Proceedings of the 2006 American Control Conference
Minneapolis, Minnesota, USA, June 14-16, 2006

MILP control of aggregate Eulerian network airspace models*

Charles-Antoine Robelin[1], Dengfeng Sun[2], Guoyuan Wu[3] and Alexandre M. Bayen[4]

***Abstract*— A new Eulerian model of airspace is derived and applied to high altitude traffic for a full Air Traffic Control Center of the National Airspace System. The Eulerian model is reduced to a linear time invariant dynamical system, in which the state is a vector of aggregate aircraft counts. The model is validated against ASDI data and applied to the Oakland airspace. The problem of controlling sector aircraft count is posed as an Integer Program, in which the dynamical system appears in the constraints. To improve the computational time of calculating the solution, the Integer Program is relaxed to a Linear Program, solved for instances with more than one million variables. The computational results show that a high proportion of solutions of the LP are integers. The computational time is satisfactory for two hour Traffic Flow Management problems.**

I. Introduction

The almost uninterrupted growth of US air traffic over the last few decades 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. ATC is operated at the sector level, where a sector is a small portion of the airspace controlled by a single human Air Traffic Controller. *Traffic Flow Management* (TFM) typically deals with traffic at the *Center* level, i.e. 10 to 20 sectors. TFM problems include maintaining the aircraft count in each sector below a legal threshold in order to ease the human ATC workload, as well as to ensure the safety of the flights [1]. 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 [2]. This is in contrast to *Lagrangian* models, which are trajectory-based and take into account all aircraft trajectories.

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 only on the size of the physical problem of interest. (ii) Their control-theoretic structure enables the use of standard methodologies to analyze them. This article presents a new model, developed jointly with NASA Ames, demonstrating the two benefits outlined above.

The seminal Eulerian model proposed for TFM is presented in Menon *et al.* [2]. This model is based on a discretized version of the *Lighthill-Whitham-Richards* (LWR) *partial differential equation* (PDE) [3], [4], inspired by the Daganzo cell transmission model, which is traditionally used in highway traffic modeling [5], [6]. The model presented in [2] has been extended to a stochastic framework [7], [8] to account for expected aircraft count values. One important characteristic of the three approaches [2], [7], [8] is the diffusion that occurs in these models. While this is not a problem in a stochastic framework (since the results are in the expected sense), this is more problematic for the original model [2], since it potentially leads to aircraft losses or inaccurate predictions (see [9], [10] for more details). A first attempt to resolve these issues was proposed as a continuous time continuous space model in [9], [10], based directly on the LWR PDE. While this approach solves the diffusion problem, its computational tractability is disputable (it depends on the required space discretization). We propose a discrete space discrete time Eulerian model of the airspace with no diffusion, in the form of an integer linear dynamical system, which is computationally less expensive.

We first outline the construction of a graph-theoretic model of traffic flow. Air traffic flow on this graph is modeled as a discrete time dynamical system (Section II). This model is validated using *Aircraft Situation Display to Industry* (ASDI) data and *Future ATM Concepts Evaluation Tool* (FACET) [11]. In particular, we show that the metric of interest for TFM (aircraft count) is reproduced adequately by the model (Section III). We then pose the problem of controlling aircraft counts in the airspace as a *Mixed Integer Linear Program* (MILP), where the dynamical system appears in the constraints (Section IV), as is traditionally done in optimal control [12]. Numerical experiments are run to demonstrate the tractability of these methods for problems involving more than one million integer variables (Section V). The running time is improved by relaxing the MILP to a *Linear Program* (LP). The computational results show a high proportion of integer solutions to the LP, appropriate for the application of interest. The resulting computational time is satisfactory for two hour Traffic Flow Management problems (a few minutes for a two hour window).

Our main contribution is thus a new Eulerian model appropriate for TFM forecast and actuation. Experiments suggest that the MILP solution for one million variables is tractable in real time (a few minutes for two hour TFM). When the relaxed MILP problem (solved as a LP) is infeasible, our method provides a guaranteed running time algorithm to prove infeasibility of the aircraft count control problem.

