Feedback Control of the National Airspace System

Feedback Control of the National Airspace System

Jerome Le Ny*

University of Pennsylvania, Philadelphia, PA 19104, USA

Hamsa Balakrishnan†

Massachusetts Institute of Technology, Cambridge, MA 02139, USA

This paper proposes a general modeling framework adapted to the feedback control of traffic flows in Eulerian models of the National Airspace System (NAS). It is shown that the problems of scheduling and routing aircraft flows in the NAS can be posed as the control of a network of queues with load-dependent service rates. We can then focus on developing techniques to ensure that the aircraft queues in each airspace sector, which are an indicator of the air traffic controller workloads, are kept small. This paper uses the proposed framework to develop control laws that help prepare the NAS for fast recovery from a weather event, given a probabilistic forecast of capacities. In particular, the model includes the management of airport arrivals and departures subject to runway capacity constraints, which are highly sensitive to weather disruptions.

I. Introduction

The frequent occurrence of air traffic delays in the National Airspace System (NAS), along with the projected increase in demand, motivate the scheduling of flight operations to better utilize available system resources. The process of planning operations in order to balance the available capacity and the demand for resources is known as Traffic Flow Management (TFM). This task is currently conducted manually by air traffic controllers (ATC), and contributes significantly to their workload. In order to meet the increasing traffic demand, there is a desire to introduce a greater level of automation and decision support for air traffic management.

Research on the TFM problem has traditionally focused on developing open-loop policies for scheduling aircraft operations. Due to the typical travel times of cross-country flights, open-loop traffic flow management policies need

*Postdoctoral Researcher, Department of Electrical Engineering. Levine Hall L465, 3330 Walnut Street, Philadelphia, PA 19104. AIAA Member.

†Assistant Professor, Department of Aeronautics and Astronautics. 77 Massachusetts Avenue, Room 33-328, Cambridge, MA 02139. AIAA Member.

to be determined 5-6 hours ahead of the time of operations. Such policies prescribe the position of each aircraft in the system at each instant, and are obtained by solving large-scale integer programs.[6,44] This approach is difficult to scale to the scheduling of approximately 40,000 flights a day, and typically does not address the many sources of uncertainty present in the system. Weather, in particular, is a major source of disruption that requires constant adjustment of the schedules. For instance, 66% of all NAS delays in 2009 were attributed to weather.[9] Moreover, open-loop traffic flow management algorithms require precise weather forecasts several hours ahead of time, which are arguably beyond the limits of even state-of-the-art weather forecasting tools.[27]

The disturbance attenuation properties of feedback control make closed-loop control policies for the NAS very attractive. Attempts have been made to introduce some feedback in the decision algorithms while still trying to optimize each aircraft trajectory.[32,46] More recently, researchers have started developing new models that are more tractable for the purpose of control, which only record aircraft counts in specific control volumes of airspace rather than follow individual aircraft. These aggregate flow models, called *Eulerian models*, are gaining popularity.[24,25,36,40,42] They have been shown to have reasonable predictive capabilities,[36,40,42] but in general these models are less precise than trajectory-based simulation tools such as FACET[7] for high-fidelity system simulation. On the other hand, their advantage is that they are sufficiently tractable and flexible for the development of flow control policies, and can provide insight into system behavior. Importantly, they are compatible with today's traffic flow management system, which regulates traffic flow rates rather than planning for individual aircraft.[40] Airport operations in the presence of adverse weather are currently planned using the Collaborative Decision Making (CDM) paradigm, under which airlines are allocated landing slots for their flights, given the planned arrival rates at the airport.[3] Eulerian models that determine arrival rates at airports therefore can be used within the CDM framework.

Feedback control schemes using Eulerian models have been previously proposed, both in the context of centralized traffic flow management,[25,29] and in a decentralized setting for networks with a single origin and destination.[1] In this paper, we present a new Eulerian model for TFM in the spirit of the two-dimensional Eulerian model of Menon et al.[25] However, in contrast to most prior work, our model can be used to control all resources of the NAS, rather than focusing on high-altitude traffic.[24,25,42] The inclusion of airports is particularly important because they are typically the bottlenecks of the system. Moreover, as shown in Section II.D, our general model captures as special cases other recent Eulerian models such as the CTM(L) model,[42] while offering additional modeling flexibility.

Eulerian models suggest strong parallels between approaches to air traffic flow management and the control of stochastic networks. A survey of control approaches for other complex networks, such as semiconductor manufacturing systems or the Internet,[26] shows that discrete formulations (typically based on deterministic integer programs or stochastic controlled Markov chains) have been considered intractable and too detailed for the purpose of controlling

realistic networks. As a first approximation, the discrete effects are usually neglected and continuous traffic flows are considered instead, much like Eulerian models of the NAS. For stochastic networks, these continuous traffic flow models used for control purposes are also called fluid models. However, unlike some Eulerian models of the NAS that involve Partial Differential Equations,[42] fluid models for stochastic networks yield Ordinary Differential Equations (ODEs), thereby simplifying the analysis and control synthesis tasks significantly. Such fluid models attempt to capture the average behavior of the system, and are usually sufficiently tractable to design a rudimentary control policy. Unmodeled components and variability in the dynamics are accounted for by appropriately modifying this basic policy. Our model is adaptable to the use of standard network control tools, and in particular we present the MaxWeight policy[43] that results in *distributed* feedback control laws for traffic flow planning at different facilities. This policy reflects the control structure in the current system,[20,33] in which facility (airport or sector)-level traffic planning is done locally through coordination with neighboring facilities. Note that prior queuing network models of the NAS have generally been developed with a view towards performance prediction and analysis, and not control,[10,22,23,36] which is our objective in this work.

At the TFM level, our goal is to provide high-level directives to air traffic controllers (ATCs) prescribing the desired flow rates of aircraft traveling through constrained resources in the NAS. These resources can be runways, metering points at certain airspace fixes, sectors, or flow constrained areas during Airspace Flow Programs (AFPs).[35] ATCs can implement these directives by issuing orders at the path planning level, such as aircraft speed changes, vector for spacing or holding patterns, which modify the time that an aircraft takes to travel between control boundaries. The specific choices adopted by the ATC at the tactical level depend on the spatial configuration of aircraft at the time of operations, and are not specified at the TFM level. As a consequence of this hierarchical decomposition, a TFM directive may be modified by the ATC during a given period, for example, due to path-planning or separation constraints. The deviation of the ATC actions with respect to the TFM directives are treated as disturbances at the TFM level, and are accounted for by the feedback form of the traffic flow control policies proposed. The development of models for closed-loop TFM should balance precision, in order to minimize these lower level disturbances, with tractability for real-time computations. As demonstrated in this paper, our model can also be used to prepare the system for disturbances of larger magnitude, for example due to weather events, by integrating probabilistic weather forecasts. This is arguably the most desirable feature of TFM procedures, since local weather events can be greatly amplified by network effects in the absence of proper congestion control.

The rest of the paper is organized as follows. Section II describes the general Eulerian model that we use for TFM. We show how various capacity-limited NAS resources, including airports, can be easily modeled in our framework. We also discuss how our model generalizes some previously proposed Eulerian models. Section III describes some

natural control policies for this system, namely the First Come, First Served (FCFS) policy, the MaxWeight policy (a distributed network control policy), and a strategy based on Model Predictive Control (MPC). In Section IV, the model is modified to accommodate probabilistic capacity forecasts. The MPC controller can then be used to mitigate the impact of weather disruptions, given such probabilistic forecasts. In Section V we present some simulation results, including a small problem illustrating network congestion effects under the various policies due to sector capacity constraints, and a larger, more realistic TFM scenario over part of the Western United States. Finally, Section VI summarizes our approach to the TFM problem and describes some directions for future work.

