JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 31, No. 3, May–June 2008

Multicommodity Eulerian–Lagrangian Large-Capacity Cell Transmission Model for En Route Traffic

Dengfeng Sun* and Alexandre M. Bayen†
University of California at Berkeley, Berkeley, California 94720-1710

DOI: 10.2514/1.31717

A new paradigm for building an Eulerian–Lagrangian cell transmission model for air traffic flow is developed. It is based on an aggregation of track data and is applied to the full National Airspace System in the United States. The Eulerian–Lagrangian model is based on a multicommodity network flow model constructed from historical air traffic data. The flow model is reduced to a linear time invariant dynamical system, in which the state is a vector of aggregate aircraft counts. This model is called a large-capacity cell transmission model in reference to the cell transmission model in highway traffic, which inspired this work. The predictive capabilities of the model are successfully validated against recorded air traffic data by showing an accurate match between predicted sector counts (based on filed flight plans) and measured sector counts. The problem of controlling the sector aircraft count is posed as an integer program in which the dynamical system appears in the constraints. To improve the computational time for solving the problem, the integer program is relaxed to a linear program. The computational results show that a high portion of solutions of the linear program are integers, making this method appropriate for high-level traffic flow management.

I. Introduction

THE last few decades have witnessed the almost uninterrupted growth of U.S. air traffic, which 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. Investigations of traffic flow models have been intensively studied to mitigate the imbalance of demand and capacity of the National Airspace System (NAS). This paper presents a new model for optimization-based traffic flow management (TFM). TFM handles demand management and typically deals with traffic at the air route traffic control center (ARTCC) level, that is, 10–20 sectors. TFM problems include maintaining the aircraft count in each sector below a specified threshold to ease the human ATC workload, as well as to ensure the safety of the flights [1–8].

This task is quite cumbersome; furthermore, extensive traffic forecast simulations (including all airborne aircraft) are computationally too expensive to include systematic investigations of traffic patterns that 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 [9]. These are in contrast to Lagrangian models, which are trajectory based and take into account all aircraft trajectories [10,11]. Eulerian models have two main advantages over Lagrangian models: 1) they are computationally tractable, and their computational complexity does not depend on the number of aircraft but only on the size of the network problem; and 2) their control theoretic structure enables the use of standard methods to analyze them.

This paper presents a new model, large-capacity cell transmission model, or CTM(L), in which the two benefits outlined above are demonstrated. The terminology CTM(L) is in reference to the seminal Daganzo cell transmission model, commonly used for highway traffic [12,13]. 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, corresponding to a sector. Different from most other existing Eulerian models, the new model 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 of some Eulerian models [9,14]. For this, a multicommodity flow structure [15] has been included in the model.

Recent years have witnessed increased interest in developing new models of the NAS for the air traffic management (ATM) community. To the authors’ best knowledge, the first Eulerian model of air traffic flow was proposed by Menon et al. [9]. This model is motivated by a discretized version of the Lighthill–Whitham–Richards (LWR) partial differential equation (PDE) [16,17] and inspired by the Daganzo cell transmission model [12,13]. This Eulerian model [9] has since inspired several research groups to generate similar models including a stochastic framework, leading to results in the expected sense [18,19]. Two-dimensional models [14] have also emerged, in the hope of capturing flow patterns better. An important characteristic of these approaches [9,14,18,19] is the diffusion and dispersion that occur in these models. Although this is not a problem in a stochastic framework, because the results are in the expected sense, it is more problematic for the deterministic models [9,14], because this potentially leads to aircraft losses or inaccurate predictions (this fact has been reported in the literature [20]). A first attempt to resolve these issues was proposed as a continuous time–continuous space model in the earlier work by Bayen et al. [20], based directly on the LWR PDE. Although this approach solves the diffusion problem, its computational tractability is disputable (it depends on the required space discretization), and the resulting optimization programs require heavy computations, based on adjoint problems. A two-level control system for optimal TFM was developed recently [21] 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 the present paper, 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 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.

Received 19 April 2007; revision received 16 September 2007; accepted for publication 27 September 2007. Copyright © 2007 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved. Copies of this paper may be made for personal or internal use, on condition that the copier pay the $10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0731-5090/08 $10.00 in correspondence with the CCC.

*Ph.D. Student, Systems Engineering, Department of Civil and Environmental Engineering, Davis Hall 604; sundf@berkeley.edu. Student Member AIAA.

†Assistant Professor, Systems Engineering, Department of Civil and Environmental Engineering, Davis Hall 604; bayen@berkeley.edu. Member AIAA.

This work is related to existing frameworks, which are methods developed to solve TFM problems and which assume a flow model. Several frameworks to solve TFM problems have been proposed in the ATM community. Bertsimas and Stock [11] developed a framework of a 0–1 integer programming method for the deterministic, multi-airport air traffic flow management problem (TFMP) that addresses capacity restrictions in the en route airspace. The TFMP was shown to be nondeterministic polynomial time (NP) hard (equivalent to job-shop scheduling). This framework can be applied to many general network-based NAS models. The problem of capacity allocation between two (or more) aircraft was addressed in several papers [5,6,22,23]. The description of the state of the aircraft is made through the dynamics of individual aircraft, therefore Lagrangian. Methods for choosing conflict-free trajectories for all of the agents (aircraft) in the system were proposed for cases in which the number of aircraft in a system is known and possible trajectories of the aircraft are given [5,6,23]. In this framework, arriving aircraft are guaranteed to choose safe approach trajectories and the available airspace and arrivals runway capacities could be efficiently used. Other approaches [22] have focused on designing conflict-free trajectories using constant-speed heading-change maneuvers. The capacity of the sector was determined by acceptable output flow rate. The algorithms are decentralized and can be used for sectors with multiple intersections. A token-based ATM paradigm was also presented and successfully applied to the en route airspace [2].

The main contributions of the present paper are now summarized:

1) 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 resulting flow patterns. Therefore, the model is physically meaningful and tractable for control and optimization.

2) Because the model takes into account the OD information, it does not have split parameters (usually called β in the literature [9,13]) and eliminates the diffusion problem.

3) 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.

4) Because the model is control volume based (Eulerian) and has a very classical discretized linear dynamical representation, it is computationally less expensive.

5) The model is scalable. The granularity of the model is dependent on the time step (1 min in this study), which can be changed to different time scales and represents models at different levels, for example, from the sector level to center level of the NAS.

6) The model is successfully validated against recorded Aircraft Situation Display to Industry (ASDI), Enhanced Traffic Management System (ETMS) data for a whole year and for the whole NAS, that is, 20 continental ARTCCs. Several TFM scenarios are successfully studied based on the model.

This paper is organized as follows. The construction of a graph-theoretic multicommodity network model of traffic flow is outlined first. Air traffic flow on this graph is modeled as a discrete-time dynamical system. Then the model is validated for the full NAS using ASDI/ETMS data and the Future ATM Concepts Evaluation Tool (FACET) [10]. In particular, it is shown that one very important metric for TFM (aircraft count) is reproduced adequately by the model, which serves as a validation for the model. The problem of controlling aircraft counts in the airspace is posed as a mixed integer linear program (MILP) in which the dynamical system appears in the constraints, as is traditionally done in optimal control [24]. Numerical experiments are run to demonstrate the tractability of these methods.

II. Parameter Identification

A. Graph-Theoretic Model

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 a multicommodity network model of the observed flows. The suite of algorithms investigated includes a variety of techniques, some of them relying purely on flight tracks, others using 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. To illustrate the difficulty encountered in using canned algorithms, a series of experiments were run throughout this work. Several approaches that were investigated were summarized in the previous work [25]. 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. First, 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 to be identified. And second, 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 aforementioned algorithms from being applicable. First, the K–means algorithm requires a priori knowledge of the number of clusters (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. Second, waypoint-based classification is inappropriate because of the extremely large number of different waypoint acronyms. Even though these data are "noise free," their size is prohibitive for the present study. And third, jetway-based classification is not applicable, because ASDI/ETMS data do 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 [26]. All of these difficulties are motivation for the method presented in this paper.