*This work is supported by NASA, under Task Order TO.048.0.BS.AF.

[1]Ph.D. candidate, Dept. of Civil and Environmental Eng., Univ. of California, Berkeley CA, 94720-1710, USA. Phone: 510-642-7390. Fax: 510-643-5264. Email: robelin@berkeley.edu. Corresponding author.
[2]Ph.D. student, Mechanical Eng., Univ. of California, Berkeley.
[3]Ph.D. student, Civil and Environm. Eng., Univ. of California, Berkeley.
[4]Assistant Professor, Dept. of Civil and Environmental Engineering, University of California, Berkeley.

II. Automated Model Building

A. Data Aggregation

ASDI data provides flight information for all airborne aircraft at a given time, updated every minute. It includes a time stamp and flight data (latitude, longitude, heading, altitude, etc.).

The objective of automated model building is to construct a graph-theoretic model of air traffic flow directly from the ASDI files. Several pattern recognition methods have been implemented to automatically build a graph 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 data (e.g. flight plan data). None of these algorithms have provided satisfactory results for practical purposes, thus leading to another approach outlined later in this section. We summarize below the several approaches investigated.

1) *K-Means* [13]. 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 [13] assigns data points to clusters by finding the nearest cluster mean and assigning the data point to that cluster. The K-means algorithm has a time complexity which is linear in the number of data points.
2) *Generalized Principal Component Analysis (GPCA) Algorithm* [14], [15]. GPCA is an algebraic geometric approach to problems of estimating clusters from sample data points. GPCA automatically determines the number of clusters and, unlike [13], does not need this information *a priori*. Additionally, the complexity of GPCA is determined by the algorithm's identifying the number of clusters and corresponding basis (characterizing the clusters) and grouping the data points into the clusters to which they belong. It does not involve iterative procedures as many other clustering algorithms do; once the number of clusters is determined, the complexity of the algorithm is the cost of solving a linear system of equations.
3) *Flight plan based algorithms* [16]. A flight goes from a departure airport to an arrival airport by traveling through a set of fixes (waypoints). We classify the trajectories based on sequences of waypoints. Waypoint-based flow pattern classification should be noise-free, since the trajectory of a flight is defined by strings. Similarly, we also used jetways as a classification criteria. Because jetways are similar to highways, they act as guidelines that the flights should follow. The flights using the same jetway are supposed to use similar trajectories.

There are numerous explanations for the failure of all algorithms above. (i) The mathematical optimum for all of these methods is not relevant for practical purposes; in other words, a suboptimal solution might be physically more relevant than the optimum. (ii) The noise in 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 acronyms).

In addition to these general considerations, specific reasons prevent the above algorithms from being applicable. (iii) For example, the K-means algorithm requires *a priori* knowledge of the number of clusters, on which we cannot rely in practice. Furthermore, it is extremely sensitive to an initial guess, which makes it hard to use in an automated manner. (iv) Waypoint-based classification is inappropriate because of the extremely large number of different waypoint acronyms. Even though this data is noise-free, its size is prohibitive for the present study. (v) Jetway-based classification is not applicable, since ASDI data does not provide us with the location of the entry point of an aircraft onto a jetway, leading to the well-known underconstrained OD estimation problem in highway traffic [17]. A summary and analysis of all results obtained with the techniques stated above is available in [16].

