}^{|K|} \lambda_s(t) Z_s^k(t) - \sum_{t=0}^{T} \sum_{s=1}^{|S|} \lambda_s(t) C_s(t) \\
= \; &\max_{\lambda \geq 0} \left\{ - \sum_{t=0}^{T} \sum_{s=1}^{|S|} \lambda_s(t) C_s(t) + \min_{x,u,Z} \sum_{k=1}^{|K|} \left(\sum_{t=0}^{T} c^{k^T} x_t^k \right) + \sum_{k=1}^{|K|} \sum_{t=0}^{T} \sum_{s=1}^{|S|} \lambda_s(t) Z_s^k(t) \right\} \\
= \; &\max_{\lambda \geq 0} \left\{ - \sum_{t=0}^{T} \sum_{s=1}^{|S|} \lambda_s(t) C_s(t) + \min_{x,u,Z} \sum_{k=1}^{|K|} \left(\sum_{t=0}^{T} c^{k^T} x_t^k + \sum_{t=0}^{T} \sum_{s=1}^{|S|} \lambda_s(t) Z_s^k(t) \right) \right\} \\
= \; &\max_{\lambda \geq 0} \left\{ - \sum_{t=0}^{T} \sum_{s=1}^{|S|} \lambda_s(t) C_s(t) + \sum_{k=1}^{|K|} \left(\min_{x,u,Z^k} \sum_{t=0}^{T} c^{k^T} x_t^k + \sum_{t=0}^{T} \sum_{s=1}^{|S|} \lambda_s(t) Z_s^k(t) \right) \right\} \\
= \; &\max_{\lambda \geq 0} \left\{ - \sum_{t=0}^{T} \sum_{s=1}^{|S|} \lambda_s(t) C_s(t) + \sum_{k=1}^{|K|} d^{k^*}(\lambda) \right\},
\end{aligned}$$

where

$$d^{k^*}(\lambda) = \min_{x,u,Z^k} \sum_{t=0}^{T} {c^k}^T x_t^k + \sum_{t=0}^{T} \sum_{s=1}^{|S|} \lambda_s(t) Z_s^k(t),$$

which is actually the subproblem for path k.

$\boxed{\textbf{Step 6}}$ Iteration procedure. At each iteration, the master problem provides updated Lagrange multipliers λ for the subproblems, while the subproblems compute the optimal controls u and states x for the master problem.

- Subproblems $d^{k^*}(\lambda)$, $k \in \mathbb{K}$

$$\begin{aligned}
\min_{x,u,Z^k} \quad & \sum_{t=0}^{T} {c^k}^T x_t^k + \sum_{t=0}^{T} \sum_{s=1}^{|S|} \lambda_s(t) Z_s^k(t) && (4.19)\\
\text{s.t.} \quad & x_0^k = B_2^k f_0^k \\
& x_{t+1}^k = A^k x_t^k + B_1^k u_t^k + B_2^k f_t^k, && t \in \mathbb{T}_0\\
& 0 \le u_t^k \le x_t^k, && t \in \mathbb{T}\\
& \sum_{(i,k)\in Q_s} x_t^{i,k} = Z_s^k(t), && s \in \mathbb{S}, t \in \mathbb{T}.
\end{aligned}$$

  There are $|K|$ (number of paths) subproblems.

- Master problem

$$d^*(\lambda) = \max_{\lambda \ge 0} \left\{ -\sum_{t=0}^{T} \sum_{s=1}^{|S|} \lambda_s(t) C_s(t) + \sum_{k=1}^{|K|} d^{k^*}(\lambda) \right\}. \tag{4.20}$$

To solve the dual problem (4.18), or (4.20), we need to compute the subgradient of $d^*(\lambda)$. In the subgradient method used in this study, we start with initial $\lambda = \lambda_0 > 0$. At each iteration step $i = 1, 2, 3, \cdots$, we compute a subgradient of the dual function (4.18):

$$g(t) = -\left(\sum_{k=1}^{|K|} Z^k(t) - C(t) \right), \quad t = 0, \cdots, T.$$

Then we update the dual variable (Lagrange multiplier, in this formulation) by

$$\lambda(t) := (\lambda(t) - \alpha_i g(t))_+, \quad t = 0, \cdots, T,$$

where $(\cdot)_+$ denotes the non-negative part of a vector (i.e., projection onto the non-negative orthant), and α_i is the subgradient step size rule, which is any nonsummable positive sequence that converges to zero [17]:

$$\alpha_i \to 0, \quad \sum_{i=1}^{\infty} \alpha_i = \infty.$$

In this study, we use the diminishing step size rule $\alpha_i = 1/\sqrt{i+1}$.

Proof of convergence of the above algorithm can be found in Chapter 2 of [68]. Numerous methods to accelerate the convergence can be found in literature [68; 10].

The dual decomposition method, with the subgradient method for the master problem outlined above, gives Algorithm 2.

Once the algorithm converges, one obtains the x_t^k, u_t^k, etc. These can be assembled into control policies at NAS-wide level, i.e., the number of delay minutes u_t^k, assigned to flights on path k at time t. The resulting traffic is given by x_t^k, the number of aircraft on path k at time t.

4.4 Results

The dual decomposition method was implemented in C++, with subproblems (4.19) solved using ILOG CPLEX Concert [3]. A one-hour TFM problem was solved for the whole continental NAS in the United States.

Figures 4.1–4.6 show the delay control situation of the sectors using the dual decomposition method at different times. Sectors with a larger controlled count are colored with a darker

Inputs:

Initial state x_0.

Time horizon T.

Inputs f_t, $t \in \mathbb{T}$.

Required sector capacity constraints $C_s(t)$, $s \in \mathbb{S}$, $t \in \mathbb{T}$.

Initial $\lambda := \lambda_0 \geq 0$.