1. *Definitions*

The system to be modeled is a continental en route U.S. airspace the size of 20 ARTCCs, including 284 high-altitude sectors, for altitudes 24,000 ft and above (Fig. 1). All nonmilitary flights traveling through the considered airspace are included in the scope of this work. The ASDI/ETMS data used in this study provide the position and altitude of all airborne aircraft in the United States, every minute. Additional information related to flight plans or other flight parameters, such as speed and heading, are also provided in the data, but were not used to build the present aggregate model.

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

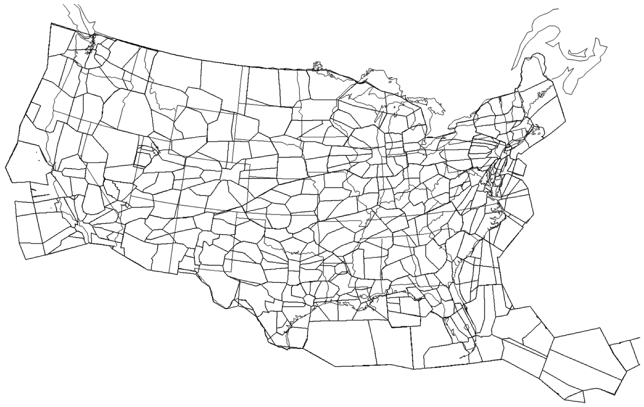

Fig. 1 Map of airspace considered in this study; 20 continental ARTCCs including 284 high-altitude sectors in the United States. Figure obtained using FACET [10].

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: 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, whereas 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 [27]. Figure 2 illustrates the concept of a link.

Interface between the considered region and the rest of the airspace (outside of the model): 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 by S 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 ft. 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 ft. 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 Fig. 1). The appellation sector for these two regions of the airspace must not be understood in the ATC sense. This appellation 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 region 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 shows a few examples of vertices and links. Note that not all vertices and links are represented in this figure.

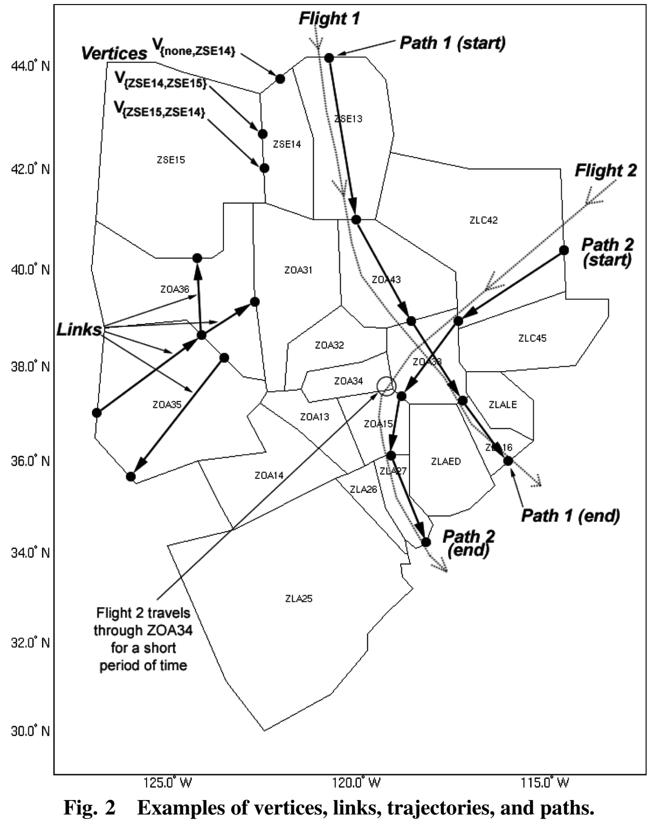

Fig. 2 Examples of vertices, links, trajectories, and paths.

Size of the network model and computational burden: Using the method described, in the multicommodity network model, there are 284 high-altitude sectors, 1598 links, and 1841 nodes. It takes 102 h 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 5 min to build the multicommodity network model. The computation was done on a 1.4 GHz CPU, 1 GB RAM PC running Linux.

3. *Classification of Trajectories and Estimation of Link Travel Times*

The aforementioned graph is used to model streams of flights traveling through the considered region of the airspace, which means that trajectories must be assigned to a set of links of the graph. The position of all airborne aircraft is provided by ASDI/ETMS data, from which sector information about all flights can be deduced.

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 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 Fig. 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, to take into account flights that stay in a given sector (s_2) for a short period of time while traveling from one sector (s_1) to another sector (s_3), in the event that the two 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, because it does not increase significantly 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 described earlier. 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 2 min. 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 Fig. 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: $V_{\{\mathrm{ZLC42,ZOA33}\}}$, $V_{\{\mathrm{ZOA33,ZOA15}\}}$, $V_{\{\mathrm{ZOA15,ZLA27}\}}$. If only the fundamental rule were applied, the sequence of vertices of flight 2 would be: $V_{\{\mathrm{ZLC42,ZOA33}\}}$, $V_{\{\mathrm{ZOA33,ZOA34}\}}$, $V_{\{\mathrm{ZOA34,ZOA15}\}}$, $V_{\{\mathrm{ZOA15,ZLA27}\}}$.

Interpolation required to decrease error on travel times estimation: Because ASDI/ETMS data provide 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 30 s, which leads to a possible error in the travel time estimation through a link as large as

1 min. Interpolation is based on the following reasonable assumptions: 1) the speed of an aircraft remains constant between the two records, which are 1 min apart; and 2) 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 intersection of the trajectory 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 of gravity of those points of boundary crossing. Note that the center of gravity is not taken in the plane but along the unfolded boundary of s_1 and s_2 [28].

B. Travel Time Analysis

For each link of the graph, the flight times for a full year (1 October 2004 to 30 September 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, that is, the aggregated travel time along the link. Figure 3 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 next. In the present setting, cells correspond to 1 min of flight time.

In the present derivation of the graph-theoretic model, the mean of the distribution of the flight time is used as the length of the link. In practice, there are several variations that should be taken into account. For example, the length (travel time) of a link changes in general, therefore, a time-varying graph model can be derived based on the time-varying link length. Figure 4 shows the mean and standard deviation of 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. For example, in the summer (June, July, and August), there is more variance in the travel time, because of the impact of the weather. 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. [29].

Depending on the objectives of the modeling, different types of distribution of the travel time can be used. For example, if the primal 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/hourly distribution will be more important.

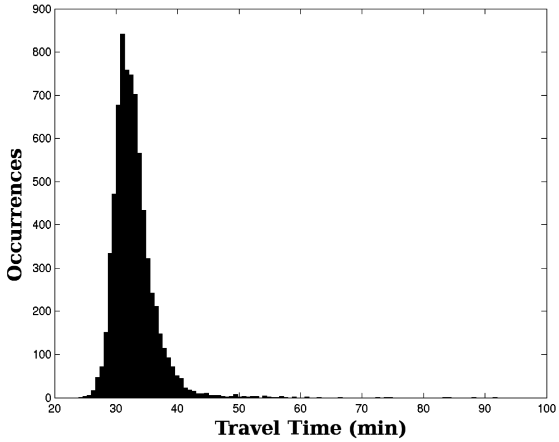

Fig. 3 Distribution of travel time on one link (ZLC45–ZOA33–ZOA34). One full year of aggregated data.

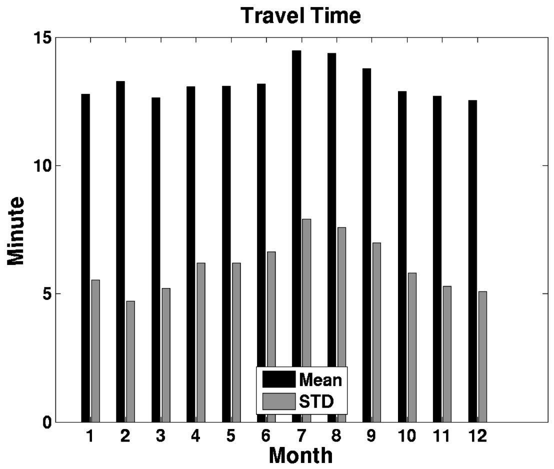

Fig. 4 Mean and standard deviation of travel time on one link (ZOA31–ZOA13–ZOA14) in different months of a year.