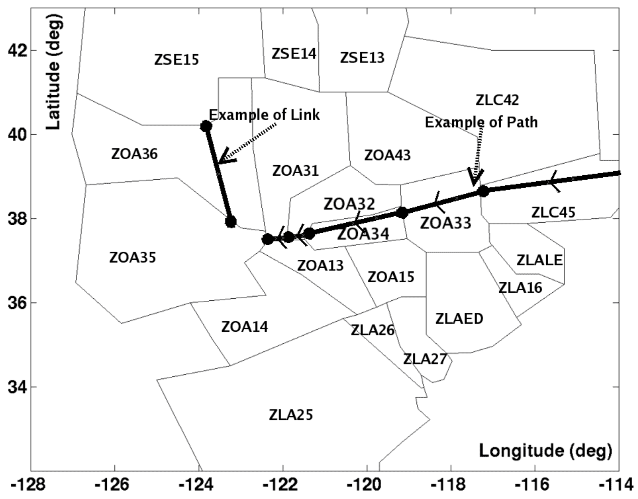

Figure 1. *Map of the ZOA (Oakland) Center.* **Example of Path***: A path in ZOA that a flight follows when it comes from BOS to SFO. It includes five consecutive links in ZOA;* **Example of Link***: A link in sector ZOA36.*

As a result, we developed an algorithm that takes the geographic structure of the airspace as a starting point for building the desired air traffic flow graph. ATC sector frontiers are used as a discriminating boundary to isolate subgraphs that are interconnected at nodes along the boundary (see Figure 1). The subgraphs consist of trajectory clusters assembled into *links* or *edges*. These subgraphs are connected to each other through the boundaries. Figures 2 and 3 show the graph by sector-based air traffic flows classification. More details about the algorithm are provided in [16].

The computational complexity of building the graph is proportional to the number of links in the network. Given the structure of the U.S. airspace, this number is itself proportional to the number of sectors (i.e. subgraphs) considered in the model, since there exists a constant upper bound to the number of links in one sector, independent from other parameters of the model. Therefore, the computational complexity of building the graph is $O(n_{\text{sectors}})$, where n_{sectors} is the number of sectors considered.

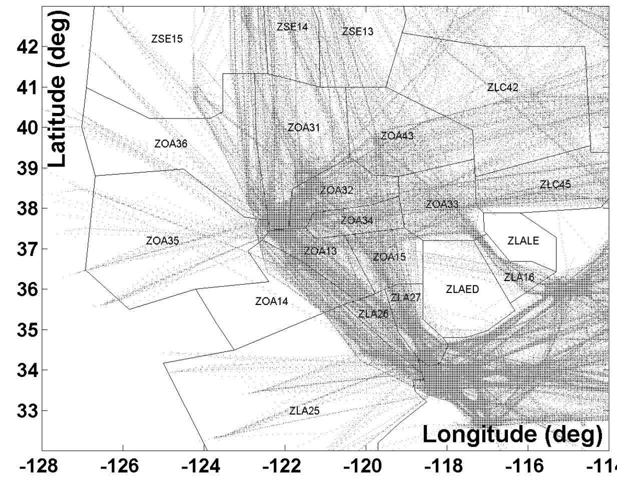

Figure 2. *Flight tracks in Oakland Center and nearby airspace, for 24-hour of ASDI data.*

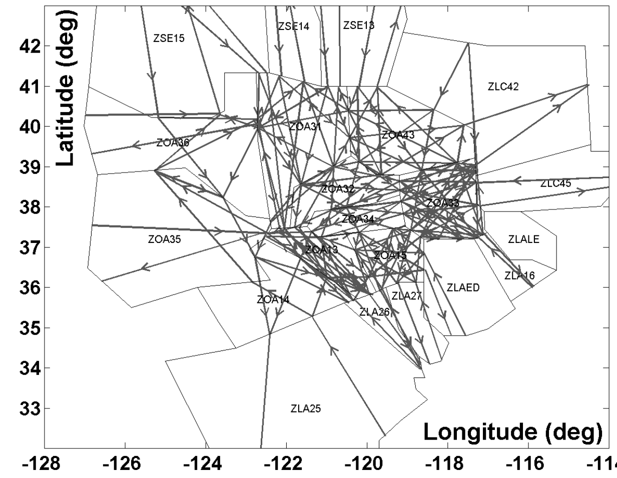

Figure 3. *Graph model constructed by our algorithm to represent the flow patterns of Figure 2.*

B. Distribution of Travel Time

For each link of the graph depicted in Figure 3, we aggregate the flight times for the seven days of ASDI data. 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 that link. The data shows that the distribution of travel time approximately follows a truncated Gaussian distribution (see Figure 4). It is "truncated" because flight times can be lengthened by ATC, but in general not shortened. The expected travel time of a flight through a link is used to determine the length of the link. In our ATC model, we divide every link 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.