Start: Iteration number $i = 0$.

repeat

$i := i + 1$.

Solve the subproblems (4.19) for $d^{k^*}(\lambda)$, obtain $x_t^k, u_t^k, Z^k(t)$, $k \in \mathbb{K}$, $t \in \mathbb{T}$.

Master algorithm subgradients $g(t) = -\left(\sum_{k=1}^{|K|} Z^k(t) - C(t)\right)$, $t \in \mathbb{T}$.

Master algorithm update $\lambda(t) := (\lambda(t) - \alpha_i g(t))_+$, $t \in \mathbb{T}$.

until d^* converges or i = max iterations.

Output: x_t^k, u_t^k, $t \in \mathbb{T}$, $k \in \mathbb{K}$.

Algorithm 2: Dual decomposition algorithm.

color.

Figures 4.7–4.9 show a comparison between sector counts for three sectors. For each of them, the counts are displayed for an uncontrolled scenario (when the controls u in system (4.3) are set to be zero) and a controlled scenario (when the controls u are optimized by the dual decomposition method). The sector capacities are also represented in the figures as a reference. As can be seen from the solution, by generating an optimal delay allocation in the NAS, the dual decomposition algorithm minimizes the total travel time of the flights in the entire airspace while respecting sector capacity constraints. For example in Figure 4.8, the capacity of sector ZOB26 (a high altitude sector in Cleveland ARTCC) is 18; when no delay control is applied, the number of flights in ZOB26 is above 18 after 52 minutes and can reach 22 (at 55 and 56 minutes), which exceeds the sector capacity (18 flights at a time), while the number of flights stays below the sector capacity at all times when delay control is applied. Figure 4.9 shows the situation for sector ZOB29, a neighbor sector of ZOB26 in Cleveland ARTCC, whose capacity is 17. When no delay control is applied, the number of flights in ZOB29 is always below the capacity (under-utilized). However, with an optimal delay control, the dual decomposition algorithm allocates delays in ZOB29, increasing its sector load, which reaches the sector capacity at 35-37 minutes. This is exactly how dual decomposition helps: reducing sector loads by allocating delays in under-utilized sectors (usually in under-utilized neighbors).

The computation is done on a Dell Server PE1900 with a 2.33GHz Intel(R) Xeon(R) CPU E5345, 8GB RAM, running Microsoft Windows Server 2003 R2, Standard x64 Edition with Service Pack 2. The computing time for a one-hour TFM problem is about 90 minutes. With a fixed network model of the NAS, the computing time does not significantly change with different scenar-

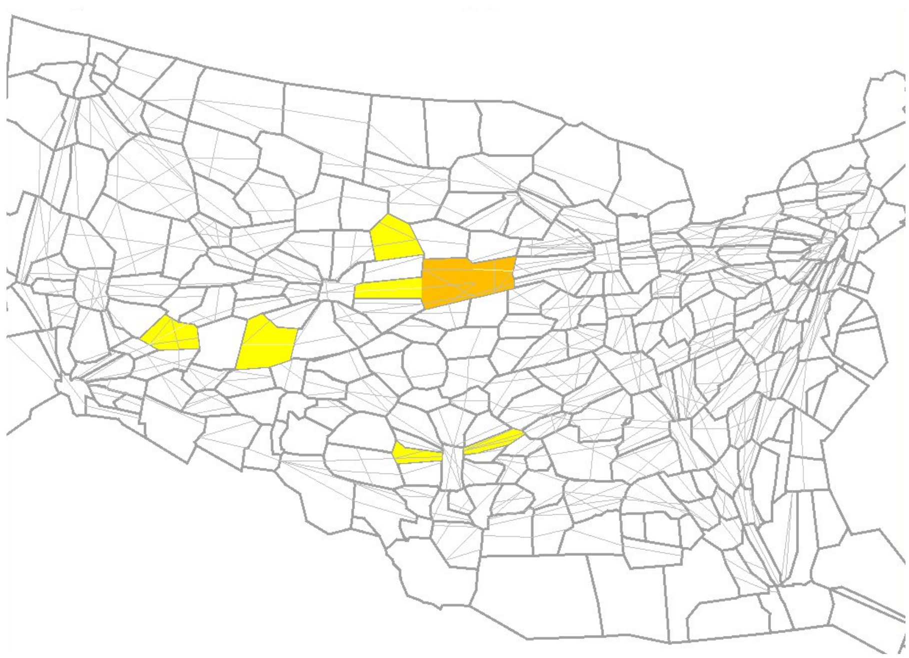

Figure 4.1: Solution nine minutes after the start time, with delay control computed by the dual decomposition algorithm. Shading of sectors indicates aircraft occupancy in each sector with control.

ios (amount of inputs, level of sector capacities, etc.). Using the fact that the dual decomposition method is highly suitable for parallel computing, the computing time can be greatly reduced when parallel computing facilities are available, which makes real-time NAS-wide TFM possible.