III. Derivation of the New Eulerian–Lagrangian Model

A. 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 inspired by the Lighthill–Whitham–Richards theory [16,17] and by the Daganzo cell transmission model [12,13] commonly used in highway traffic. The CTM(L) is based on the graph-theoretic multicommodity network model constructed from historical ASDI/ETMS traffic data, as described in Sec. II. The model is reduced to a linear time invariant dynamical system 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.

1. Link Level Model

In the CTM(L) derived next, a link is understood in the graph-theoretic sense, that is, an edge of a graph [27]. 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 exit point incident to this sector.

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

1) Each link is modeled as a directional edge. In Fig. 5, the arrow represents the flow direction. In other words, the graph is unidirectional.
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.
3) The number of cells in one link is scaled by the steps of expected travel time. In the implementation, 1 min is taken as a unit time step. For example, if it takes around 12 min 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, of course, arbitrary. In the model, a link indexed by i has m_i

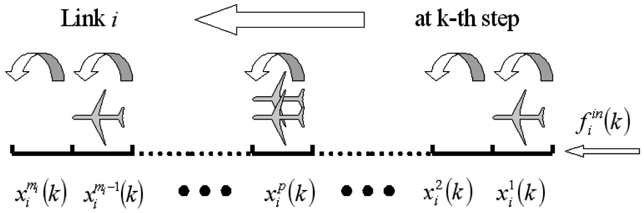

Fig. 5 Illustration of the CTM(L) at link level; everywhere inside the link, $x_i^{p+1}(k+1) = x_i^p(k)$, unless some control action was applied.

cells. The smaller the time step, the more accurate the model, but this also makes the computation more complex.

4) At the link level, only aircraft whose altitude is above 24,000 ft are taken into account for the calculation of aircraft count. This choice is also arbitrary and can be adjusted to any user-defined level.
5) 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 and VFS, as well as HP. It is supposed to be applied in 1 min time increments.
6) The model is deterministic. No statistical factor, such as weather impact, is taken into consideration at this stage. Note that it could be added later using a stochastic framework [18,19].
7) In this model, all values, including the states, inputs, and outputs, should be integers. This might increase the complexity of the computation or analysis, but provides higher accuracy.

Definitions: The following definitions are used in this work:

1) 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 $m_i \times 1$ vector, whose element $x_i^p(k)$ represents the aircraft count in cell p of link i at time k. For example, in Fig. 5, $x_i^p(k) = 2$, because there are two aircraft in the pth cell at time instant of k. m_i is the number of cells in this link as mentioned in the assumptions.
2) The forcing input, $f_i^{\text{in}}(k)$, is a scalar input, which 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 $f_i^{\text{in}}(3) = 5$.
3) 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 descent to a lower flight level. For example, in Fig. 6, $f_i^{\text{desc}}(k) = 1$.
4) The climb to en route input ("climb input" for 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 a climb from a lower flight level. Also, in Fig. 6, $f_i^{\text{climb}}(k) = 1$.
5) The control input, $u_i(k)$, is an $m_i \times 1$ vector, representing delay-based control. This type of actuation is common in the presence of congestion; the traffic controllers will typically use delay as a way of controlling air traffic flows in the en route airspace when aircraft are already airborne. The pth element denotes the number of aircraft under delay control in the pth cell of link i at the time instant of k. In this model, the cycle of increment delay is 1 min. In Fig. 7, one type of delay control, 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.
6) The output, $y_i(k)$, is the aircraft count in link i in a user-specified set of cells at time step k, for example, the total number of aircraft in all cells of this link at time step k. For example, in Fig. 5, $y_i(k) = 4$.

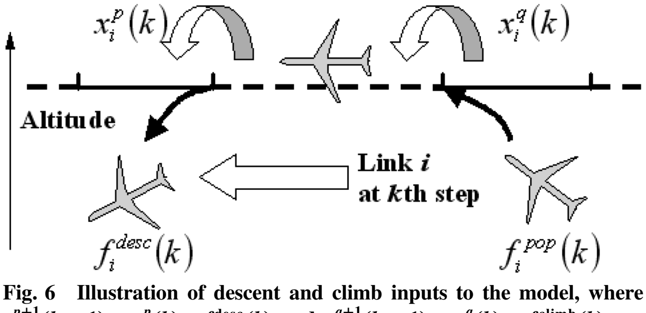

Fig. 6 Illustration of descent and climb inputs to the model, where $x_i^{p+1}(k+1) = x_i^p(k) - f_i^{\text{desc}}(k)$ and $x_i^{q+1}(k+1) = x_i^q(k) + f_i^{\text{climb}}(k)$ are satisfied unless some control action was applied.

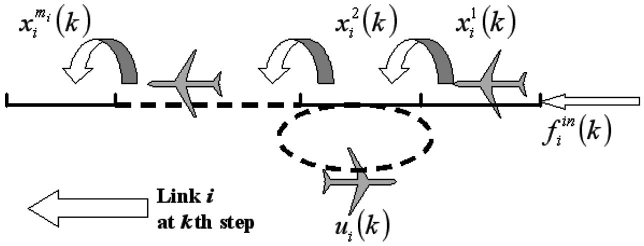

Fig. 7 Illustration of delay-based control (which could, for example, model holding pattern control, vector for spacing, or ATC prescribed deceleration) to the model, where $x_i^3 = (k+1) = x_i^2(k) - u_i(k)$ and $x_i^2(k+1) = x_i^1(k) + u_i(k)$ unless other control action was applied.

Model Description: A deterministic, linear time invariant model for link i is developed in state-space form as follows:

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

$$y_i(k) = C_i x_i(k) \tag{2}$$

where A_i is called a system matrix and is an $m_i \times m_i$ nilpotent matrix with ones on its superdiagonal. The forcing input matrix, $B_i^{\text{in}} = [0, \cdots, 0, 1]^T$, is an $m_i \times 1$ 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 ones mean that aircraft will leave from the pth cell of link i for descent or enter the qth 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 zeros except with ones on its diagonal and negative ones on its superdiagonal. 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 ones.

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, Eq. (1) can be rewritten in a more compact form:

$$x_i(k+1) = A_i x_i(k) + B_i^f f_i(k) + B_i^u u_i(k) \tag{3}$$

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:

1) The delay control always takes place at the beginning of the time step.
2) 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 mth cell of link i at the time instant of k, and the control action lasts for $p+1$ time units, then the controlled input vector will be $u_i(k) = u_i(k+1) = \cdots = u_i(k+p) = [0, \ldots, 0, n, 0, \ldots, 0]^T$, where the mth 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 control taking place at the same time, the gross controlled input vector is just the summation of each controlled input for each corresponding delay control.

2. *Sector Level Model*

It is straightforward to extend this modeling technique to set up a sector level model, 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

$$x(k+1) = Ax(k) + B^{\text{in}} f^{\text{in}}(k) + B^{\text{desc}} f^{\text{desc}}(k) + B^{\text{climb}} f^{\text{climb}}(k) + B^u u(k) \tag{4}$$

$$y(k) = Cx(k) \tag{5}$$

where $x(k) = [x_n(k)^T, \ldots, x_1(k)^T]^T$ denotes the state and $f^{\text{in}}(k) = [f_n^{\text{in}}(k)^T, \ldots, f_1^{\text{in}}(k)^T]^T$ is the forcing input vector, that is, entry count into the considered sector during a unit time interval from k to $k+1$. The descent input vector $f^{\text{desc}}(k) = [f_n^{\text{desc}}(k)^T, \ldots, f_1^{\text{desc}}(k)^T]^T$ and the climb input vector $f^{\text{climb}}(k) = [f_n^{\text{climb}}(k)^T, \ldots, f_1^{\text{climb}}(k)^T]^T$ are both column vectors with n elements. The controlled input vector, $u(k) = [u_n(k)^T, \ldots, u_1(k)^T]^T$, 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

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

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

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

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

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

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

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 Eq. (4) can be rewritten as

$$x(k+1) = Ax(k) + B^f f(k) + B^u u(k) \tag{6}$$

where

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

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

whose elements have been defined by Eq. (3).