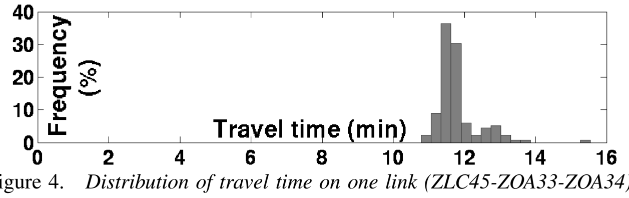

Figure 4. *Distribution of travel time on one link (ZLC45-ZOA33-ZOA34): truncated Gaussian.*

The complexity of the determination of the graph and the estimation of the travel time on all links is $O(n_{\text{sectors}}.n_{\text{days}})$, where n_{sectors} is the number of airspace sectors considered, and n_{days} is the number of days of data used for the estimation. On a 3GHz CPU, 512MB RAM PC running Linux, model building takes approximately one minute per day of data, for a network of 20 airspace sectors covering a tenth of the surface of the continental U.S.

III. Model Description and Validation

A. Model Description

A state space model of air traffic flow is first developed at the link level, where a link is understood in the graph-theoretic sense, i.e. an edge of a graph (see Figures 3 and 5). Under the assumption that air traffic flow can be accurately represented by an aggregated travel time, the behavior of aircraft flows on a single link can be modeled by a deterministic linear model with unit time delay, defined as follows.

$$\begin{aligned} x_i(k+1) &= A_i x_i(k) + B_i^f f_i(k) + B_i^u u_i(k) \quad (1)\\ y(k) &= C_i x_i(k) \quad (2)\end{aligned}$$

where $x_i(k) = [x_i^{m_i}(k), ..., x_i^1(k)]^T$ 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 which denotes the entry count onto link i during the 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 C_i correspond to the cells in the user-specified set, and are equal to 1. A_i is an $m_i \times m_i$ nilpotent matrix with 1's on its super-diagonal. $B_i^f = [0, ..., 0, 1]^T$ is the forcing vector with m_i elements, and B_i^u is the $m_i \times m_i$ delay matrix, in which all nonzero elements are 1 on the diagonal and -1 on the super-diagonal. It is easy to see that, if (1) – (2) is unconstrained, the system is controllable and observable. However, it is not the case in practice since only nonnegative integer inputs and states are allowed (otherwise the model is not physical). An illustration of the model is shown in Figure 5, for one link.

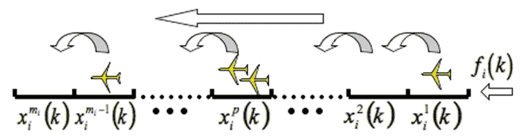

Figure 5. *Illustration of the model of a link as a delay system: everywhere in the link, $x_i^{p+1}(k+1) = x_i^p(k)$, unless some control action was applied.*

It is straightforward to extend this modeling technique to set up a sector level model, for there is no interconnection (neither inputs, nor states) between different links in one sector. Suppose there are n links in a sector, then the state space equations at the sector level are as follows.

$$\begin{aligned} x(k+1) &= Ax(k) + B^f f(k) + B^u u(k) \quad (3)\\ y(k) &= Cx(k) \quad (4)\end{aligned}$$

where $x(k) = [x_n(k), ..., x_1(k)]^T$ denotes the state, and $f(k) = [f_n(k), ..., f_1(k)]^T$ is the [forcing] input vector, i.e.