II. Eulerian Model of the NAS for Network Control

In this section, we propose an Eulerian Model of the NAS that lends itself to network control approaches. We also show that our model generalizes other existing Eulerian models of the air traffic system.[42] Note that this model is not necessary to implement our simplest control policies in section III, namely the FCFS policy and the MaxWeight policy, which are model-free. Only the MPC policy requires a model of the NAS dynamics for its implementation. However, this model is used for all three policies to evaluate their expected performance in the simulated scenarios of section V.

To construct the network model that we use for TFM, we start by deciding the points or lines through which traffic flow rates need to be determined. These boundaries, henceforth called *control boundaries*, can consist of sector boundaries, runways, airspace fixes, intersections of major jet routes, or other metering points. Adding more control boundaries provides more decision support, but decreases the flexibility to adapt (at the tactical level) to factors not precisely modeled at the TFM level. Each control boundary has an associated traffic flow direction (therefore physical boundaries such as those of sectors correspond to two control boundaries, one for each flow direction).

In developing Eulerian models, we are interested in controlling the aircraft counts in certain *control volumes* rather than individual aircraft trajectories. A control volume is delimited by an input and output control boundary, that is, all the traffic associated with it enters through the same control boundary and exits through the same one. Control volumes can overlap, for example, due to intersecting traffic flows. Within a control volume, we have one or more queues, also called buffers. Using several queues in the same control volume allows us to separate the traffic based on distinct characteristics, such as destinations. Figure 1 shows an example of an Eulerian model with five control volumes and two distinct flows.

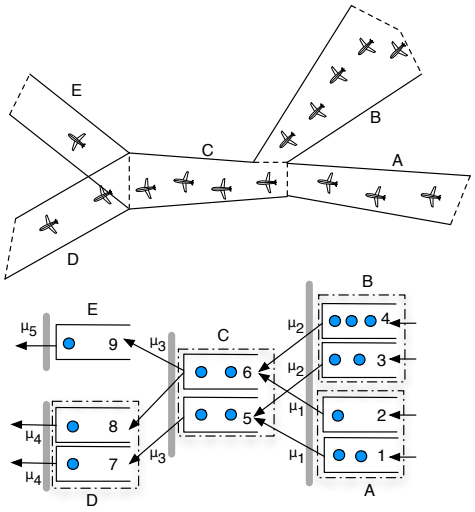


Figure 1. Five control volumes and their corresponding abstract buffer model. The situation depicted here corresponds to two routes merging and then diverging. Each control volume except E supports two flows with distinct characteristics (e.g., different destinations, as described Section II.B.1), and each flow is associated with one buffer. Buffers 8 and 9 have the same flow type (destination), which allows us to include routing decisions at the output boundary of volume C. Each buffer has an associated maximum throughput function, as discussed in Section II.A. In this case, the buffers associated with the same control volume have the same maximum throughput function. Thick gray lines represent additional resource constraints coupling the control variables of different buffers, as described in Sections II.B.1 and II.B.2.

II.A. Maximum Throughput of a Single Buffer

We consider discrete-time models, and choose a suitable time-discretization T to model the system dynamics. This time period also determines the frequency of updates to flow rate directives. The number of aircraft in buffer i at time kT, $k \in \mathbb{N}$, is denoted $Q_i(k)$. Every buffer is associated with a control volume (see Fig. 1), and hence to an input and an output boundary. The *maximum* rate at which traffic can flow out of a buffer during any time period depends on the number of aircraft that it contains. The aircraft count associated with buffer i follows the dynamics

$$Q_i(k+1) = Q_i(k) + A_i(k) + \sum_{j \in \mathcal{I}_i} U_{ji}(k) - \sum_{l \in \mathcal{O}_i} U_{il}(k), \tag{1}$$

$$0 \leq U_i(k) := \sum_{l \in \mathcal{O}_i} U_{il}(k) \leq D_i(k), \ \forall k \geq 0. \tag{2}$$

Here, $A_i(k)$ is the number of *external* arrivals in buffer i during the the $(k+1)^{\text{th}}$ period (that is, time interval $[kT, (k+1)T)$), originating from unmodeled parts of the system (e.g., pushbacks from airport gates, flights entering the modeled airspace, etc.). $U_i(k)$ is the number of departures from buffer i during the same period, and is controlled by the ATC. It is the sum of components $U_{il}(k)$, which represents the number of aircraft transiting from buffer i to buffer l, allowing for routing decisions at the output boundary of buffer i. Such tactical routing decisions can help accommodate dynamically changing conditions in the network, such as the impact of weather on capacities.[29] We

denote by $\mathcal{I}_i$ the set of buffers sending aircraft to buffer i, and $\mathcal{O}_i$ is the set of buffers to which buffer i sends aircraft. For example, on Fig. 1 we have $\mathcal{I}_6 = \{2, 4\}$ and $\mathcal{O}_6 = \{8, 9\}$. In this section, we focus on the quantity $D_i(k)$, which is the *maximum* possible number of departures from buffer i during the $(k+1)^{\text{th}}$ period. Note that there is an additional nonnegativity constraint, $Q_i(k) \geq 0$, which will generally be automatically satisfied by imposing the condition $D_i(k) \leq Q_i(k) + A_i(k)$. We assume that $U_i(k)$ can depend on $Q_i(k)$ as well as $A_i(k)$ and $D_i(k)$, therefore these quantities must be determined prior to determining $U_i(k)$. Evaluating $D_i(k)$ is a simple trajectory prediction problem for the typical values of T used in flow control (say 1-15 min). These values $D_i(k)$ can be determined at the tactical level by the ATC and communicated to the TFM level at each period.

In general, the travel times of aircraft through a control volume vary due to differences in speeds, trajectories and environmental factors such as wind speed. Since it is not tractable to keep track of all these variations exactly, we treat them as disturbances on a nominal aggregate model. At the TFM level, we assume a stochastic model of the number of arrivals $\{A_i(k)\}_{k\in\mathbb{N}}$ and maximum number of departures $\{D_i(k)\}_{k\in\mathbb{N}}$ in Equations (1) and (2). Moreover, we assume that for a given load level $Q_i(k) = Q_i$, the variables $D_i(k)$ have the same conditional expectation denoted $\mu_i(Q_i) = \mathbb{E}[D_i(k)|Q_i(k) = Q_i]$. We call the function μ_i the *maximum throughput function* for buffer i. Usually, the same function μ_i can be used for all buffers in the same control volume, unless traffic flows are differentiated based on characteristics such as aggregate trajectories or velocities. In order for this to be a reasonable model of the maximum number of aircraft departures in a period, the control volume should be large enough relative to the sampling period that aircraft cannot typically enter and leave the volume during the same period. A significant number of such aircraft would require modifying the model such that the function μ depends on $A(k)$.

Intuitively, μ_i should increase with Q_i, since an increase in traffic in a volume reduces the time between departures from the volume. However, μ_i remains bounded due to the minimum required separation distance between aircraft, which limits the rate at which aircraft can cross the exit boundary of the control volume. In general, we expect μ_i to be a concave saturating function, as depicted on Figure 2. The exact values of the function depend on the geometry of the control volume and the typical aircraft trajectories between its boundaries. Note that $\mu(0) = 0$, and $\mu(1)$ is approximately inversely proportional to the typical minimum travel time of an aircraft through the region (measured in increments of length T). For the simulation results shown on Figure 2, we assume that aircraft can travel at up to 500 knots, that the control volume length is 100 nm with a simple narrow linear geometry in which all aircraft strictly follow each other, and use ATC directives asking that aircraft entering the volume set their velocity to the maximum possible while respecting the separation constraint with the previous aircraft. The sampling period T is 10 min. Note that for the purpose of evaluating μ_i, we can consider the situation where the control volume contains only buffer i. The interactions among flows of different buffers within the same control volume are modeled as additional constraints

on the control variables U_i, as discussed in Section II.B.1.