The dimension of the state space for each sector depends on the number of total cells in the sector. For example, there are 84 cells in sector ZOA33 of the Oakland ARTCC, using 1 min of flight time as the size of a cell. Therefore, the dimension of the state space is 84 and A is an 84×84 matrix for the ZOA33 sector level model.

3. *ARTCC or Multicommodity Network Level Model*

When an ARTCC level model is created, it is necessary to include merge/diverge nodes in the network [9,18–20]. Merge nodes are straightforward; flows are added as streams of aircraft merge. Figure 8 illustrates decoupled multicommodity network models for several destination airports. These decoupled multicommodity networks are in fact trees, with air traffic flows originating from the airports in the continental United States. 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. 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 [9] inspired by the highway transportation literature [13,30]. In the present work, an alternate way of modeling the problem is proposed based on a priori knowledge of the destination of the aircraft (provided by ASDI/ETMS data); knowledge of aircraft destination is available in the form of filed flight plans long before aircraft depart. One significant contribution of this paper is to incorporate this knowledge into the model, unlike previous Eulerian models [9,14,18–20]. 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 (4) and (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 written as follows:

$$x(k+1) = Ax(k) + B^f f(k) + B^u u(k) \tag{7}$$

$$y(k) = Cx(k) \tag{8}$$

where

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

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

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

are all block matrices whose elements are similar to those defined by Eq. (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 1 min of flight time as the size of a cell, the dimension of the state space is 1096 and A is a 1096×1096 matrix. For the network level model (the full continental U.S. airspace), the dimension of the state space is 27,104. Because 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 Eqs. (7) and (8); instead, it is implemented as follows:

$$\begin{gathered} x_{k+1,p,i} = x_{k,p,i-1} + u_{k,p,i} - u_{k,p,i-1}, \qquad k \in \{0, \ldots, N-1\} \\ p \in P, \qquad i \in \{2, \ldots, n_p\} \qquad x_{k,p,1} = f_{k,p} + u_{k,p,1} \\ k \in \{0, \ldots, N\}, \qquad p \in P \end{gathered} \tag{9}$$

where N is the time horizon, P is a set of paths, and 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 scale. For validation purposes, 1 min of flight time is chosen for each cell in this paper, which corresponds to the sampling rate of recorded ASDI/ETMS data. Although this framework requires more space and computational time than existing models (for example, the dimension of the Menon model [9] is the number of control volumes, which is 5 in the example model; the dimension of the dynamic

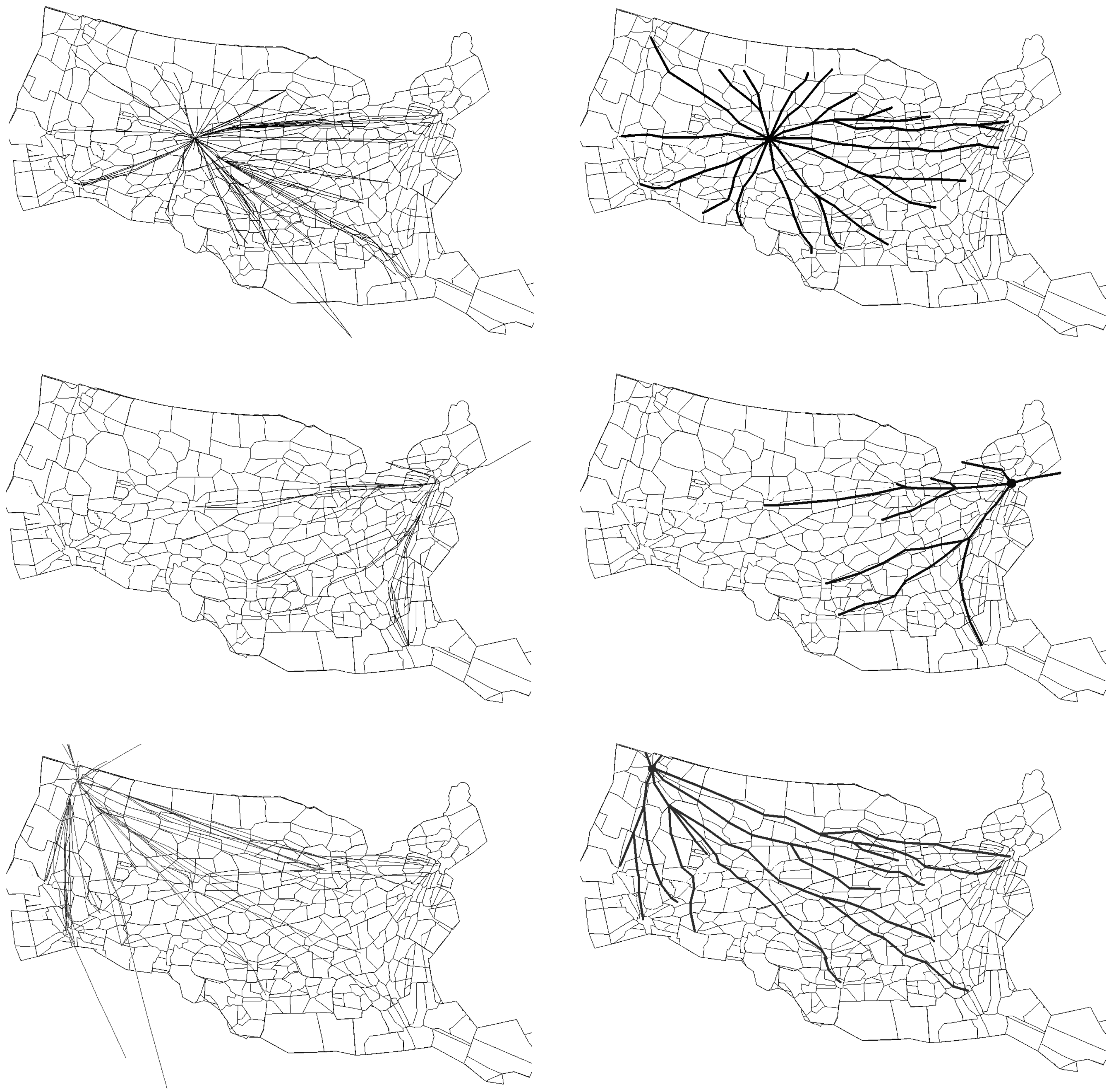

Fig. 8 An illustration of decoupled multicommodity network models by destination for airports (DEN, LGA, SEA). An aggregation of the trees corresponding to the destination airports provides a complete multicommodity network level model. Left: recorded flight tracks, right: corresponding air traffic flow trees.

stochastic model [18] is 23, which is the number of ARTCCs including one for the international region), its scalability greatly facilitates the network model. A full comparison between this model, the PDE model [20], the one-dimensional Menon model [9], and the two-dimensional Menon model [14] is available in paper [31].

4. Controllability and Observability

Controllability and observability play very important roles in control theory [32]. On the 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 important in CTM(L) for TFM. If the system (7) and (8) is not controllable and observable, one cannot guarantee the existence of a feasible solution for a TFM problem formulated with Eq. (23). If a target desired 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 controllability and observability of a dynamical system (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 [32]. For the system (7) and (8), it is easy to show that the controllability and observability matrices have full rank. This means that if there was no constraints on inputs and state (in particular, components of x 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 integer-valued system in the future. In fact, the verification of controllability and observability is related to feasibility checks of a linear (integer) program, which is potentially NP hard in the present case.

B. Flight Rerouting

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 over these three connections, rerouting flights can be achieved. Intercell flows can be determined using a set of laws for different types of intercell connections, described as follows.

1. *Definitions*

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

2. *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:

$$x^j(k+1) = x^i(k) - u^i(k) + u^j(k) \tag{10}$$

where

$$0 \le u^p(k) \le x^p(k), \qquad p = i, j \tag{11}$$

Equation (10) is a simple mass balance, whereas Eq. (11) encodes the fact that one cannot actuate more aircraft than are 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 [12,13].