the entry count onto the sector. The [control] input vector is denoted $u(k) = [u_n(k), ..., u_1(k)]^T$. $y(k)$ 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, such that $A = diag(A_n, \dots, A_1)$, $B^f = diag(B^f_n, \dots, B^f_1)$, and $B^u = diag(B^u_n, \dots, B^u_1)$. The vector C is given by $[C_n, ..., C_1]$. The quantities, $x_i(k)$, $f_i(k)$, $u_i(k)$, A_i, B^f_i, B^u_i and C_i are all defined by Equations (1) – (2). Note that the system (3) – (4) is also controllable and observable with the same limitations as (1) – (2).

When a Center level model is created, it is necessary to include *merge/diverge* nodes in the network [2], [7], [8], [9]. Merge nodes are straightforward: flows are added as streams of aircraft merge. For diverge nodes, the corresponding routing choice must rely on knowledge of aircraft destination. Several approaches have been proposed to solve this problem, in particular *split coefficients*, used by Menon *et al.* [2], inspired from the highway transportation literature [6]. We propose an alternate manner to model this problem, based on *a priori* knowledge of the destination of the aircraft (provided to us by ASDI data). 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, for example, from ZLC45 to San Francisco (see Figures 1 and 3). If two or more paths have one link in common, this link will be duplicated. Therefore, the Center level model can also be cast in the (3) – (4) framework, 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 considered center. The output, $y(k)$, denotes the aircraft count in a user-specified set of cells at time step k. Although this framework requires more space and computational time, it greatly facilitates the network model by decoupling the state or input variables around the *merge/diverge* nodes.

B. Validation procedure and limit of the model

Validation of the model against ASDI data. A simulation is performed to validate the model against ASDI data. Sector ZOA33 (see Figures 1 and 3) is used for the validation. ASDI data, from 8:00am GMT on January 24th, 2005 to 8:00am GMT on January 25th, is used. The input, $f(k)$, to the model is the number of aircraft entering the considered region during the unit time time interval from k to $k + 1$. The predicted state is computed from the model for k in the range above (one minute increments) and compared with the actual data in the network. The cumulative entry counts for both simulation results and ASDI data are shown in Figure 6 (top). A comparison of aircraft counts in sector ZOA33 is presented in Figure 6 (bottom).

Analysis of the results. As is shown in Figure 6 (top), the two curves match and display the general trends. The largest difference between them is by eight flights. It is also noted that the aircraft count is preserved throughout the simulation. In Figure 6 (bottom), the sector count of our model and ASDI data differ by noise of a non-negligible magnitude, because the model uses the expected travel time.

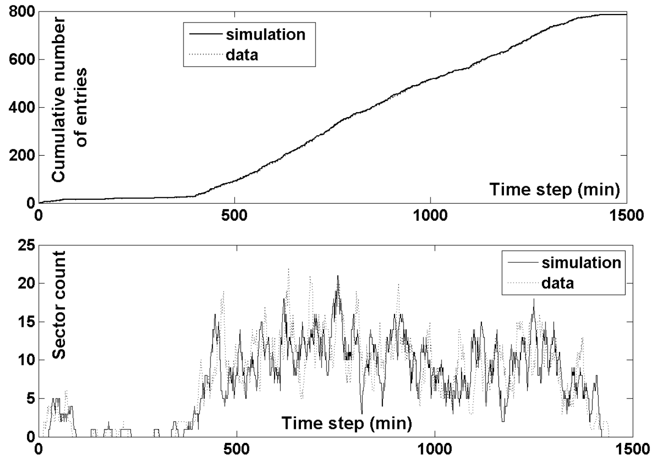

Figure 6. *Cumulative entry count onto ZOA33 (top), and comparison of sector count in ZOA33 (bottom). An animated visualization of the model validation is available for download at [18].*

Correlation coefficient. We calculate the length of threshold breach, i.e. the summation of time intervals under the condition that sector counts are greater than or equal to a user-specified capacity limitation, within a certain time window.