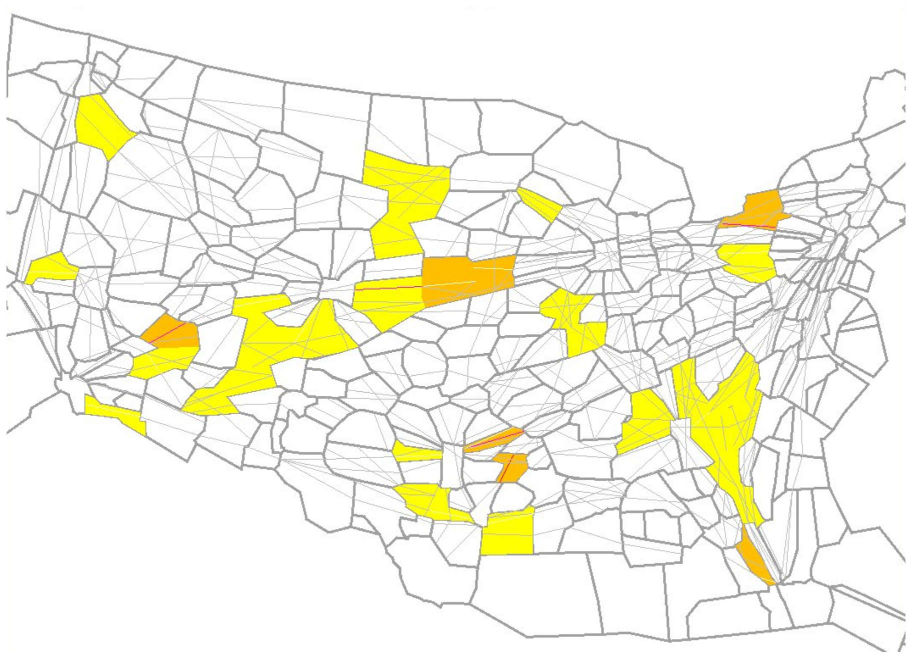

Figure 4.2: Solution 19 minutes after the start time, with delay control computed by the dual decomposition algorithm. Shading of sectors indicates aircraft occupancy in each sector with control.

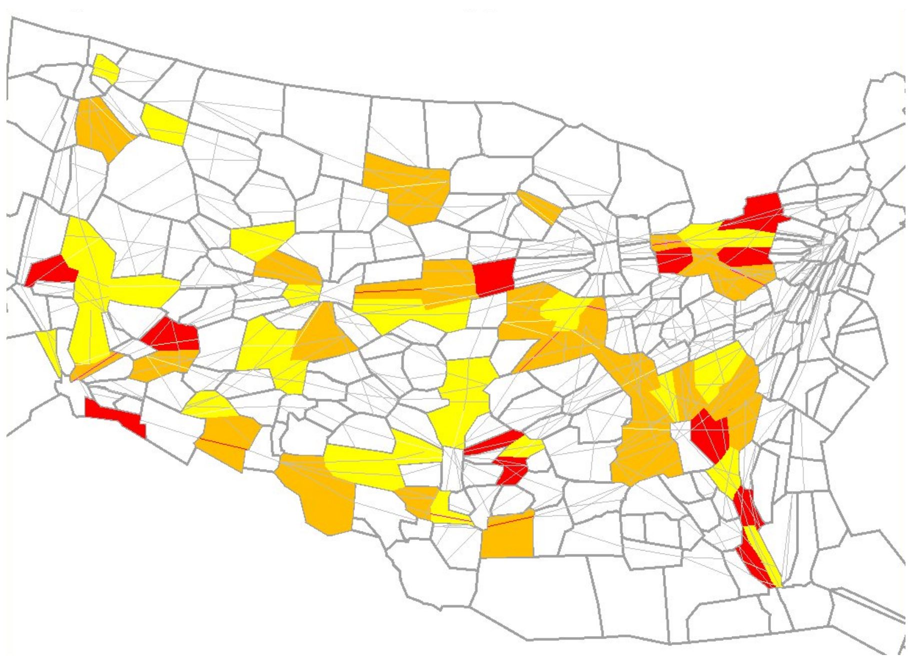

Figure 4.3: Solution 29 minutes after the start time, with delay control computed by the dual decomposition algorithm. Shading of sectors indicates aircraft occupancy in each sector with control.

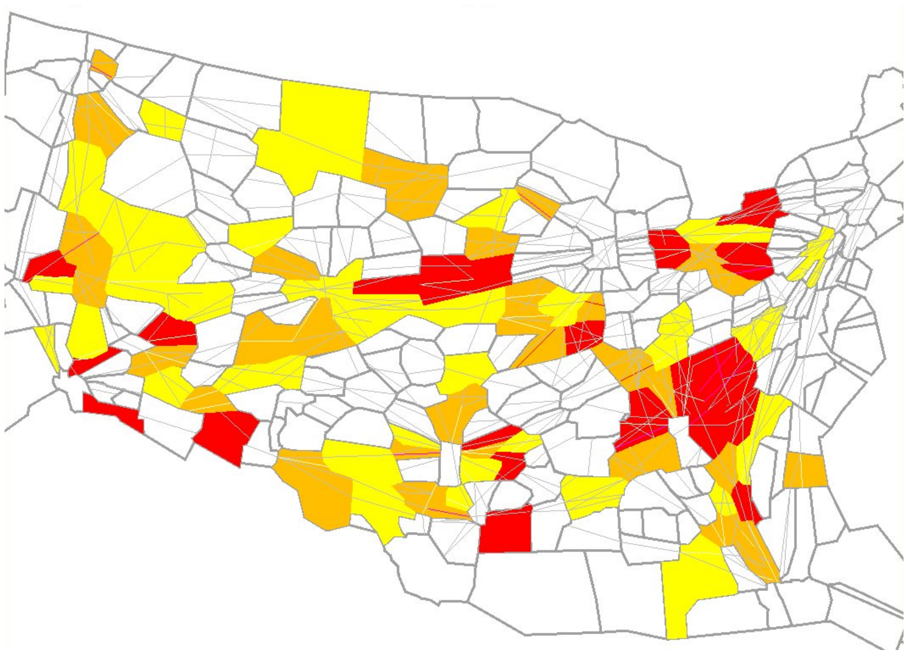

Figure 4.4: Solution 39 minutes after the start time, with delay control computed by the dual decomposition algorithm. Shading of sectors indicates aircraft occupancy in each sector with control.