3. *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:

$$x^l(k+1) = x^i(k) + x^j(k) - u^i(k) - u^j(k) + u^l(k) \tag{12}$$

where

$$0 \le u^p(k) \le x^p(k), \qquad p = i, j, l \tag{13}$$

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

$$x^l(k+1) = \sum_{p\in\mathcal{P}(l)} [x^p(k) - u^p(k)] + u^l(k) \tag{14}$$

where

$$0 \le u^p(k) \le x^p(k), \qquad p \in \mathcal{P}(l) \cup \{l\} \tag{15}$$

4. *Diverge Connection*

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

$$x^j(k+1) = u^{i\to j}(k) + u^j(k), \qquad x^l(k+1) = u^{i\to l}(k) + u^l(k) \tag{16}$$

where

$$\begin{aligned} u^{i\to j}(k) + u^{i\to l}(k) + u^i(k) &= x^i(k), \qquad 0 \le u^p(k) \le x^p(k) \\ p = i, j, l \qquad u^{i\to p}(k) &\ge 0, \qquad p = j, l \end{aligned} \tag{17}$$

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

$$x^p(k+1) = u^{i\to p}(k) + u^p(k), \qquad p \in \mathcal{N}(i) \tag{18}$$

where

$$\begin{aligned} &\sum_{p\in\mathcal{N}(i)} u^{i\to p}(k) + u^i(k) = x^i(k), \qquad 0 \le u^p(k) \le x^p(k) \\ &p \in \mathcal{N}(i) \cup \{i\}, \qquad u^{i\to p}(k) \ge 0, \quad p \in \mathcal{N}(i) \end{aligned} \tag{19}$$

If some link, say the link starting with cell l, is completely shut off, for example, because of weather, special use airspace, congestion, etc., the situation can be modeled by imposing one additional constraint as follows to the constraints in Eq. (19):

$$u^{i\to l}(k) = 0 \tag{20}$$

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

IV. Validation

Validation is the process of testing a model on a data set to demonstrate that the model performs as expected. Demonstration of the accuracy of flow models is obviously key to the process of incorporating them into decision support tools.

In the present case, validation consists of using OD input, that is, 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 ETMS/ASDI 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 0800 GMT on 24 January 2005 to 0800 GMT on 25 January 2005. 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.

A. 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 ASDI/ETMS data have the same trends for all the sectors of interest and differ by an error of a small magnitude. This can be explained as follows. The travel time on a link in the network is computed as the aggregated travel time for all flights in the data set used for the identification (one year in the present work).

To avoid small amplitude, high-frequency fluctuations in the data due to sampling time and boundary crossing, a moving average filter (MAF) technique [33] 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, whereas a large time window filters variations but loses the dynamics of the flow errors. To determine a proper size of time window of the MAF, 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 data (unfiltered) over the course of a simulation. Figure 9 shows the results obtained. It shows how the mean error increases as the time window (number of data points in the average) increases. Note that for most sectors the mean errors are below one aircraft per sector when the time window is 20 min. For this reason, 20 is chosen as the number of data points in the average (the time window, or time span). Removing variation makes physical sense for this problem. Indeed, very often, sector count exceeds legal values for a few minutes (if aircraft are about to exit a sector), which is tolerated in practice as

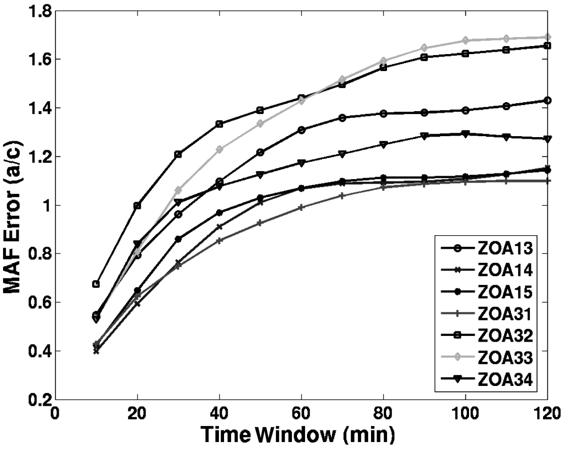

Fig. 9 Average error between the filtered and unfiltered ASDI/ETMS data (sector counts). The mean error increases as the time window increases.

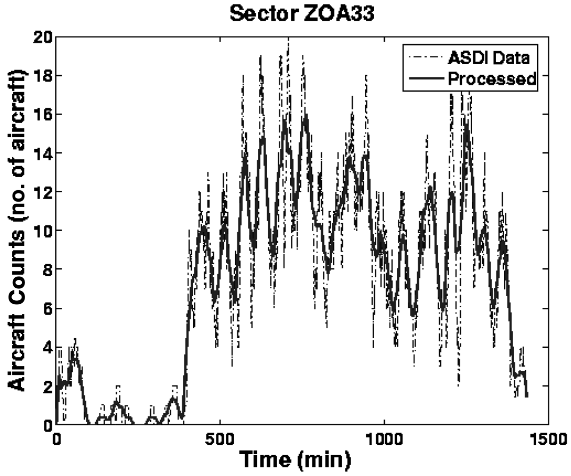

Fig. 10 MAF data processing; the dotted curve represents the unfiltered sector counts of sector ZOA33 and the solid curve represents the filtered data using a time window of 20 min.

such flights usually do not pose significant problems to air traffic controllers.

Figure 10 shows an example of the unfiltered raw data overlayed with the filtered data using MAF, which is more meaningful 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.

Figure 11 shows the predicted and actual sector counts as a function of time in four sectors: medium loaded sectors ZOA32 and ZOA34, highly loaded sector ZOA33, and low-traffic sector ZOA35. The data shown in the figure are filtered by MAF. From the figures it can be seen qualitatively that the model correctly predicts the trends of sector counts.

B. Quantitative Error Analysis

The sector count error analysis involves two comparisons: sum of error breach, S, and the instantaneous error. Following previous work [25], S is defined as the summation of time intervals under the condition that the difference of sector counts between the simulation and ASDI/ETMS data is greater than or equal to a user-specified

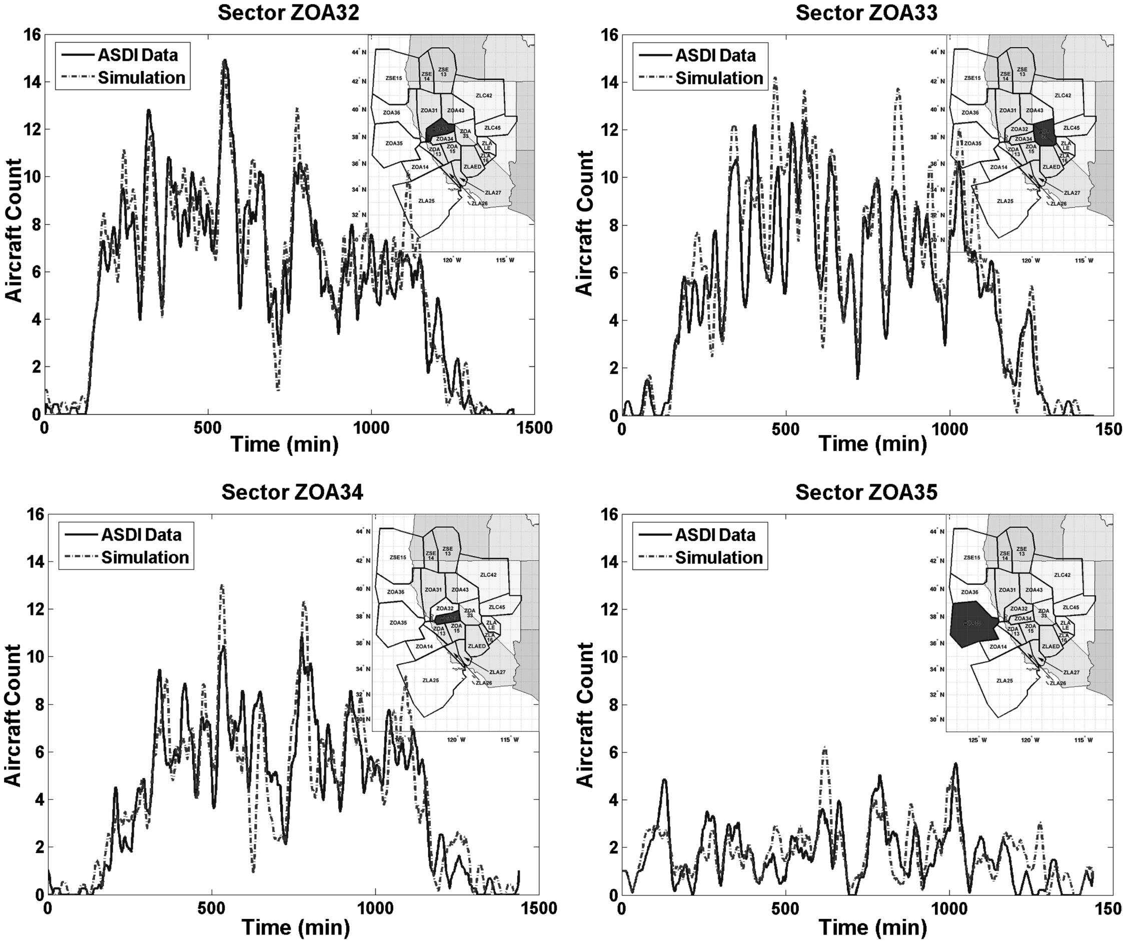

Fig. 11 Comparison of the predictions of aircraft sector counts with the CTM(L) and ASDI/ETMS. Curves represent the processed sector counts after filtering. The map in each figure (shaded) illustrates the location of the corresponding sector.