Figure 2 also assumes that successive aircraft crossing the exit boundary of this control volume are separated by at least 2 min. This results in the saturation of the curve at 5 departures per time period. For control volumes that are not subject to such explicit metering constraints, the saturation phenomenon still persists due to the mandatory minimum separation between aircraft (currently 5 nm in enroute airspace). In our example, this would result in a curve saturating at a value of at most 16 aircraft per period instead of 5. Finally, we note that if the length of the time period (T) is changed from T_1 to T_2, the resulting curve $\mu_i(Q_i)$ can be obtained by scaling the curve for T_1 by T_2/T_1.

In this paper, we assume that quantitative models of the maximum throughput curves of the buffers are available, and we leave the discussion of this system identification step largely to future work. In the simulations of section V, we assume simple throughput curves similar to the one show on Fig. 2, which is well adapted however to many regions of airspace with well-defined jet routes. As pointed out below in subsection II.B.4, some previous work describes how to fit these curves using historical data when the buffers model surface operations.[34,38] For airborne traffic, more ATC actions are available than for ground traffic, and it is unlikely that historical data alone will be sufficient to reconstruct the complete throughput curves. The quantitative modeling step can then be done for each control volume first using trajectory based simulation tools, including a model of the actions available to the ATC, as done for the curve of Fig. 2. Much more extensive work is done in chapter 3 of Bayen's thesis,[4] where aircraft trajectories and available ATC actions in a given region are modeled using hybrid automata. The author develops a simulator subsequently used to predict flight times, but this approach could be used to predict the throughput curves as well. This work is validated against historical data, which shows good agreement of the simulation results with observations in the real system. Such a combination of simulation and validation based on empirical data can be used more generally to fit the throughput curves quantitatively.

II.B. Additional Resource Constraints

We construct a network model of the NAS using as building blocks control volumes containing buffers that follow the dynamics of Equations (1) and (2). Traffic flows in different buffers compete for limited airspace resources, resulting in additional linear constraints on the control variables, which are described in the following paragraphs.

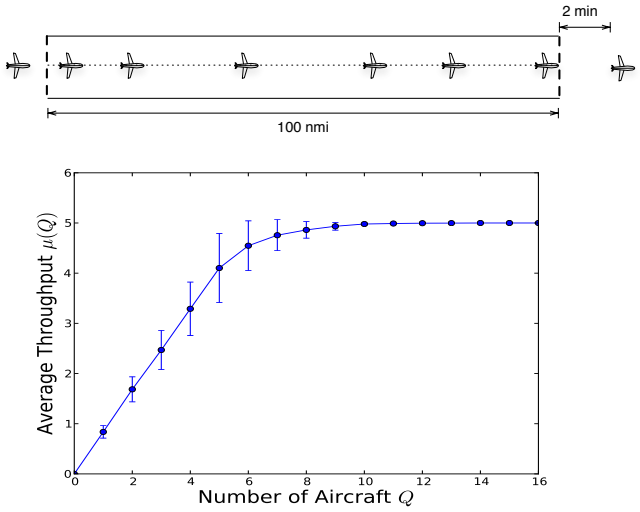


Figure 2. Maximum throughput for the exit boundary of a simple linear control volume. The empirical curve of the mean number of departures per period was obtained via discrete-event simulation, with random arrivals. The error bars show the empirical variance of the number of departures per period. In this example, the exit boundary includes an explicit metering constraint, specifying a mandatory minimum separation of 2 min between successive aircraft.

II.B.1. Shared Buffers within a Control Volume

Given a control volume containing m buffers with (output) controls $U_1, \ldots U_m$ (see the definition in (2)), the control vector for the exit boundary of the volume is denoted by

$$U(k) = [U_1(k), \ldots, U_m(k)]^T.$$

If several distinct flows carry a significant number of aircraft, the bounds $D_i(k)$ in Equation (2) can be large for several values of the index i and the exit boundary of the volume might not be able to accommodate $\sum_{i=1}^{m} D_i(k)$ departures in a single period. It may therefore be necessary to give priority to certain flows over others, i.e., to *schedule* the flows at the control boundaries. We add scheduling constraints of the form

$$c^T(k)U(k) = \sum_{i=1}^{m} c_i(k)U_i(k) \leq r(k), \quad \forall k \geq 0. \tag{3}$$

In general we take $c_i(k) = 1$ for all i, and then $r(k)$ is simply the maximum number of aircraft that can cross the boundary at period k. However, we may prioritize certain flows by varying c_i, and adjust r appropriately.

II.B.2. Intersecting and Merging Flows

Consider the scenario depicted in Figure 1. The input boundary of sector C coincides with the output boundaries of sectors A and B. Let $U^{(1)}$ and $U^{(2)}$ denote the control vectors associated with sectors A and B respectively, which

support two flows each. At the merge point, suppose we cannot accommodate the sum of the maximum flow rates of A and B, and increasing the flow rate out of one volume requires reducing the flow rate out of the other. This aspect can be incorporated by imposing linear constraints on the control vectors of the form

$$c_1^T U^{(1)}(k) + c_2^T U^{(2)}(k) \leq r, \tag{4}$$

for some vectors c_1, c_2, and some scalar r. Here again c_1, c_2 can be all-one vectors, in which case r represents the maximum number of aircraft that can enter volume C per period. Note that constraint (4) is not active if the volumes are only lightly loaded and the resulting bounds $D_i(k)$ in Equation (2) are small. The parameters in (4) could also be time varying.

Intersections of major jet routes can be handled similarly, and competition between flows for passage through limited airspace resources can be modeled by additional linear constraints of the form

$$C(k)U(k) \leq R(k), \tag{5}$$

where $C(k)$ is a matrix, $U(k)$ is the vector of all control variables for the problem, and $R(k)$ is a vector. Each limited resource contributes one or more constraints to Equations (5).

II.B.3. Sector Load Capacities

We can add bounds on the vectors $Q(k)$ to impose limits on the sector capacities. In general, these constraints take the linear form

$$MQ(k) \leq S(k), \tag{6}$$

where M is a matrix and S a vector, which can again be time varying.