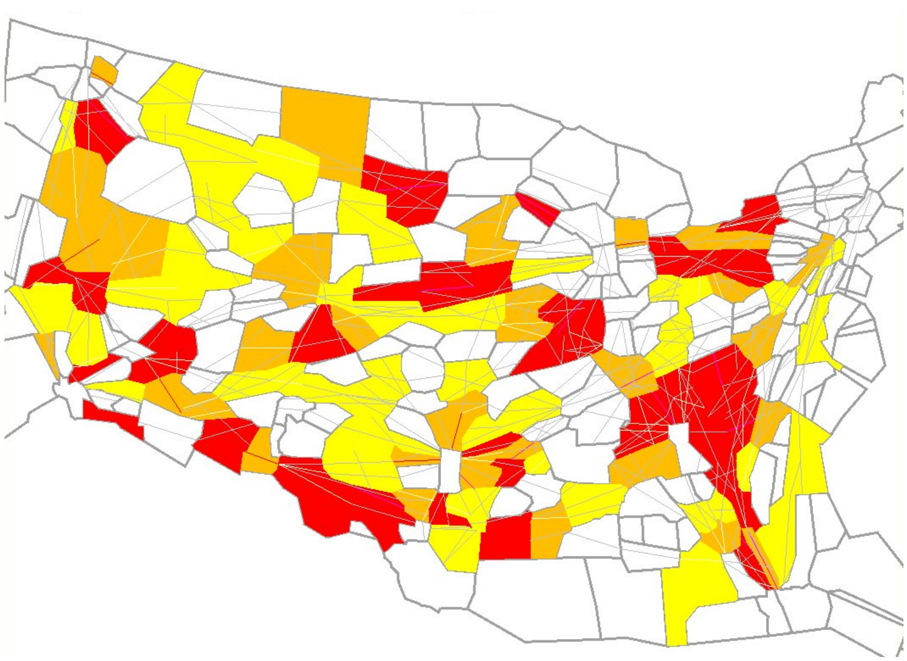

Figure 4.5: Solution 49 minutes after the start time, with delay control computed by the dual decomposition algorithm. Shading of sectors indicates aircraft occupancy in each sector with control.

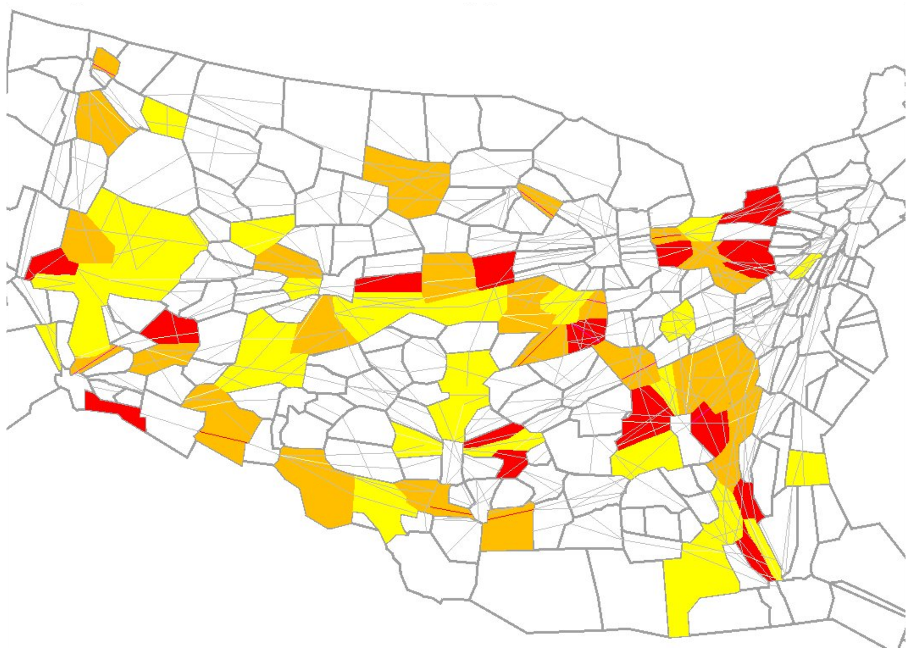

Figure 4.6: Solution 59 minutes after the start time, with delay control computed by the dual decomposition algorithm. Shading of sectors indicates aircraft occupancy in each sector with control.

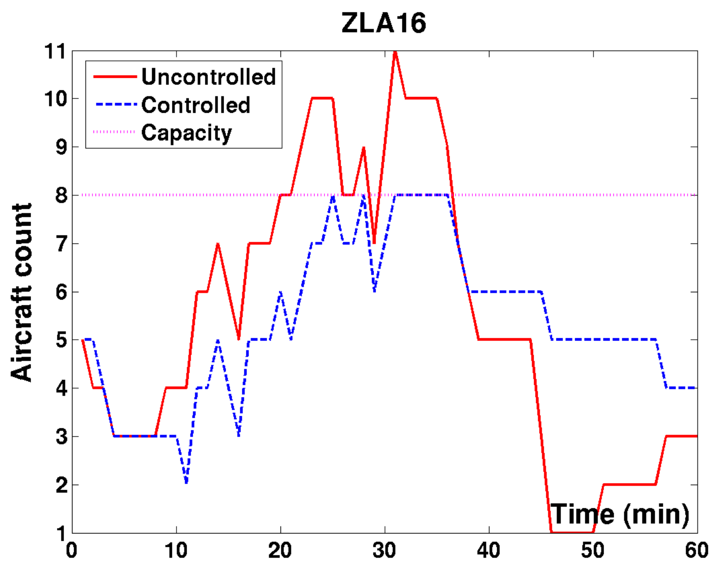

Figure 4.7: Comparison of controlled and uncontrolled sector counts in Sector ZLA16. The sector capacity is set to be eight.