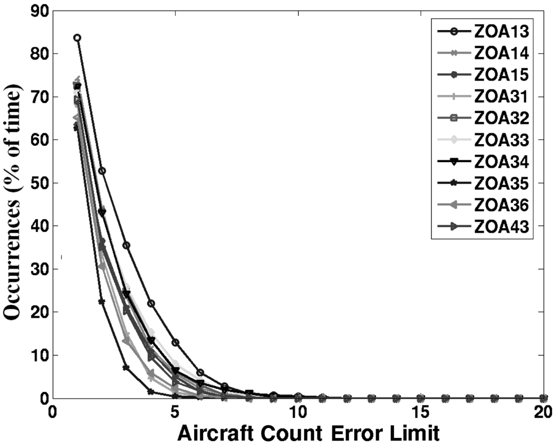

Fig. 12 Occurrences of breach of sector count error for 10 sectors in the Oakland Center.

capacity limitation within a certain time window. This is summarized in Eq. (21):

$$S = \sum_{k=1}^{T} \mathbb{I}_{\{|y_{\text{sim}}(k) - y_{\text{ASDI/ETMS}}(k)| \geq C_s\}} \tag{21}$$

where $\mathbb{I}$ represents the indicator function. The sector count is denoted by $y(k)$, ASDI/ETMS, and simulated. The constant C_s is a user-defined threshold. The time window chosen in the simulation is $T = 1440$ min (24 h). To measure the similarity between the simulation and the ASDI/ETMS data, different values of C_s are used, and plots of the percentage of breaches versus C_s are shown in Fig. 12. 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 it 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 instantaneous error and relative error are shown for sectors ZOA32, ZOA33, and ZOA34 in Fig. 13. From Fig. 13, 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 4 (for a very short period of time), but in general, relative errors are less than one.

A summary of the prediction errors on 1 July 2005 for all the sectors in the study is presented in Sun's thesis [34]. It can be seen that the CTM(L) works very well: for 51% of the sectors, the mean errors are below one aircraft; for 99.65% of the sectors, the mean errors are below two aircraft. The maximum of the mean errors of all sectors is about two (for sector ZTL15), which also has the largest standard deviation.

V. Optimization-Based Sector Count Control

A. Formulation

One of the major features of the model is that it provides a computationally tractable optimization framework for TFM problems. The present section formulates the problem of minimizing overall delay under sector capacity (sector counts) constraints. The goal of such algorithms is high-level TFM actuation to meet FAA defined regulations (limits in sector counts), as optimally as possible (i.e., to limit the delay caused by the imbalance between capacity and demand).

The time horizon of the scenario presented here, of the order of magnitude of 2 h, is discretized in N time steps of length τ. For convenience, $\tau = 1$, which is the time spent by one aircraft in one cell in the absence of ATC actuation. The state of the system at time step $k \in \{0, \ldots, N\}$ 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 multicommodity network [see Eq. (7)]. The control variables are denoted $u_k \in \mathbb{R}^n$ for $k \in \{0, \ldots, N\}$, where u_k represents the number of aircraft held in each cell at time step k [see Eq. (8)]. The input to the system at time step $k \in \{0, \ldots, N\}$ 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 the standard control framework terminology, the input f_k is not controlled, which is an "exogenous forcing" from outside the system [see Eq. (7)].

Using a traditional optimal control formulation [24], the dynamics (4) and (5) become part of the constraints of an MILP formulation:

$$\begin{gathered}
\text{min:} \ \sum_{k=0}^{N} c^T x_k \qquad \text{subject to: } Ex_k + Lu_k \leq M \\
k \in \{0, \ldots, N-1\} \qquad x_N \in \chi_f \qquad x_{k+1} = Ax_k + B^f f_k + B^u u_k \\
k \in \{0, \ldots, N-1\} \qquad x_0 = B^f f_0
\end{gathered} \tag{22}$$

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 described previously; 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 $[\tau, \tau, \ldots, \tau]^T$.

B. Implementation

To solve Eq. (22) in practice, it is necessary to encode it in a computationally efficient manner, given the sparsity of the matrices involved. Flights are clustered on paths, as explained in Sec. III.A. 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 is 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 $i \in \{1, \ldots, n_p\}$ of path $p \in P$ at time step $k \in \{0, \ldots, N\}$. The corresponding control variables are denoted $u_{k,p,i}$ for $k \in \{0, \ldots, N\}$, $p \in P$, and $i \in \{1, \ldots, n_p\}$, 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 $f_{k,p}$ for $k \in \{0, \ldots, N\}$ 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 capacity-controlled sectors, have capacities C_s, $s \in S$. The adapted MILP formulation of the problem is as follows:

$$\begin{gathered}
\text{min:} \ \tau \sum_{k=0}^{N} \sum_{p \in P} \sum_{i=1}^{n_p} x_{k,p,i} \qquad \text{subject to: } \sum_{(p,i) \in I_s} x_{k,p,i} \leq C_s \\
k \in \{0, \ldots, N\}, \qquad s \in S \qquad 0 \leq u_{k,p,i} \leq x_{k,p,i} \\
k \in \{0, \ldots, N\}, \qquad p \in P, \qquad i \in \{1, \ldots, n_p\} \\
x_{k+1,p,i} = x_{k,p,i-1} + u_{k,p,i} - u_{k,p,i-1}, \qquad k \in \{0, \ldots, N-1\} \\
p \in P, \qquad i \in \{2, \ldots, n_p\} \qquad x_{k,p,1} = f_{k,p} + u_{k,p,1} \\
k \in \{0, \ldots, N\}, \qquad p \in P \qquad x_{0,p,i} = 0, \qquad p \in P \\
i \in \{2, \ldots, n_p\} \qquad x_{k,p,i} \in \mathbb{Z}, \qquad k \in \{0, \ldots, N\} \\
p \in P, \qquad i \in \{1, \ldots, n_p\}
\end{gathered} \tag{23}$$