II.B.4. Airport Resources

An airport is modeled using arrival queues (aircraft waiting to land, which are associated with air traffic flows in the vicinity of the airport), and one or more departure queues (aircraft on the ground waiting to take-off). All queues have dynamics of the form given by Equations (1) and (2). Since arrivals and departures at an airport share ground resources, the arrival and departure control vectors are also subject to resource constraints. If U_d is the control vector for the departure queues and U_a is the control vector for the arrival queues, the global vector $U = [U_a^T, U_d^T]^T$ is again subject to linear constraints of the form (5), as discussed by Gilbo.[17,18] These constraints depend on runway

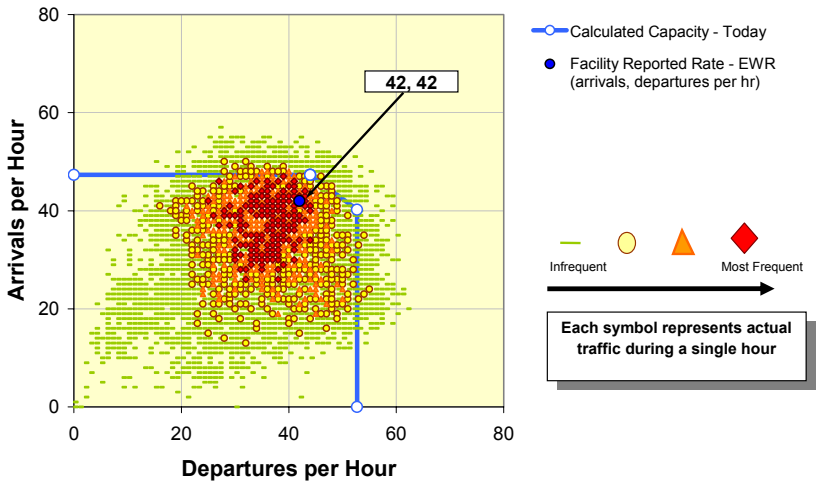


Figure 3. Capacity envelope observed at Newark airport, under optimal conditions.[45] Any pair of arrival/departure allocation rates within the polytope is achievable. This polytope corresponds to constraints of the form (5), and depends on the airport runway configuration.[17]

configurations and can be determined empirically or analytically.[17,45] An example is shown in Figure 3. These linear constraints (5) are not necessarily active in low traffic conditions due to the potentially more restrictive conditions (2). For the maximum departure throughput μ_d, we can consider a model incorporating aircraft from the time they pushback from the gate.[38,39] The airport transit zone then constitutes the control volume. In this model, we have a nonlinear maximal throughput curve for departures that depends on the number of aircraft in transit between the gates and the runway, has the form shown in Figure 2, and can be fitted using historical data.[34,38,39] Additional queues can be used to model the possibility of reordering aircraft between pushback and takeoff time, for example at taxiway intersections.[2]

II.B.5. Building Larger Models

Using the basic building blocks presented in the previous sections, it is straightforward to construct larger networks adapted to the control of traffic flows in the NAS. We illustrate this process in Section V. Consider, for example, a model incorporating the scheduling and routing of aircraft between origin-destination (O-D) pairs along a set of possible preferred routes. Within each control volume, we separate flows into distinct buffers based on their O-D pair, indexed by m. A flight trajectory corresponds to a path through the set of control boundaries. We can index buffers by their associated input and output boundaries as well as O-D pair, in other words, $Q_{ij}^m(k)$ is the load at period k for

O-D pair m in the control volume going from boundary i to boundary j. The dynamics of this buffer are then given by

$$Q_{ij}^m(k+1) = Q_{ij}^m(k) + A_{ij}^m(k) + \sum_{s \in I(i,m)} U_{sij}^m(k) - \sum_{t \in O(j,m)} U_{ijt}^m(k), \tag{7}$$

$$\sum_{t \in O(j,m)} U_{ijt}^m(k) \leq D_{ij}^m(k),\ 0 \leq U_{ijt}^m(k),\ \forall t \in O(j,m),$$

where $I(i,m)$ and $O(j,m)$ are the set of all control boundaries preceding i, and the set of all control boundaries following j respectively, that are on an allowed route between the OD pair m. A_{ij}^m denotes the number of external arrivals of O-D pair m in the control volume going from boundary i to boundary j. The variable U_{ijt}^m corresponds to aircraft of O-D pair m routed through the control boundaries i, j and then through the control boundary t. We note that other choices of models are clearly possible;[21] for example, one can use the index m to differentiate buffers in control volumes based only on the destination airport rather than the OD pair.

II.C. General Discrete Model

In previous sections we have seen that, after discretization of the airspace into control volumes carrying flows separated based on characteristics relevant to the TFM problem, we obtain a network model with dynamics that can be written in matrix form as

$$Q(k+1) = Q(k) + B\,U(k) + A(k), \quad \forall k \geq 0 \tag{8}$$

$$MQ(k) \leq S(k), \quad \forall k \geq 0 \tag{9}$$

$$C_1 U(k) \leq D(k), \quad \forall k \geq 0 \tag{10}$$

$$C_2(k) U(k) \leq R(k), \quad \forall k \geq 0 \tag{11}$$

$$U(k) \geq 0, \quad \forall k \geq 0. \tag{12}$$

The matrix B consists of $+1$'s and -1's and is the incidence matrix[15] for a graph whose nodes are the buffers and whose edges connect successive buffers. In addition, there are nonnegativity constraints $Q(k) \geq 0$, which can usually be automatically enforced through the constraints (10) (see Subsection II.A). The difference between model (8) and queueing network models studied in the literature on communication or manufacturing networks[26] is the addition of the load dependent constraint (10), modeling the fact that the number of aircraft in a sector influences the maximum rate at which aircraft can leave the sector. However, we note that if all components of $Q(k)$ are large, then $D(k)$ tends to a constant vector. Therefore, for high loads and when considering stability issues, we expect the analysis to be close

to the one developed for standard queueing network models.

II.D. Comparison with Other Eulerian Traffic Flow Management Models

Several previous Eulerian models can be represented as special cases of our model (8), which is not surprising since the latter only depends on the flow-balance equation (1). Menon et al.[24] propose a model inspired by the the Lighthill-Whitham-Richards (LWR) partial differential equation (PDE), traditionally used to model road traffic, and its discretization known as the Cell Transmission Model[12,13] (CTM). The control strategies they propose rely on linear systems theory, and do not account for crucial state and control constraints present in the system. A refined model is presented in a later paper by the same authors,[25] and a model predictive control strategy is proposed to handle these constraints. Similar models are used by Sridhar et al.,[40] and by Sun and Bayen.[41] These models tend to emphasize particular choices (based on geometry, size, etc.) of control volumes, whereas our discussion in the previous sections left these choices largely open, depending on the particular region and problem under consideration (a modeling example is presented in Section V). More important, in our view, is the fact that the dynamics in all these cases can be represented by the flow-balance equation (8), with a proper choice of control variables.

For example, let us see how we can write the recently proposed CTM(L) model[41] as a special case of our model. Flights going from one boundary of a sector to another are aggregated, and the model represents these flights along the links of a graph. Only traffic above 24000ft is considered, and aircraft climbing to and descending from this altitude are represented as external arriving and departing flows.[41] All aircraft in a given link fly at an aggregate speed, obtained from historical ASDI/ETMS data. A link is divided into cells, where each cell corresponds to one minute of flight time. For example, if it takes 10 minutes for a flight to cross a link, then the link is divided into 10 cells. A link is thus a linear succession of cells. The state of each cell is the number of aircraft in that cell. If link i is decomposed into m_i cells, then we can write the dynamics of cell $p \in \{1, \ldots, m_i\}$ as

$$Q_p^i(t+1) = Q_p^i(t) + A_p^i(t) + U_{p-1,p}^i(t) - U_{p,p+1}^i(t), \tag{13}$$

where the superscript indicates the link and the subscript the cell number. The external arrivals $A_p^i(t)$ come from aircraft ascending to the altitude considered. Descending aircraft which exit the flow can be easily taken into account by adding to a term $-U_{p,exit}^i(t)$ to Equation (13). For the boundary cells $p = 1$ in (13), $p-1$ represents the last cell of the previous link traversed by the aircraft, and similarly for $p = m_i$, $p+1$ is the first cell of the next link. Sun and Bayen[41] choose to control the number of aircraft delayed (i.e., the number of aircraft remaining in the same cell for the next time step), hence their control variables are $\tilde{U}_{p-1,p}^i(t) = Q_{p-1}^i(t) - U_{p-1,p}^i(t)$ and $\tilde{U}_{p,p+1}^i(t) = Q_p^i(t) - U_{p,p+1}^i(t)$.