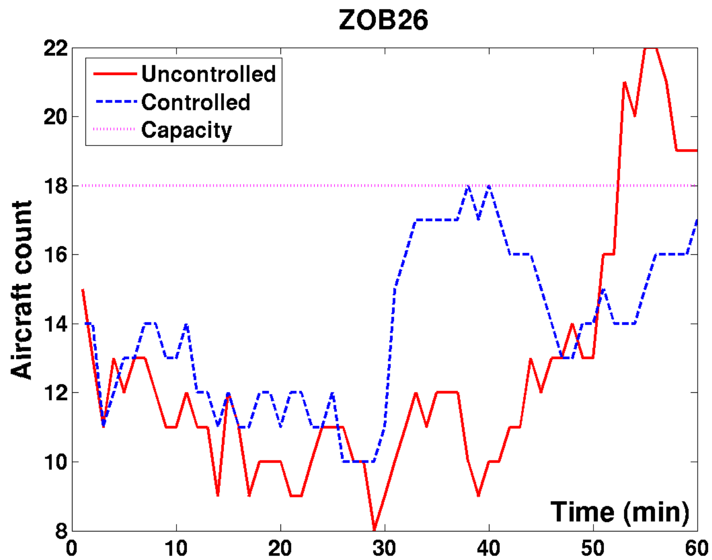

Figure 4.8: Comparison of controlled and uncontrolled sector counts in Sector ZOB26. The sector capacity is set to be 18.

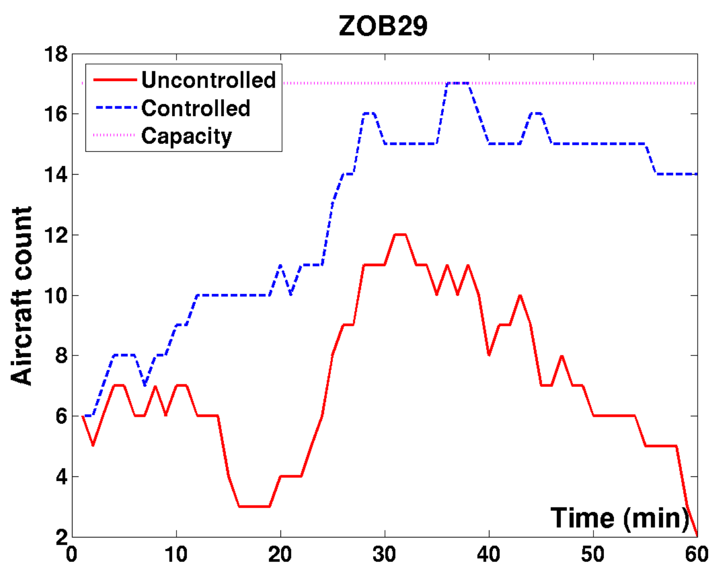

Figure 4.9: Comparison of controlled and uncontrolled sector counts in Sector ZOB29. The sector capacity is set to be 17.

Chapter 5

Summary, future work

In this brief and final chapter, we summarize the work presented in this dissertation and the systems engineering contributions.

5.1 Contributions

We presented a new Eulerian-Lagrangian model, the CTM(L), of en route air traffic flow. The CTM(L) is based on a network graph model generated using a full year of historical air traffic data. A flow model is plugged on the network to describe the evolution of traffic along its links. The model is compared with three other aggregate air traffic models, in which the predictive capability and computational efficiency of each model are compared and analyzed. Optimal control problems are formulated for two of the air traffic models. A solution procedure is subsequently developed for the continuous model, relying on adjoint based optimization. A dual decomposition algorithm is designed and implemented for the CTM(L) model. It enables a computationally tractable solution of the optimal control problem. It makes the real-time NAS-wide Traffic Flow Management problem possible to solve despite its billions of variables and constraints formulation.

The main contributions of this dissertation are briefly outlined below.

- A numerical parameter identification method is developed to automatically generate a multicommodity network model for a user-defined airspace, using ETMS/ASDI data.
- A flow-based Eulerian-Lagrangian NAS model is constructed on top of a graph-theoretic multicommodity network model incorporating the topology of the NAS and the resulting flow patterns. Therefore, the model is physically meaningful.
- The model is reduced to a linear time invariant dynamical system, in which the transition matrix is nilpotent. This feature greatly facilitates the design (optimization) and analysis of the model.
- The model is successfully validated against ASDI/ETMS air traffic data for a whole year and for the whole NAS, i.e. 20 continental ARTCCs.
- For the first time, four aggregate Air Traffic Control models have been implemented and compared in regards to their predictive capability and computational efficiency on the same benchmark scenarios.
- A framework is designed for the formulation of large scale optimization problems that is useful for NAS-wide TFM. It relies on *mixed integer linear program* (MILP).
- A computationally tractable optimization algorithm based on a dual decomposition method is designed for solving the MILP formulation of the optimal control problem. This makes a NAS-wide Traffic Flow Management problem with billions of variables and constraints solvable in real-time.

- The new NAS-wide model and the optimization algorithm are integrated in FACET, a software developed at NASA Ames Research Center, in collaboration with Metron Aviation.

- A NAS-wide TFM problem is solved numerically, which include approximately 6,000 aircraft over the course of one to four hours.

Novel and interesting features of these results include the following items:

1. Since the model takes into account the OD information, it does not have split parameters and eliminates the diffusion problem, a difficulty commonly faced in transportation engineering problems.

2. Since the model is control volume based (Eulerian) and has a very classical discretized linear dynamical representation, it is computationally less expensive than discretization of partial differential equations or other nonlinear models.

3. The model is scalable. The granularity of the model is dependent on the time step (one minute in this study), which can be changed to different time scales and represents models at different levels, e.g., from sector level to center level of the NAS.

5.2 Future work, broader impacts of this work

It is expected that this work will impact TFM research in the following areas.