$$S = \sum_{k=k_0}^{k_0+\Delta k} \mathbb{I}_{\{y(k) \geq C_s\}} \tag{5}$$

where $\mathbb{I}$ represents the indicator function. The sector count, $y(k)$ is defined by (4), and C_s is a user-defined capacity limitation. The time window we choose in our simulation is 15 minutes, i.e. $\Delta k = 14$. In Equation (5), S can be computed either using ASDI data (denoted S_{data}) or by the model predictions (denoted S_{model}). To measure the similarity in the length of threshold breach between the simulation and ASDI data, with different values of C_s, the relationship between the correlation coefficient and the user-defined capacity limitation in the specified sector is presented in Figure 7. The decreasing tendency of the correlation coefficient against the user-defined capacity limitation is intuitive: for large values of the limitation, S will only be non-zero during brief periods and thus S_{model} and S_{data} will have no correlation because of the noise in the data. It can be seen from Figure 7 that the correlation between our model and ASDI data is above 60% if the user-defined threshold is less than 12 flights.

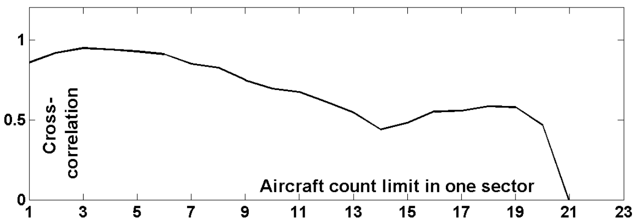

Figure 7. *Correlation of threshold breach duration vs. aircraft count limit*

IV. MILP Formulation of Two Hour TFM

A. Formulation

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

The time horizon of the problem, of the 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 actuation. 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 during the unit time interval from k to $k+1$. 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 during the unit time interval from k to $k+1$ is represented by the vector $f_k \in \mathbb{R}^n$. Note that, unlike in 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 traditional optimal control framework such as in [12], the dynamics (3)-(4) becomes part of the constraints of the MILP formulation:

$$\begin{aligned}
&\textbf{min:} \quad \sum_{k=0}^{N} c^T x_k \\
&\textbf{subject to:} \\
&Ex_k + Lu_k \leq M,\ k\in \{0,\cdots,N-1\} \\
&x_N \in \chi_f \\
&x_{k+1} = Ax_k + B^f f_k + B^u u_k,\ k\in \{0,\cdots,N-1\} \\
&x_0 = B^f f_0
\end{aligned} \tag{6}$$

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, the number of aircraft held in a cell cannot be greater than the number of aircraft in that cell, and the elements of x_k and u_k are non-negative for all k. 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

In order to solve (6) in practice, we need to encode it in a computationally efficient manner, which we now present. Flights are clustered on *paths*, as explained in section III. The set P of paths is determined from the data, as well as the number n_p of cells along path $p \in P$. 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 $i\in \{1,\cdots,n_p\}$ of path $p \in P$ at time step $k\in \{0,\cdots,N\}$. The corresponding control variables are denoted $u_{k,p,i}$, 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 $f_{k,p}$ for $k\in \{0,\cdots,N\}$, and $p \in P$, where $f_{k,p}$ represents the number of aircraft entering 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.

$$\begin{aligned}
&\textbf{min:} \quad \tau \sum_{k=0}^{N} \sum_{p\in P} \sum_{i=1}^{n_p} x_{k,p,i} \\
&\textbf{subject to:} \\
&\sum_{(p,i)\in I_s} x_{k,p,i} \leq C_s,\ k\in \{0,\cdots,N\},\ s \in S \\
&u_{k,p,i} \leq x_{k,p,i}, \\
&\qquad k\in \{0,\cdots,N\},\ p \in P,\ i\in \{1,\cdots,n_p\} \\
&x_{k+1,p,i} = x_{k,p,i-1} + u_{k,p,i} - u_{k,p,i-1}, \\
&\qquad k\in \{0,\cdots,N-1\},\ p \in P,\ i\in \{2,\cdots,n_p\} \\
&x_{k,p,1} = f_{k,p} + u_{k,p,1},\ k\in \{0,\cdots,N\}, p \in P \\
&x_{0,p,i} = 0,\ p \in P,\ i\in \{2,\cdots,n_p\} \\
&x_{k,p,i} \in \mathbb{Z}, k\in \{0,\cdots,N\},\ p \in P,\ i\in \{1,\cdots,n_p\}
\end{aligned} \tag{7}$$