Equation (13) then yields

$$Q^i_p(t+1) = Q^i_{p-1}(t) - \tilde{U}^i_{p-1,p}(t) + \tilde{U}^i_{p,p+1}(t) + A^i_p(t). \tag{14}$$

These are, in fact, the exact dynamics presented by Sun and Bayen[41] (Section III.A.1) at the link level (see also Equation (9) in their paper). The saturation constraint (10) is omitted however, and replaced by the simpler linear constraint $U^i_{p,p+1}(t) \leq Q^i_p(t)$. This forces the authors to use very small control volumes (of the order of about 10 nautical miles) in order to obtain a reasonable model. With small control volumes, each cell contains very few aircraft, and hence operates in the linear part of the throughput curve. The drawbacks of such a model are the large number of control volumes, most of which remain empty most of the time, and the inflexibility introduced at the interface with the tactical level, since with such small volumes, the Eulerian model effectively attempts to control each aircraft individually, rather than leaving this task to the ATC.

To our knowledge, the saturation constraint (10) described in Figure 2 and in references[34,38,39] is in fact ignored in all prior Eulerian models of the NAS, yielding particularly unrealistic models in the case of high-density operations. Moreover, many Eulerian models consider a single type of flow, with the exception of the CTM(L) model,[41] which differentiates aircraft based on their OD pairs. Even in the CTM(L) model, the interactions between different flows, leading to the scheduling, routing, and other resource constraints (11), are not included. Note that the possibility of prioritizing certain flows over others is crucial for proper congestion control across a network in general.[26] This possibility is also easier to exploit in air traffic than in road traffic due to the prevailing control structure.

Other models proposed Ball et al.,[3] and Mukherjee and Hansen[29] are also Eulerian models, since they regulate aircraft counts rather than optimize individual aircraft trajectories. Our model is a generalization of the ones presented in these papers, which in particular consider a single destination airport, and focus on integer programming formulations that can efficiently solve TFM problems in this setting. In contrast, we treat the various quantities as continuous in our optimization scheme described in the section III.C, and use naive rounding techniques for those quantities that require an integer solution. The rationale for this is that the TFM model (8) is intended to be an approximation of the system dynamics, and the actual dynamics is expected to deviate from this model at the tactical level. Under such conditions, the actual impact of obtaining and implementing the optimal integer solution of the full, disaggregate TFM model is not easy to quantify.

III. Control Strategies

Having presented an Eulerian model of NAS operations, we now develop algorithms to determine appropriate decisions $U(k)$ for the control of the system (8)-(12). A number of control techniques for such network models have been investigated in the past decades, particularly in the context of communication and semiconductor manufacturing systems.[26] We note that the state and control constraints in particular, neglected in the original work of Menon et al.,[24] play a fundamental role in the analysis of a network's stability and performance. For example, simple scheduling problems typically correspond to a square invertible constant matrix B, which in turn implies the controllability of a version of the model that neglects the control constraints; yet in practice not all such models are stabilizable.

We describe below three natural control strategies, which have very different computation and implementation requirements, and can be used for scheduling and routing aircraft through the NAS. The first strategy is the well-known First Come, First Served (FCFS) policy, and is arguably close to the strategy currently adopted by air traffic controllers. The MaxWeight policy[43] is a distributed policy with very few implementation requirements, much as the FCFS policy. It can be used for routing, scheduling, and load balancing and tends to reduce overload in downstream sectors compared to the FCFS policy. Both these policies can be used under nominal conditions and do not depend on the model developed in section II. In contrast, the third approach is a Model Predictive Control (MPC) strategy that has greater implementation and information requirements, and is more suitable for use in uncertain and changing capacity scenarios such as bad weather days, as discussed in Section IV.

III.A. First Come, First Served Policy

In the FCFS policy, resource limited facilities such as runways and control boundaries are assigned to aircraft in the order of their arrival in the queues corresponding to that facility. Nearly all airspace and airport resources are currently allocated using the FCFS policy, since it is considered to be both fair, and simple to implement by air traffic controllers.[14,31] For example, in the absence of other control mechanisms such as ground delay programs and of finer modeling of the taxiway (see section II.B.4), aircraft at airport gates would be scheduled to use the runway in the order in which they are ready to pushback. In its pure form considered here, the policy does not implement rerouting, congestion control, or any other feedback mechanism, and it does not take into account the potentially different costs of waiting in different queues (for example, the cost difference between ground and airborne delays). It simply pushes the aircraft through the control boundaries as soon as possible. Specifically, to determine $U(k)$ we first set $U = 0$. Then we increase U by unit increments in the order in which aircraft arrived at the various facilities, until we cannot increase any coordinate of U because the constraints (9), (10), (11) would then be violated. This policy can be implemented

independently locally, since only the buffers that share a resource need to coordinate. Since air traffic control networks can be expected to the acyclic, this policy should be stable in general[37] and have good throughput properties under nominal conditions. However, in case of reduced landing capacity at an airport, the FCFS policy quickly saturates the terminal airspace with aircraft waiting for their turn to land, which in turn creates uncontrolled upstream congestion due to sector capacity constraints.

III.B. Distributed MaxWeight Policy

The celebrated MaxWeight or maximum back pressure policy for network control,[43] which was previously proposed by the authors for air traffic control,[21] can be obtained as follows. At period k, the network is in state $Q(k) = Q$, and we can also observe the current values of the parameters M, S, C_1, D, C_2, R, A in (8). Consider the quadratic Lyapunov function

$$V(Q) = \frac{1}{2} Q^T \Xi Q,$$

where Ξ is a diagonal weighting matrix with positive coefficients ξ_i, and define the Lyapunov drift

$$\mathcal{D}V(Q,U) = \mathbb{E}[V(Q(k+1)) - V(Q(k))|Q(k) = Q, U(k) = U].$$

We have immediately

$$\mathcal{D}V(Q,U) \leq Q^T \Xi (BU + \mathbb{E}[A(k)]) + \beta, \tag{15}$$

where

$$\beta = \frac{1}{2} \max_u \mathbb{E}[(BU(k) + A(k))^T \Xi (BU(k) + A(k))|U(k) = u].$$

Assuming the process A has a finite second moment, then β is a finite constant because the maximization in its definition is over the feasible control vectors, which is a bounded set.

The MaxWeight policy aims at minimizing the upper bound on the Lyapunov drift by minimizing the first term on the right hand side of (15). In other words at period k, in state $Q(k) = Q$ of the network we apply the control

$$U^{MW}(Q) \in \arg\min_{U \in \mathcal{U}} Q^T \Xi B U, \tag{16}$$

where $\mathcal{U} = \{U | C_1 U \leq D, C_2 U \leq R, U \geq 0, M(Q + BU + A) \leq S\}$. The choice of Ξ is left open here and allows the controller to prioritize certain flows over others.[21]

III.B.1. Implementation Requirements of MaxWeight

The interesting fact about the MaxWeight policy, as given by Equation (16), is that the minimization can be performed in a distributed way, due to the sparsity pattern of the matrices B, C_1, C_2. Recall that B is the incidence matrix of a directed graph whose nodes are the buffers and edges are present between successive buffers. Moreover, note that exactly one control variable is associated with each such edge, specifying the number of aircraft that are allowed to transit from one buffer to the next. Denoting the set of edges by $\mathcal{E}$, Equation (16) can be written explicitly as

$$U^{MW}(Q) \in \arg\min_{U \in \mathcal{U}} \sum_{e \in \mathcal{E}} U_e(\xi_{e^+} Q_{e^+} - \xi_{e^-} Q_{e^-}), \quad \text{or,}$$
$$U^{MW}(Q) \in \arg\max_{U \in \mathcal{U}} \sum_{e \in \mathcal{E}} U_e(\xi_{e^-} Q_{e^-} - \xi_{e^+} Q_{e^+}), \tag{17}$$

where e^- and e^+ denote the buffers at the beginning and end of edge e respectively. Let $\pi_e(Q) = \xi_{e^-} Q_{e^-} - \xi_{e^+} Q_{e^+}$ denote the (weighted) "pressure" on edge e when the network is in state Q. We see then from Equation (17) that each control variable U_e for which $\pi_e < 0$ should be set to 0.