- **Incorporating airports in the CTM(L).** Current CTM(L) model focuses on en route air traffic. Airports are not part of the model because of several reasons, one of which is a lack of air traffic data for flights close to airports at the time when this work started. It would be straightforward to generate additional links connecting airports to high altitude sectors, expanding the current network model of the NAS. This will rely on either additional air traffic data (close to airports) or other data processing procedures such as interpolation techniques, and is the focus of ongoing work.

- **Modeling the NAS with seasonal patterns.** The CTM(L) model developed in this thesis is a time invariant system. In the future, the uncertainty in en route demand and the volatility associated with seasonal patterns will be studied, which will make the model time varying, and further more, stochastic. A time varying/stochastic model will be able to take into account of additional factors that have an impact on the NAS (e.g., weather or special used airspace and operations), and sophisticated optimization algorithms (robust optimization, stochastic optimization) will be employed for TFM problems in this generalized framework.

- **Assessing en route delays.** Since the CTM(L) is a path (route) based model, and the dual decomposition algorithm uses the underlying network structure of the CTM(L), it will be natural to use the modeling and optimization algorithms developed in this thesis for en route delay assessment. The optimal delay control generated by the dual decomposition algorithm can be directly interpreted by holding procedures when required. Because it complies with approach procedures and controller instructions, it can be used to assess en route delays.

- **Automated bottleneck identification in the NAS.** Bottlenecks in the NAS are usually understood as sections (e.g. sectors, airports) with carrying capacities substantially below those that characterize other sections of the same route. When sensitivity analysis [17] is applied, the formulation in Chapter 4 for the TFM problem can be used to identify the bottlenecks in the NAS, which is related to the Lagrange multipliers in the problem (4.17) of Chapter 4. More efforts can be allocated to increase the capacities of identified bottlenecks while minimizing the total effort that can be used to increase the capability of the entire NAS.

- **Airspace Flow Programs.** The FAA introduced a new capability in the spring of 2006 known as the *Airspace Flow Program* (AFP). The AFP combines the power of *Ground Delay Programs* (GDPs) and *Flow Constrained Areas* (FCAs) to allow more efficient, effective, equitable and predictable management of airborne traffic in congested airspace [2]. Using the framework of CTM(L), a model including airports will be built. The impact of weather will be part of the model. [1] Powered by the model, traffic patterns will be studied and the problem areas will be identified to create FCAs. Optimization algorithms will be developed, which incorporate the dynamics of the model, FCAs in constraints. The outputs of the optimization algorithms will be optimal air traffic control procedures including optimal departure times and optimal en route delays. Compared to current approaches, the AFP will address unnecessary delays while providing better control of demand, more equity and more flexibility for customers.

[1] A weather model may be developed and combined with the traffic flow model based on CTM(L).

Bibliography

[1] http://www.faa.gov/.

[2] http://www.metronaviation.com.

[3] http://www.ilog.com/corporate/training/acrobat/CPLEX.pdf.

[4] R. K. Ahuja, T. L. Magnati, and J. B. Orlin. *Network flows: theory, algorithms, and application*. Prentice Hall, Upper Saddle River, NJ, 1993.

[5] E. Alpaydin. *Introduction to Machine Learning*. The MIT Press, Cambridge, MA, 2004.

[6] Anonymous. Terminal area forecast summary, fiscal year 2004-2020. Technical Report FAA-APO-05-1, U.S. Department of Transportation, Federal Aviation Administration, March 2005.

[7] H. Balakrishnan. Techniques for reallocating airport resources during adverse weather. In *Proceedings of the* 46^{th} *IEEE Conference on Decision and Control*, pages 2949–2956, New Orleans, LA, December 2007.

[8] G. Bastin and V. Guffen. Congestion control in compartmental network systems. *Systems and Control Letters*, 55(8):689–696, 2006.

[9] A. M. Bayen, R. L. Raffard, and C. J. Tomlin. Adjoint-based control of a new Eulerian network model of air traffic flow. *IEEE Transactions on Control Systems Technology*, 14(5):804–818, 2006.

[10] D.P. Bertsekas. *Nonlinear programming*. Athena Scientific, Belmont, MA, 2nd edition, 2002.

[11] D. Bertsimas and S. Stock Patterson. The air traffic flow management problem with enroute capacities. *Operations Research*, 46(3):406–422, May–June 1998.

[12] D. Bertsimas and J. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, Belmont, MA, February 1997.

[13] K. Bilimoria and H. Q. Lee. Aircraft conflict resolution with an arrival time constraint. In *AIAA Conference on Guidance, Navigation, and Control Conference and Exhibit*, Monterey, CA, August 2002. AIAA Paper 2002–4444.

[14] K. Bilimoria, B. Sridhar, G. Chatterji, K. Sheth, and S. Grabbe. FACET: Future ATM concepts evaluation tool. In *Proceedings of the* 3^{rd} *USA/Europe ATM 2001 R&D Seminar*, Naples, Italy, June 2001.

[15] P.A. Bonnefoy and R.J. Hansman. Scalability and evolutionary dynamics of air transportation networks in the United States. In *AIAA Aviation Technology, Integration, and Operations Conference*, pages 785–795, Belfast, Northern Ireland, Ireland, September 2007.

[16] F. Borelli. *Constrained Optimal Control of Linear and Hybrid Systems*, volume 290 of *Lecture Notes in Control and Information Sciences*. Springer Verlag, New York, NY, 2003.

[17] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, New York, NY, 2004.

[18] B. Chandran and H. Balakrishnan. A dynamic programming algorithm for robust runway scheduling. In *Proceedings of the American Control Conference*, pages 1161–1166, New York, NY, July 2007.

[19] M. Chiang, S.H. Low, A.R. Calderbank, and J.C. Doyle. Layering as optimization decomposition: a mathematical theory of network architectures. *Proceedings of the IEEE*, 95(1):255–312, January 2007.