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

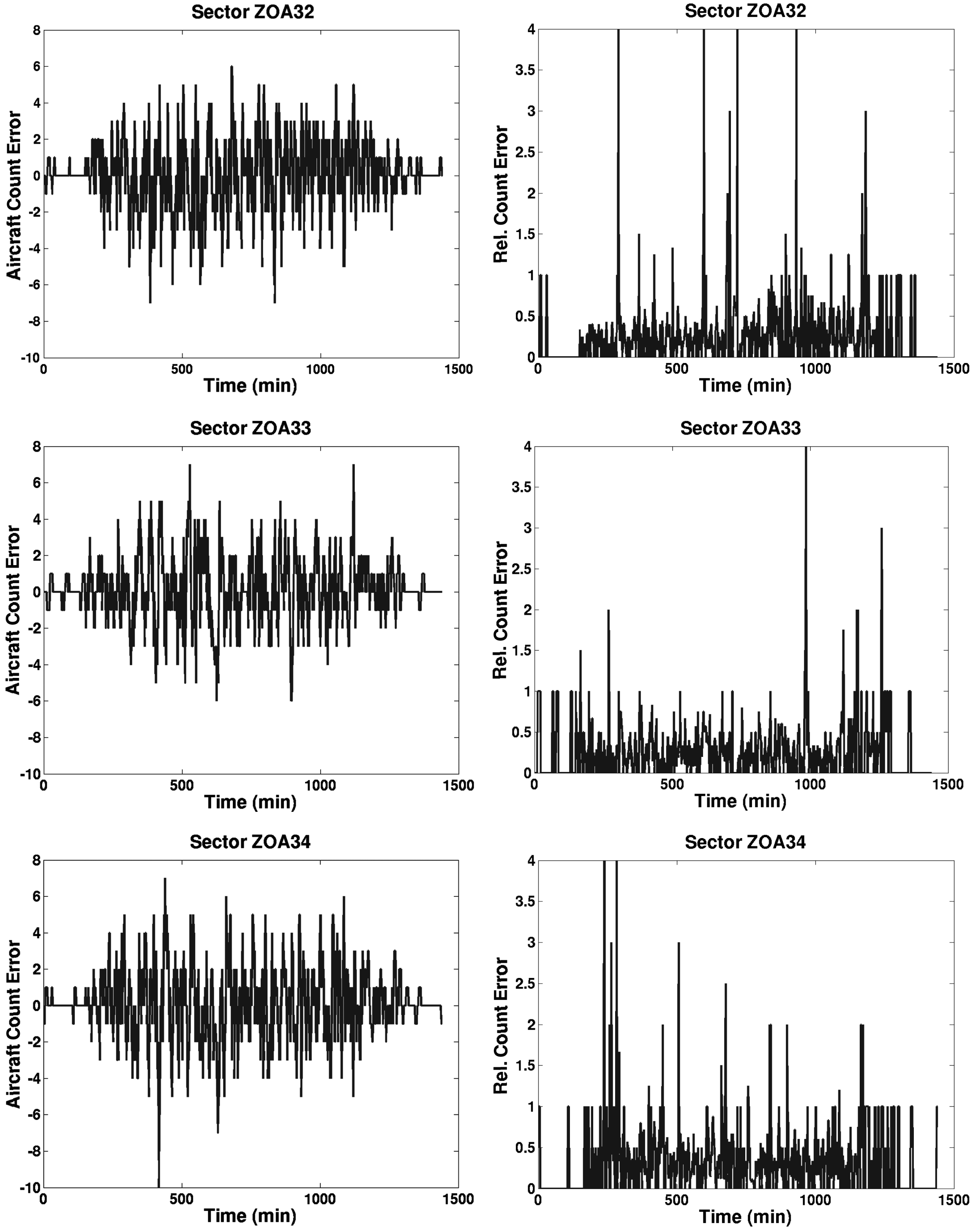

Fig. 13 Left: instantaneous error for three high-altitude sectors (ZOA32, ZOA33, and ZOA34), right: relative error for the three sectors.

number of aircraft in each cell ensures the integrality of the number of aircraft held in each cell, because the inputs to the system are assumed to be integers.

C. Linear Program Relaxation of the MILP Formulation

Because problem (23) cannot be solved in polynomial time deterministically, it is relaxed to a linear program (LP), which is faster to solve in practice and theoretically polynomial time solvable.[‡] The relaxed MILP was solved for instances with more than one million variables and was solved on a statistical sample of 1000 different sets of input parameters. 85% of the runs led to an integer solution. For the remaining 15%, the optimal solution of the LP (OPT_{LP}) was compared with the optimal solution of the corresponding MILP (OPT_{MILP}). The integrality gap α (i.e., $OPT_{\mathrm{MILP}} = \alpha \cdot OPT_{\mathrm{LP}}$), was always smaller than 1.0015. The corresponding solutions are fractional. Several techniques might apply in the future to alleviate this difficulty, in particular LP rounding, which would yield to suboptimal but integer solutions.

On the one hand, there is no guarantee of integrality of the LP solution, but on the other hand, the running time of computing the MILP's solution is not guaranteed. Despite the inherent difficulties of these two approaches, one conclusion can still be guaranteed from the LP approach: when it returns no solution, it provides a certificate

[‡] The usefulness of the guaranteed computational complexity of the LP explicitly in the present case is not assessed. Indeed, the fact that LPs are polynomial time solvable can only be used with a specific analysis of the constant multiplying the corresponding highest order monomial.

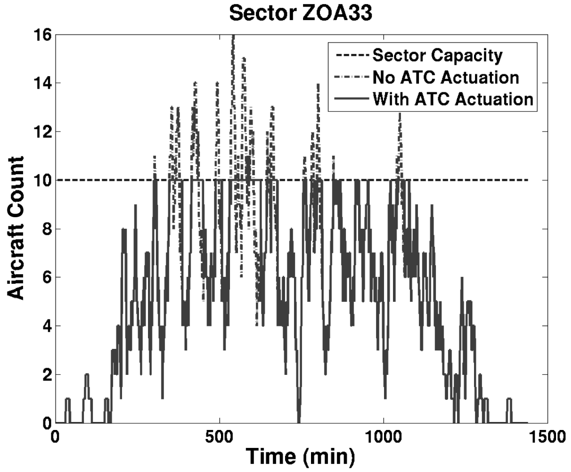

Fig. 14 Example: results of an MILP.

of infeasibility with a guaranteed running time. Also, given the structure of the problem, minimizing the total travel time is equivalent to minimizing the number of holding patterns (i.e., delay). Therefore, the number of holding patterns provided by the LP solution is a lower bound of the number of holding patterns for which there may exist a physical solution. In other words, no ATC actuation can enforce the sector count limitations with less holding patterns than the number of holding patterns provided by the LP relaxation.

D. Results

The relaxed MILP was implemented for controlling sector counts in sector ZOA33 in the Oakland ARTCC (Fig. 2). The threshold value C_s is arbitrarily set to be 10. Figure 14 shows the results solved by th eMILP. It takes 40 min to solve the MILP with AMPL/CPLEX on a 1.4 GHz CPU, 1 GB RAM PC running Linux. In Fig. 14, it can be seen that the method tries to maximize the occupancy of ZOA33, while respecting capacity constraints. As a result, the controlled solution saturates ZOA33 at 10 aircraft for durations that exceed the capacity breach times if there was no control. In no case is the capacity threshold of 10 violated by the solving program (23).

E. Final Note on the Terminology "Large-Capacity Cell Transmission Model"

The model presented in this paper could be viewed as an extension of the seminal Daganzo cell transmission model (widely used in the highway transportation literature [12,13]) for the air traffic control problem. The added term large capacity refers to the specificity of the constraints of the problem. In the present case, capacity (limit on sector counts) is enforced through linear constraints that appear in the optimization programs (22) or (23). This is in contrast to the Daganzo cell transmission model, in which the constraints (highway capacity) are enforced with a (nonlinear) min operator, common in Godunov schemes [35]. In the highway case, the constraints are enforced on each cell individually, whereas here they are enforced on a set of cells. Therefore, the term large capacity expresses that a particular cell can accommodate a large number of aircraft as long as this aggregated number does not exceed the safety capacity. It thus reflects the hard upper bound imposed on counts, which couples the different flows (paths) through the constraints in a linear manner.

VI. Conclusions

A multicommodity Eulerian–Lagrangian large-capacity cell transmission model of airspace was derived 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 for a full day. It was applied to 2-h traffic flow management problems. The problem of controlling the sector aircraft count was posed as an integer program in which the dynamical system appears in the constraints. To improve computational efficiency of the model, the integer program was relaxed to a linear program solved for instances with more than one million variables and showed good performance for a high-level traffic flow management to minimize delays in the presence of an imbalance between demand and capacity.

Acknowledgments

This work was supported by NASA under task order TO.048.0.BS.AF. The authors are thankful to Carlos Daganzo for his advice regarding the model. The authors are thankful to Shon Grabbe and Kapil Sheth for their help with ASDI/ETMS data and FACET. The authors are grateful to Gano Chatterji for his suggestions regarding the models of the National Airspace System. The authors want to acknowledge George Meyer and Banavar Sridhar for their vision on modeling, fruitful conversations on air traffic control, and ongoing support for this research; and Charles-Antoine Robelin for his help with generating the initial model, sharing his source code, and helping with the graphs presented in this paper. The authors are grateful to Larry Hogle for helping with the collaboration between University of California, Berkeley and NASA Ames Research Center through the University Affiliated Research Center. The authors are grateful to Dave Knorr from FAA for fruitful discussions.