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 inputs to the system are assumed to be integers.

C. LP relaxation of the MILP formulation

Since (7) cannot be solved in polynominal time deterministically, it is relaxed to a Linear Program, 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 lead to an integer solution. For the remaining 15 percent of the runs, the optimal solution of the LP (OPT_{LP}) was compared to the optimal solution of the corresponding MILP (OPT_{MILP}). The integrality gap α (given by $OPT_{\text{MILP}} = \alpha \cdot OPT_{\text{LP}}$) was always smaller then 1.0015. However, the corresponding solutions are fractional, thus impractical.

On one hand, there is no guarantee of integrality of the LP solution, but on the other hand, the running time of computing the MILP solution is not guaranteed. Despite the flaws 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 with guaranteed running time. Also, given the structure of the problem, minimizing the total travel time is equivalent to minimizing the number of delay controls. Therefore, the number of delay controls provided by the LP solution is the lower bound of the number of delay controls for which there may exist a physical solution. In other words, no Air Traffic Control actuation can enforce the sector count limitations with less delay than the number of delay controls provided by the LP relaxation.

V. Computational Results

We consider two types of control scenarios and three networks with different sizes. The control scenarios are: (a) only sector ZOA33 is controlled; (b) sectors ZOA33 and ZOA34 are controlled. The three networks consist of 11, 16 and 21 sectors, respectively. Applying two control scenarios on each of the three networks, we consider six types of network structures.

[1]We did not assess the usefulness of the guaranteed computational complexity of LP explicitely 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 the corresponding higher order monomial.

The optimization programs are solved using CPLEX and the modeling language AMPL [19]. We compare the measured CPU time[2] for solving the LP and MILP formulations (see section IV). We run simulations with a time horizon of two hours. Shifting the time window by 20 minute increments, we have 67 simulations per day of data, and the simulations are done for six different days. The statistics of CPU times are summarized below.

	11 sectors	16 sectors	21 sectors
scenario (a)	104 (seconds)	207 (seconds)	222 (seconds)
scenario (b)	147 (seconds)	299 (seconds)	303 (seconds)

Table 1: LP: average CPU time time for different network structures.

	11 sectors	16 sectors	21 sectors
scenario (a)	216 (seconds)	442 (seconds)	474 (seconds)
scenario (b)	258 (seconds)	579 (seconds)	597 (seconds)

Table 2: LP: standard deviation (σ) of CPU time

network size (sectors)	11	16	21
Scenario (a)			
percentage of instances within 1σ	91.4%	92.0%	91.6%
percentage of instances within 2σ	94.6%	95.1%	95.8%
percentage of instances within 3σ	97.1%	97.0%	97.3%
Scenario (b)			
percentage of instances within 1σ	87.8%	90.9%	91.0%
percentage of instances within 2σ	95.4%	95.1%	95.8%
percentage of instances within 3σ	98.1%	98.1%	97.6%

Table 3: LP: percentage of instances within $n\ \sigma$ ($n = 1, 2, 3$).

network size (sectors)	11	16	21
mean CPU time (seconds)	389	522	637
standard deviation (seconds)	2242	1278	2113
percentage within 1σ	98.1%	91.6%	93.9%
percentage within 2σ	98.7%	95.4%	96.2%
percentage within 3σ	99.4%	96.2%	98.5%

Table 4: MILP statistics, with control scenario (a).