[20] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. Prentice Hall, Upper Saddle River, NJ, 2nd edition, 2002.

[21] J. M. Coron, B. d'Andréa-Novel, and G. Bastin. A strict Lyapunov function for boundary control of hyperbolic systems of conservation laws. *IEEE Transactions on Automatic Control*, 52(1):2–11, 2007.

[22] C. Daganzo. The cell transmission model: a dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research*, 28B(4):269–287, 1994.

[23] C. Daganzo. The cell transmission model, part II: network traffic. *Transportation Research*, 29B(2):79–93, 1995.

[24] G.B. Dantzig and P. Wolfe. Decomposition principle for linear programs. *Operations Research*, 8:101–111, 1960.

[25] S. Devasia, M. Heymann, and G. Meyer. Automation procedures for air traffic management:

A token-based approach. In *Proceedings of the American Control Conference*, pages 736–741, Anchorage, AK, May 2002.

[26] C. Dever, B. Mettler, E. Feron, J. Popovic, and M. McConley. Nonlinear trajectory generation for autonomous vehicles via parameterized maneuver classes. *AIAA Journal of Guidance, Control and Dynamics*, 29(2):289–302, 2006.

[27] H. Erzberger, T.J. Davis, and S. Green. Design of center-TRACON automation system. In *AGARD Guidance and Control Symposium on Machine Intelligence in Air Traffic Management*, pages 11.1–11.12, 1993.

[28] M. Garavello and B. Piccoli. *Traffic Flow on Networks*, volume 1. American Institute of Mathematical Sciences on Applied Math, Springfield, MO, 2006.

[29] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. W. H. Freeman and Company, New York, NY, 1979.

[30] M. Gugat, M. Herty, A. Klar, and G. Leugering. Optimal control for traffic flow networks. *Journal of Optimization Theory and Applications*, 126(3):589–616, 2005.

[31] M. Hansen, A. Mukherjee, D. Knorr, and D. Howell. Effect of T-TMA on capacity and delay at los angeles international airport. *Transportation Research Record No. 1788*, pages 43–48, 2002.

[32] B. Haut and G. Bastin. A second order model of road junctions in fluid models of traffic networks. *Networks and Heterogeneous Media*, 2(2):227–253, 2007.

[33] M. Heymann, G. Meyer, and S. Resmerita. An agent based framework for control of merg-

ing air traffic. In *Proceedings* 16^{th} *IFAC Symposium on Automatic Control in Aerospace*, St. Petersburg, Russia, June 2004.

[34] B. Hoffman, J. Krozel, S. Penny, A. Roy, and K. Roth. A cluster analysis to classify days in the national airspace system. In *AIAA Conference on Guidance, Navigation, and Control Conference and Exhibit*, Austin, TX, August 2003. AIAA Paper 2003–5711.

[35] S. R. Hu, S. Madanat, J. Krogmeier, and S. Peeta. Estimation of dynamic assignment matrices and OD demands using adaptive Kalman filtering. *Intelligent Transportation Systems*, 6:281–300, 2001.

[36] D. Iamratanakul, G. Meyer, G. Chatterji, and S. Devasia. Quantification of airspace sector capacity using decentralized conflict. In *Proceedings of the* 43^{rd} *IEEE Conference on Decision and Control*, pages 2003–2009, Atlantis, Paradise Island, Bahamas, December 2004.

[37] A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3:233–260, 1988.

[38] F. Kelly, A. Maulloo, and D. Tan. Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49:237–252, 1997.

[39] C. Kirchner, M. Herty, S. Göttlich, and A. Klar. Optimal control for continuous supply network models. *Networks and Heterogeneous Media*, 1(4):675–688, 2006.

[40] P. Kopardekar and S. Green. Airspace restriction planner for sector congestion management. In *AIAA 5th ATIO and 16th Lighter-Than-Air Sys Tech. and Balloon Systems Conferences*, Arlington, Virginia, September 2005. AIAA Paper 2005–7435.

[41] O. Koroleva, M. Krstic, and G. Schmid-Schonbein. Decentralized and adaptive control of nonlinear fluid flow networks. *International Journal of Control*, 16:801–818, 2006.

[42] Y. Kuwata, A. Richards, T. Schouwenaars, and J. How. Distributed robust receding horizon control for multivehicle guidance. *IEEE Transactions on Control Systems Technology*, 15(4):627–641, July 2007.

[43] S. Larsson and V. Thomee. *Partial differential equations with numerical methods*. Springer, Berlin, NY, 2003.

[44] R. J. LeVeque. *Finite Volume Methods for Hyperbolic Problems,*. Cambridge University Press, Cambridge, U.K., 2002.

[45] M. J. Lighthill and G. B. Whitham. On kinematic waves. II. A theory of traffic flow on long crowded roads. *Proceedings of the Royal Society of London*, 229(1178):317–345, 1956.

[46] X. Litrico and V. Fromion. Frequency modeling of open-channel flow. *Journal of Hydraulic Engineering*, 130(8):806–815, 2004.

[47] P. O. Malaterre. PILOTE: Linear quadratic optimal controller for irrigation canals. *Journal of Irrigation and Drainage Engineering*, 124(4):187–194, 1998.

[48] Z. H. Mao, E. Feron, and K. Bilimoria. Stability and performance of intersecting aircraft flows under decentralized conflict avoidance rules. *IEEE Transactions on Intelligent Transportation Systems*, 2(2):101–109, June 2001.

[49] P. K. Menon, G. D. Sweriduk, and K. Bilimoria. New approach for modeling, analysis and

control of air traffic flow. *AIAA Journal of Guidance, Control and Dynamics*, 27(5):737–744, 2004.