The control variables for the edges for which $\pi_e \geq 0$ can be determined by solving local linear programs in which priority is given to edges that are subject to higher pressure. Firstly, for each buffer it is necessary to know the load in the downstream buffers in order to compute the pressure on each of the edges involving this buffer. Two different ATCs might be in charge of two successive buffers, but these controllers can easily communicate to determine the pressure on the edge between them, since successive buffers are in neighboring control volumes. Secondly, the nonzero entries in the rows of C_1 and C_2 and the capacity constraint $MQ(k+1) \leq S$ can couple control variables that involve shared resources. The linear program (17) can then be separated into independent linear programs, each involving a set of coupled control constraints. In the constraints described in Section II, the rows of C_1 couple only the routing decision variables for the same buffer, if two or more downstream buffers are available. The rows of C_2 are scheduling constraints for resources that can accommodate limited traffic flows. A control variable is only coupled with those other control variables with which it competes for resources. For example in Figure 1, the controllers in charge of sectors A, B and C need to communicate with each other to determine the pressures and their control variables, but the controllers of sectors A and B need not communicate with the controllers of sectors D and E in order to implement the MaxWeight policy.

Finally, in many cases, the local optimization problems do not require any computation or explicit optimization

procedure. Consider for example a pure scheduling constraint of the form

$$\sum_{i=1}^{N} U_{e_i} \leq R, \tag{18}$$

where the control variables U_{e_i} do not appear in any other constraints in (11) and are only subject to the throughput constraints $0 \leq U_{e_i} \leq D_{e_i}$ (i.e., no routing variables are involved). Moreover, assume for now that the current load allows the capacity constraints (9) of the sectors receiving traffic from the buffers involved in (18) to be met by any choice of control. The controllers must then optimize the corresponding part of the objective function $\sum_{i=1}^{N} \pi_{e_i} U_{e_i}$, subject to the constraint (18). Assume, for simplicity of notation, that $\pi_{e_1} \geq \pi_{e_2} \geq \pi_{e_N}$, and assume $\pi_{e_1} \geq 0$. Then the MaxWeight policy proceeds iteratively as follows: It first sets U_{e_1} to $\min\{D_{e_1}, R\}$. At step l, it has set the variables $U_{e_1}, \ldots, U_{e_{l-1}}$ with the highest pressure to their maximum value D_{e_i}, except for $U_{e_{l-1}}$ if constraint (18) became active. If this constraint is not yet active and if $\pi_{e_l} \geq 0$, it sets U_{e_l} to $\min\{D_{e_l}, R - \sum_{i=1}^{l-1} U_{e_i}\}$. If the volume capacity constraints (9) can become active, calculations might again be required for a rigorous optimization procedure. However, a good heuristic is still to increase the control values iteratively on the links until none can be increased without violating the constraints. For many situations of interest, the local decisions for the MaxWeight policy can be similarly obtained without explicitly solving a linear program.

Remark Instead of the implementation described in the previous paragraph, we can consider the following variation of MaxWeight for scheduling a utilization-limited resource, which might be preferable from the perspective of fairness between different flows. We increase the variables by *unit increments*, until none of them can be increased any further because some constraint would then be violated or all the pressures are negative. At each step, we increment the control variable e for which the pressure π_e is maximum, and update the state of the queues accordingly, by subtracting 1 from Q_{e^-} and adding 1 to Q_{e^+}. We then update the pressure values and repeat the process.

It is worth emphasizing again that the implementation of the MaxWeight policy requires very little information, similarly to the FCFS policy and in contrast to the more demanding MPC control law presented next, for example. In particular, we do not need to know the departure rates at airports, or the throughput functions μ. We have seen that decisions are based only on the local information regarding the loads of the queues. These minimal information requirements can be an advantage, providing an inexpensive yet efficient way to guide TFM decisions under nominal conditions. For complex scenarios involving significant weather perturbations, where non-local policies such as ground holding programs must be implemented based on geographically-distributed information, such a policy becomes less effective. Simulation results illustrating this point are presented in Section IV.

III.B.2. Stability of MaxWeight

Asymptotic stability considerations for the NAS do not appear to be relevant at first glance, since the system could always be drained of the remaining aircraft overnight, when no new departures are scheduled. However, for network models such as (8)-(12), asymptotic stability results provide some insight about the short term behavior of the system as well. For example, a single unstable queue (i.e., a queue for which the arrival rate is strictly larger than the service rate) simply builds up load linearly with time on average, and hence becomes quickly unmanageable in a system like the NAS, where buffers have limited capacity. The behavior for a network of queues is similar,[26] and in an unstable network some queues will quickly become unacceptably large. For this reason, it is useful to know if a TFM control policy can stabilize the NAS under nominal conditions.

Let us assume that the arrival vectors are independent and identically distributed, with $E[A(k)] = \alpha$, $\forall k$. Under the MaxWeight policy, the state vector $Q(k)$ then evolves as a Markov chain. The network is said to be *stable* if for each initial condition $Q(0) = x$, the average cost

$$\eta_x = \lim \sup_{n\to\infty} \frac{1}{n} \sum_{k=0}^{n-1} E[\|Q(k)\|_\infty], \tag{19}$$

is finite. In the absence of the capacity constraints (9) and with constant throughput functions μ rather than the nonlinear ones considered here, the MaxWeight policy is known to be stabilizing for any network that is stabilizable.[26] Define

$$\delta(Q) = \mathbb{E}\left[\min_{U\in\mathcal{U}} Q^T \Xi(BU + A(k)) \middle| Q(k) = Q\right],$$

which is the average minimum achievable value for first term on the right-hand side of (15) (note that the constraint set $\mathcal{U}$ is stochastic here). Now assume that there is a positive constant C and $\epsilon > 0$ such that

$$\text{for all } x \text{ with } \|Q\|_\infty > C, \text{ we have } \delta(Q) < -\epsilon\|Q\|_\infty. \tag{20}$$

For a constant throughput function μ, this condition is necessary for the network to be stabilizable,[26] and one can expect that this condition is also a necessary stabilizability condition for our problem with load-dependent throughput curves. Intuitively, this can be seen because $\{BU + A | U \in \mathcal{U}\}$ is the set of velocity directions in which the control can steer the network dynamics (8). For stability, at each value of the state Q there should be (at least on average) a velocity vector in this set steering the state toward a bounded rectangle around the origin. This is essentially what condition (20) says. Now this condition is also sufficient for stability. Indeed, it implies that under the MaxWeight

policy

$$\mathcal{D}V(Q) := \mathbb{E}[V(Q(k+1)) - V(Q(k))|Q(k) = Q] \leq -\epsilon\|Q\|_\infty + b, \ \ \forall \|Q\|_\infty > C, \tag{21}$$

which in turn implies the stability of the network as a consequence of the Foster-Lyapunov criterion (Theorem 8.0.3 in Meyn[26]). Note that in general, condition (20) is not easy to check, as it is non-trivial to verify if a network is stabilizable (for constant throughput functions, this can be done via linear programming[26]). In practice however, the need rarely arises to use it directly.