From the statistics, we can see that: (i) MILP solutions require longer CPU time than LP solutions. (ii) Both LP and MILP are appropriate for real-time optimization: with the current network structure, the objective function converges to the optimum within a few minutes (compared to a two hour time window). (iii) The CPU time increases when the network structure becomes more complex, i.e. when there are more sectors and/or more controlled sectors in the network.

A visualization of the control strategies resulting from the computations is available for download at [18].

VI. Conclusion

A new Eulerian model of airspace was derived and applied to high altitude traffic for a full Air Traffic Control center of the National Airspace System. The Eulerian 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 ASDI data for the Oakland Center and applied to two hour Traffic Flow Management problems. The problem of controlling sector aircraft count is posed as an Integer Program in which the dynamical system appears in the constraints. To improve computational time of the solution, the Integer Program was relaxed to a Linear Program, solved for instances with more than one million variables. The computational results showed a high proportion of integer solutions of the LP, and computational time satisfactory for two hour Traffic Flow Management problems relevant for strategic Air Traffic Control.

[2]The simulations are done on a 1.4GHz CPU, 1GB RAM PC running Linux.

Acknowledgments

We are grateful to Dr. Banavar Sridhar for initiating this project, and for his guidance through this work. We are thankful to Dr. Kapil Sheth and Dr. Shon Grabbe for fruitful conversations regarding ASDI data and FACET. We want to acknowledge Larry Hogle for setting up the project through UCSC and UARC, and Dr. George Meyer for his ongoing support.

References

[1] S. Devasia, M. Heymann, and G. Meyer, "Automation procedures for air traffic management: A token-based approach," in *Proceedings of the American Control Conference*, Anchorage, AK, May 2002, pp. 736–741.

[2] 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*, vol. 27, no. 5, pp. 737–744, 2004.

[3] 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*, vol. 229, no. 1178, pp. 317–345, 1956.

[4] P. I. Richards, "Shock waves on the highway," *Operations Research*, vol. 4, no. 1, pp. 42–51, 1956.

[5] C. Daganzo, "The cell transmission model: a dynamic representation of highway traffic consistent with the hydrodynamic theory," *Transportation Research*, vol. 28B, no. 4, pp. 269–287, 1994.

[6] ——, "The cell transmission model, part II: network traffic," *Transportation Research*, vol. 29B, no. 2, pp. 79–93, 1995.

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

[8] 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.

[9] A. M. Bayen, R. L. Raffard, and C. J. Tomlin, "Eulerian network model of air traffic flow in congested areas," in *Proceedings of the American Control Conference*, Boston, June 2004, pp. 5520–5526.

[10] ——, "Adjoint-based constrained control of Eulerian transportation networks: application to Air Traffic Control," in *Proceedings of the American Control Conference*, Boston, June 2004, pp. 5539–5545.

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

[12] F. Borelli, Ed., *Constrained Optimal Control of Linear and Hybrid Systems*, ser. Lecture Notes in Control and Information Sciences. New York, NY: Springer Verlag, 2003, vol. 290.

[13] M. Jordan, *An introduction to probabilistic graphical models*, Stat241A Class Reader, Univ. of California, Berkeley, book in preparation.

[14] R. Vidal, *Generalized principal component analysis (GPCA): an algebraic geometric approach to subspace clustering and motion segmentation*. Univ. of California, Berkeley: PhD thesis, August 2003.

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

[16] C. A. Robelin, D. Sun, G. Wu, and A. M. Bayen, "Strategic traffic flow models based on data-mining and system-identification techniques," NASA Technical Memorandum, in preparation, 2006.

[17] 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*, vol. 6, pp. 281–300, 2001.

[18] `http://www.ce.berkeley.edu/~bayen/acc06/`, at the time of submission.

[19] R. Fourer, D. M. Gay, and B. W. Kernighan, *AMPL: A Modeling Language For Mathematical Programming*. Duxbury Press/Brooks/Cole Publishing Company, 2002.