[50] P. K. Menon, G. D. Sweriduk, T. Lam, G. M. Diaz, and K. Bilimoria. Computer-aided Eulerian air traffic flow modeling and predictive control. *AIAA Journal of Guidance, Control and Dynamics*, 29(1):12–19, 2006.

[51] B. Miller and J-P Clarke. The hidden value of air transportation infrastructure. *Technological Forecasting and Social Change*, 74:18–35, 2007.

[52] K. T. Mueller, D. R. Schleicher, and K. Bilimoria. Conflict detection and resolution with traffic flow constraints. In *AIAA Conference on Guidance, Navigation, and Control Conference and Exhibit*, Monterey, CA, August 2002. AIAA Paper 2002–4445.

[53] A. Mukherjee and M. Hansen. A dynamic stochastic model for the single airport ground holding problem. *Transportation Science*, 41(4):444–456, November 2007.

[54] N. S. Nise. *Control Systems Engineering*. John Wiley, Hoboken, NJ, 2004.

[55] M. S. Nolan. *Fundamentals of Air Traffic Control*. Brooks Cole, Reading, MA, 4th edition, 2003.

[56] R. Paielli and H. Erzberger. Improved conflict detection for reducing operational errors in air traffic control. In *AIAA 4th Aviation Technology, Integration and Operations (ATIO) Forum*, Chicago, IL, September 2004. AIAA Paper 2004–6392.

[57] D.P. Palomar and M. Chiang. Alternative distributed algorithms for network utility maximiza-

tion: framework and applications. *IEEE Transactions on Automatic Control*, 52(12):2254–2269, December 2007.

[58] M. Papageorgiou. Dynamic modeling, assignment, and route guidance in traffic networks. *Transportation Research Part B*, 24(6):471–495, 1990.

[59] J. Plaettner-Hochwarth, Y. Zhao, and J. Robinson. A modularized approach for comprehensive air traffic system simulation. In *AIAA Conference on Guidance, Navigation, and Control Conference and Exhibit*, Denver, CO, July 2000. AIAA Paper 2000-4478.

[60] R. Raffard, C. Tomlin, and S. Boyd. Distributed optimization for cooperative agents: application to formation flight. In *Proceedings of the IEEE Conference on Decision and Control*, pages 2453–2459, Nassau, Bahamas, December 2004.

[61] S. Resmerita and M. Heymann. Conflict resolution in multi-agent systems. In *Proceedings of the* 42^{nd} *IEEE Conference on Decision and Control*, pages 2537–2542, Maui, HI, December 2003.

[62] S. Resmerita, M. Heymann, and G. Meyer. A framework for conflict resolution in air traffic management. In *Proceedings of the* 42^{nd} *IEEE Conference on Decision and Control*, pages 2035–2040, Maui, HI, December 2003.

[63] P. I. Richards. Shock waves on the highway. *Operations Research*, 4(1):42–51, 1956.

[64] C. A. Robelin, D. Sun, G. Wu, and A. M. Bayen. En-route air traffic modeling and strategic flow management using mixed integer linear programming. In *INFORMS Annual Meeting 2005*, New Orleans/San Francisco, November 2006.

[65] C. A. Robelin, D. Sun, G. Wu, and A. M. Bayen. MILP control of aggregate Eulerian network airspace models. In *Proceedings of the American Control Conference*, pages 5257–5262, Minneapolis, MN, June 2006.

[66] S. Roy, B. Sridhar, and G. C. Verghese. An aggregate dynamic stochastic model for air traffic control. In *Proceedings of the* 5^{th} *USA/Europe ATM 2003 R&D Seminar*, Budapest, Hungary, June 2003.

[67] A. Saraf and G. Slater. Combined Eulerian-Lagrangian two-level control system for optimal air traffic flow management. In *AIAA Conference on Guidance, Navigation, and Control*, Keystone, CO, August 2006. AIAA Paper 2006–6229.

[68] N.Z. Shor, K.C. Kiwiel, and A. Ruszcayǹski. *Minimization methods for non-differentiable functions*. Springer-Verlag, New York, NY, 1985.

[69] S.W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing, 1997.

[70] B. Sridhar, T. Soni, K. Sheth, and G. Chatterji. An aggregate flow model for air traffic management. In *AIAA Conference on Guidance, Navigation, and Control*, Providence, RI, August 2004. AIAA Paper 2004–5316.

[71] B. Sridhar, T. Soni, K. Sheth, and G. Chatterji. Aggregate flow model for air-traffic management. *AIAA Journal of Guidance, Control and Dynamics*, 29(4):992–997, 2006.

[72] D. Sun and A.M. Bayen. Multicommodity Eulerian-Lagrangian Large-capacity Cell Transmission Model for en route traffic. *AIAA Journal of Guidance, Control and Dynamics*, 31(3):616–628, 2008.

[73] D. Sun, I. Strub, and A.M. Bayen. Comparison of the performance of four Eulerian network flow models for strategic air traffic management. *AIMS Journal on Networks and Heterogeneous Media*, 2(4):569–595, 2007.

[74] C.W. Tan, D.P. Palomar, and M. Chiang. Distributed optimization of coupled systems with applications to network utility maximization. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 5, pages 981–984, Chicago, IL, May 2006.

[75] R. Vidal. *Generalized principal component analysis (GPCA): an algebraic geometric approach to subspace clustering and motion segmentation*. Ph.D. thesis, University of California, Berkeley, CA, August 2003.

[76] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (GPCA). In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 621–628, Madison, WI, June 2003.

[77] D. Work and A.M. Bayen. Convex formulations of air traffic flow optimization problems. *Proceedings of the IEEE Special Issue on Aviation Information Systems*, accepted for publication, 2008.

[78] L. Xiao, M. Johansson, and S. Boyd. Simultaneous routing and resource allocation via dual decomposition. *IEEE Transactions on Communications*, 52(7):1136–1144, July 2004.