III.C. Model Predictive Control

Model Predictive control (MPC)[16] is a general tool that is well suited for the feedback control of constrained systems of the form (8)-(12). Menon et al. previously proposed to use MPC to control a deterministic Eulerian model of the NAS.[25] We also use MPC for the weather scenario considered in Section IV. To obtain a feedback control law via MPC for the general problem (8)-(12), we proceed as follows. We fix a time horizon length of $K \geq 0$. At period k_0, we can observe the state $Q(k_0)$, the number of arrivals $A(k_0)$, and maximum number of departures $D(k_0)$ for the period. We determine $U(k_0)$ by solving the following convex program with variables $\mathbf{Q} = \{Q_k\}_{1\leq k\leq K+1}, \mathbf{U} = \{U_k\}_{0\leq k\leq K}$

$$\min_{\mathbf{Q},\mathbf{U}} \quad f(\mathbf{Q},\mathbf{U}) \tag{22}$$

subject to (23)

$$Q_1 = Q(k_0) + BU_0 + A(k_0)$$

$$Q_{k+1} = Q_k + BU_k + \alpha(k_0 + k), \ \ 1 \leq k \leq K$$

$$MQ_k \leq S(k_0 + k), \ \ 1 \leq k \leq K+1 \tag{24}$$

$$C_1 U_0 \leq D(k_0) \tag{25}$$

$$C_1 U_k \leq \mu(Q_k), \ \ 1 \leq k \leq K \tag{26}$$

$$C_2(k_0 + k)U_k \leq R(k_0 + k), \ \ 0 \leq k \leq K, \tag{27}$$

$$U_k \geq 0, \ \ 0 \leq k \leq K,$$

where f is chosen to be a convex function of $\mathbf{Q}$, for example a linear objective

$$f(\mathbf{Q},\mathbf{U}) := \sum_{k=1}^{K} \gamma_k^T Q_k + \sum_{k=0}^{K} \tilde{\gamma}_k^T U_k + \gamma_{K+1}^T Q_{K+1}, \tag{28}$$

for some positive vectors $\{\gamma_k, \tilde{\gamma}_k\}_k$. We use the certainty-equivalence heuristic[5] which consists of replacing $A(k)$ and $D(k)$ of (8) and (10) by their average values $\alpha(k)$ and $\mu(Q_k)$ for $k > k_0$. The program (22) is convex if we assume that μ is concave, which is generally true, as seen, for example, in Figure 2. Finally, following a standard MPC approach we impose a high terminal cost, i.e. $\gamma_{K+1} >> \gamma_k$ for $k \leq K$ in our experiments.

Upon solving the optimization problem (22), we obtain a sequence of vectors $U_0, \ldots, U_K$, which are real-valued. We round-off the first vector U_0, and use it as a control directive $U(k_0) = U_0$ for the current period. We discard the other vectors $U_1, \ldots, U_K$. At the next period, we repeat the procedure to obtain a new control vector, after observing the new values of Q, A and D. The convex program (22) can be solved using efficient interior point methods for various choices of objective and throughput functions.[8] For example, we can consider the linear objective function (28), and an approximation of the maximal throughput functions of the piecewise linear form

$$\mu_i(Q_i) = \min_{1 \leq j \leq m_i} \{a_{ij}^T Q_i + b_{ij}, \mu_{i,sat}\}.$$

Then the constraint $c_i^T U \leq \mu_i(Q_i)$ can be rewritten as m_i affine constraints (see Figure 4 for the case $m_i = 1$).

$$\begin{aligned} & c_i^T U \leq a_{ij}^T Q_i + b_{ij}, \ j = 1, \ldots, m_i \\ & c_i^T U \leq \mu_{i,sat}. \end{aligned} \tag{29}$$

In this case, the constraints (26) are replaced by linear constraints and the program (22) becomes a linear program.

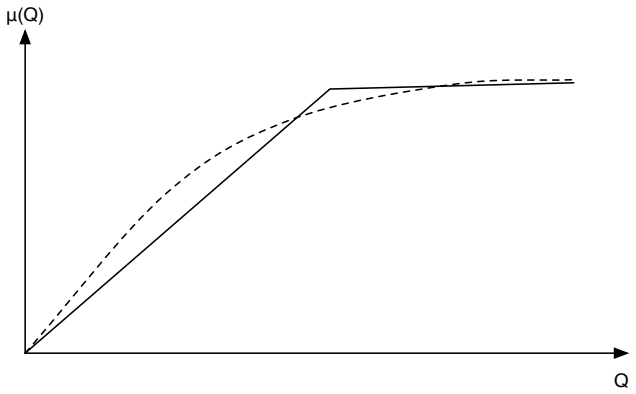


Figure 4. Simple piecewise-linear approximation of the throughput curves used in the MPC optimization step, see (29).

As mentioned in Section II.D, we use a naive rounding procedure to obtain an integer-valued control vector U from U_0 at each period. Namely, starting from $U = 0$, we increase the coordinates of the control vector by unit increments in the FCFS order, until we reach the rounded value of U_0 or until we cannot increase them because the constraints

(25), (27) for $k = 0$ or (24) for $k = 1$ would be violated. An alternative approach would be to solve the program (22)-(27) as an integer program, producing directly an integer solution for U_0. In view of the heuristic nature of the MPC approach, and because the TFM model is itself only an approximation of the real system dynamics, quantifying the performance impact of using a better rounding procedure is complicated and hard to justify. The feasibility of the MPC approach requires that the ATCs have enough time to implement the directives. Therefore, we need the computation of (22) to be finished in a time much shorter than the time period T, which is in general not possible using integer programming for realistic models.

IV. Weather Management

Our discussion so far assumed that parameters such as the average capacities at control boundaries and volumes were known. However, if a region experiences bad weather, its capacity to accommodate traffic flows can be greatly reduced and vary stochastically. Weather forecasts useful for detailed aircraft route planning are becoming increasingly available for TFM.[28] In this section, we consider the systematic integration of probabilistic weather forecasts within TFM decision-making in the NAS. Currently, some limited forms of flow management procedures take weather into account, such as Ground Delay Programs (GDPs) and Airspace Flow Programs (AFPs).[35] We would like to generalize and coordinate such programs in a more systematic way, while incorporating real-time information in a feedback loop. We extend the MPC approach of Section III.C to the situation where the capacity constraints evolve randomly in time according to a probabilistic forecast. Hence we assume for modeling and control that the weather forecast has been translated into a capacity forecast for control boundaries and control volumes. This subject is currently being actively investigated.[27,28,35]

IV.A. A Model Integrating Weather Uncertainty

We assume that the weather state over the region of interest can be classified in a finite number of states, following current practice for establishing airspace flow programs (AFPs)[35] and classifying the capacity region of airports.[45] We also assume that the weather state evolves as a Markov chain $\{w(k)\}_{k\geq 0}$ on a finite state-space W, with *time-varying* transition matrix $\mathbb{P}(w(k+1) = w' | w(k) = w) = [P(k)]_{ww'}$. The transition matrix is assumed known, provided by the from probabilistic forecast. Note that we can use such a model to represent weather scenarios of the form considered by Ball et al. or Terrab and Odoni,[3,44] for example. The weather state influences the parameters appearing in the constraints (9)-(11) of the model described in Section II.C, namely the capacity vectors S, R and the matrices C_1, C_2, M. The vector R describes the number of aircraft that control boundaries can accommodate, hence depends

on the weather state, and is decided by traffic management initiatives such as AFPs.[35] That the matrix C_2 changes with the weather state is evident from the fact that this matrix includes the models of the capacity envelopes of airports (see Section II.B.4). These envelopes vary in shape according to the weather state.[45] Hence we replace the constraints (11) by $C_2(w(k),k)U(k) \leq R(w(k),k)$. We can also include weather-related changes in the throughput function μ to obtain a function $\mu(Q,w)$ depending on the load and weather state.