References

[1] Mao, Z. H., Feron, E., and Bilimoria, K., "Stability and Performance of Intersecting Aircraft Flows Under Decentralized Conflict Avoidance Rules," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 2, No. 2, June 2001, pp. 101–109. doi:10.1109/6979.928721

[2] Devasia, S., Heymann, M., and Meyer, G., "Automation Procedures for Air Traffic Management: A Token-Based Approach," *Proceedings of the American Control Conference*, American Automatic Control Council, Dayton, OH, May 2002, pp. 736–741.

[3] Mueller, K. T., Schleicher, D. R., and Bilimoria, K., "Conflict Detection and Resolution with Traffic Flow Constraints," AIAA Paper 2002-4445, Aug. 2002.

[4] Bilimoria, K., and Lee, H. Q., "Aircraft Conflict Resolution with an Arrival Time Constraint," AIAA Paper 2002-4444, Aug. 2002.

[5] Resmerita, S., Heymann, M., and Meyer, G., "A Framework for Conflict Resolution in Air Traffic Management," *Proceedings of the 42nd IEEE Conference on Decision and Control*, Institute of Electrical and Electronics Engineers, New York, Dec. 2003, pp. 2035–2040.

[6] Resmerita, S., and Heymann, M., "Conflict Resolution in Multi-Agent Systems," *Proceedings of the 42nd IEEE Conference on Decision and Control*, Institute of Electrical and Electronics Engineers, New York, Dec. 2003, pp. 2537–2542.

[7] Paielli, R., and Erzberger, H., "Improved Conflict Detection for Reducing Operational Errors in Air Traffic Control," AIAA Paper 2004-6392, Sept. 2004.

[8] Kopardekar, P., and Green, S., "Airspace Restriction Planner for Sector Congestion Management," AIAA Paper 2005-7435, Sept. 2005.

[9] Menon, P. K., Sweriduk, G. D., and Bilimoria, K., "New Approach for Modeling, Analysis and Control of Air Traffic Flow," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 5, 2004, pp. 737–744.

[10] Bilimoria, K., Sridhar, B., Chatterji, G., Sheth, K., and Grabbe, S., "FACET: Future ATM Concepts Evaluation Tool," *3rd USA/Europe ATM 2000 R&D Seminar*, Napoli, Italy, June 2000.

[11] Bertsimas, D., and Stock, S., "The Air Traffic Flow Management Problem with Enroute Capacities," *Operations Research*, Vol. 46, No. 3, May–June 1998, pp. 406–422.

[12] Daganzo, C., "The Cell Transmission Model: A Dynamic Representation of Highway Traffic Consistent with the Hydrodynamic Theory," *Transportation Research. Part B*, Vol. 28, No. 4, 1994, pp. 269–287. doi:10.1016/0191-2615(94)90002-7

[13] Daganzo, C., "The Cell Transmission Model, Part 2: Network Traffic," *Transportation Research. Part B*, Vol. 29, No. 2, 1995, pp. 79–93. doi:10.1016/0191-2615(94)00022-R

[14] Menon, P. K., Sweriduk, G. D., Lam, T., Diaz, G. M., and Bilimoria, K.,

"Computer-Aided Eulerian Air Traffic Flow Modeling and Predictive Control," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 1, 2006, pp. 12–19.

[15] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C., *Introduction to Algorithms*, 2nd ed., Prentice–Hall, Upper Saddle River, NJ, 2002, pp. 788–789.

[16] Lighthill, M. J., and Whitham, G. B., "On Kinematic Waves. 2. A Theory of Traffic Flow on Long Crowded Roads," *Proceedings of the Royal Society of London*, Vol. 229, No. 1178, 1956, pp. 317–345.

[17] Richards, P. I., "Shock Waves on the Highway," *Operations Research*, Vol. 4, No. 1, 1956, pp. 42–51.

[18] Roy, S., Sridhar, B., and Verghese, G. C., "An Aggregate Dynamic Stochastic Model for Air Traffic Control," *5th USA/Europe ATM 2003 R&D Seminar*, Budapest, Hungary, June 2003.

[19] Sridhar, B., Soni, T., Sheth, K., and Chatterji, G., "An Aggregate Flow Model for Air Traffic Management," AIAA Paper 2004-5316, Aug. 2004.

[20] Bayen, A. M., Raffard, R., and Tomlin, C., "Adjoint-Based Control of a New Eulerian Network Model of Air Traffic Flow," *IEEE Transactions on Control Systems Technology*, Vol. 14, No. 5, 2006, pp. 804–818. doi:10.1109/TCST.2006.876904

[21] Saraf, A., and Slater, G., "Combined Eulerian–Lagrangian Two-Level Control System for Optimal Air Traffic Flow Management," AIAA Paper 2006-6229, Aug. 2006.

[22] Iamratanakul, D., Meyer, G., Chatterji, G., and Devasia, S., "Quantification of Airspace Sector Capacity Using Decentralized Conflict," *Proceedings of the 43rd IEEE Conference on Decision and Control*, Institute of Electrical and Electronics Engineers, New York, Dec. 2004, pp. 2003–2009.

[23] Heymann, M., Meyer, G., and Resmerita, S., "An Agent Based Framework for Control of Merging Air Traffic," *Proceedings 16th IFAC Symposium on Automatic Control in Aerospace*, International Federation of Automatic Control, Laxenburg, Austria, June 2004.

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

[25] Robelin, C. A., Sun, D., Wu, G., and Bayen, A. M., "MILP Control of Aggregate Eulerian Network Airspace Models," *Proceedings of the American Control Conference*, American Automatic Control Council, Dayton, OH, June 2006, pp. 5257–5262.

[26] Hu, S. R., Madanat, S., Krogmeier, J., and Peeta, S., "Estimation of Dynamic Assignment Matrices and OD Demands Using Adaptive Kalman Filtering," *Journal of Intelligent Transportation Systems*, Vol. 6, No. 3, Jan. 2001, pp. 281–300. doi:10.1080/10248070108903696

[27] Ahuja, R. K., Magnati, T. L., and Orlin, J. B., *Network Flows: Theory, Algorithms, and Application*, Prentice–Hall, Upper Saddle River, NJ, 1993.

[28] Robelin, C. A., Sun, D., Wu, G., and Bayen, A. M., "Strategic Traffic Flow Models Based on Datamining and System-Identification Techniques," NASA (to be published).

[29] Hoffman, B., Krozel, J., Penny, S., Roy, A., and Roth, K., "A Cluster Analysis to Classify Days in the National Airspace System," AIAA Paper 2003-5711, Aug. 2003.

[30] Papageorgiou, M., "Dynamic Modeling, Assignment, and Route Guidance in Traffic Networks," *Transportation Research. Part B*, Vol. 24, No. 6, 1990, pp. 471–495. doi:10.1016/0191-2615(90)90041-V

[31] Sun, D., Strub, I., and Bayen, A. M., "Comparison of the Performance of Four Eulerian Network Flow Models for Strategic Air Traffic Management," *AIMS Journal on Networks and Heterogeneous Media*, Vol. 2, No. 4, Dec. 2007, pp. 569–594.

[32] Nise, N. S., *Control Systems Engineering*, Wiley, New York, 2004, pp. 728–732.

[33] Smith, S. W., *The Scientist and Engineer's Guide to Digital Signal Processing*, California Technical Publishing, San Diego, CA, 1997, pp. 277–284.

[34] Sun, D., "Network Based Optimization Algorithms for Traffic Flow Management," Ph.D. Thesis, Univ. of California, Berkeley, CA, May 2008 (to be published).

[35] Strub, I., and Bayen, A., "Weak Formulation of Boundary Conditions for Scalar Conservation Laws: An Application to Highway Modeling," *International Journal of Robust and Nonlinear Control*, Vol. 16, No. 16, 2006, pp. 733–748. doi:10.1002/rnc.1099