IV.B. Certainty-Equivalent MPC

We can now develop a certainty-equivalent model predictive controller (CE-MPC) similar to the one described in Section III.C. Namely, at period k_0, we observe the weather state $w(k_0)$, and we replace the constraints (24)-(27) by

$$\mathbb{E}[M(w(k_0+k))|w(k_0)]Q_k \leq \mathbb{E}[S(w(k_0+k),k_0+k)|w(k_0)], \;\; 1 \leq k \leq K \tag{30}$$

$$C_1(w(k_0))U_0 \leq D(k_0)$$

$$\mathbb{E}[C_1(w(k_0+k))|w(k_0)]U_k \leq \mathbb{E}[\mu(Q_k,w(k_0+k))|w(k_0)], \;\; 1 \leq k \leq K \tag{31}$$

$$C_2(w(k_0))U_0 \leq R(w(k_0))$$

$$\mathbb{E}[C_2(w(k_0+k),k_0+k)|w(k_0)]U_k \leq \mathbb{E}[R(w(k_0+k),k_0+k)|w(k_0)], \;\; 1 \leq k \leq K, \tag{32}$$

where the expectations in the constraints (30), (31), (32) can be computed recursively for the next K stages at the cost of essentially K matrix multiplications of size $|\mathsf{W}| \times |\mathsf{W}|$. The optimization problem is still convex, and a linear program under the assumptions stated in Section III.C. Note that constraint (30) can lead to infeasibility issues in the optimization procedure, for example if the capacity of a volume drops significantly below the current value of the load because the weather state changes. One could also ask that the capacity constraints be met with high probability, e.g.

$$\mathbb{P}\Big[M(w(k))Q(k) \leq S(w(k),k)\Big|w(k_0)\Big] \geq 1-\epsilon,$$

for some specified ϵ. Such *chance constraints* can be approximated so that the MPC computations remain tractable, see e.g. Nemirovski and Shapiro.[30] In the simulation section, we assume that the constraints (30) on the state do not vary with the weather state, i.e., we use the constraint (24) instead of (30).

V. Simulation Results

V.A. Network Congestion due to Capacity Constraints

Our first simulated scenario illustrates the following well-known congestion phenomenon. Due to sector capacity constraints (see (9)), a local weather event can create upstream congestion and thereby impact a priori unrelated traffic flows. The situation is depicted on Fig. 5. Airport 1 sends traffic to airports 2 and 3. The sampling period is set to $T = 5$ min. We include the control coupling constraints

$$U_1(k) + U_4(k) + U_9(k) \leq R_1(k) \ \text{ (departure runway utilization at airport 1)}$$
$$U_2(k) + U_5(k) \leq R_2(k) \ \text{ (limited handoff rate } B \to C\text{)}$$
$$U_3(k) \leq R_3(k) \ \text{ (landing rate constraint at airport 2).}$$

We take $R_1(k)$ to be independent and identically distributed (iid) with mean $4T/3$ (i.e., on average one plane can leave every 45 s from airport 1), $R_2(k)$ iid with mean T (one plane can cross the boundary $B \to C$ every minute on average), and under nominal conditions $R_3(k)$ are iid with mean T (one plane can land at airport 2 every minute). The throughput curves are taken to be piecewise linear as on Fig. 4, with the distances shown on Fig. 5 converted to throughput curves assuming the simple geometry of Fig. 2 (the separation distance used here is 10 nmi). The external arrival rates at queues 4 and 1, i.e., the rate of pushback from the gates for the two different flows, are $0.30T$ and $0.25T$ respectively. Note that the flow from airport 1 to airport 3 can take a short or a long route, as decided by the routing control U_4 or U_9 at queue 4. Finally, we have capacity constraints in two regions. Region B can only contain 30 aircraft overall, and region D around airport 2 can only contain 11 aircraft. That is, we must enforce $Q_2(k) + Q_5(k) \leq 30$ and $Q_3(k) \leq 11$ for all $k \geq 0$. When a region reaches its capacity, it does not accept incoming traffic any more and the traffic is queued in the region upstream. Under these nominal conditions, the FCFS policy has an acceptable behavior and in particular stabilizes the system, see Fig. 5.

The capacity constraint in region B is particularly problematic however, since it couples the traffic flows to airports 2 and 3. Indeed, suppose that queue 3 incurs a sudden drop in throughput because region D around airport 2 is subject to a bad weather event reducing the landing capacity drastically. Namely, assume now that $R_3(k)$ are i.i.d. with mean $T/5$. Hence on average one aircraft can land at airport 2 every five minutes instead of five previously. In this case, it is not possible to stabilize the flow going from airport 1 to airport 3, and long lines necessarily form in queues $1, 2, 3$. By using the same FCFS policy however, it turns out that the flow from airport 1 to airport 3 becomes unstable as well, as illustrated on Fig. 6 (a). The reason is that as queue 3 becomes saturated, the length of queue 2 grows and does not

leave enough capacity resources to queue 5. As a result, the backlog at queue 4 grows dramatically even though this traffic is not directly headed to the region impacted by the weather event.

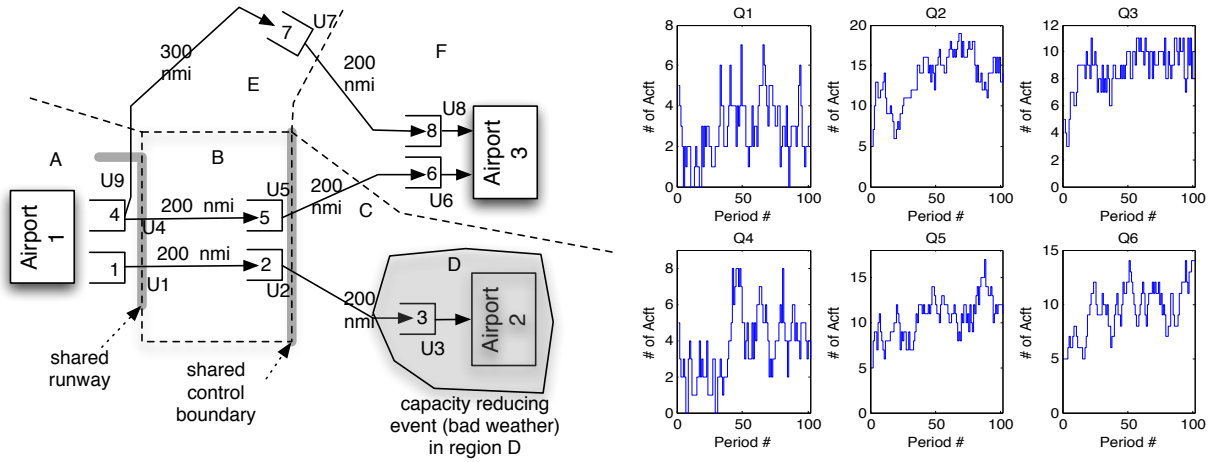
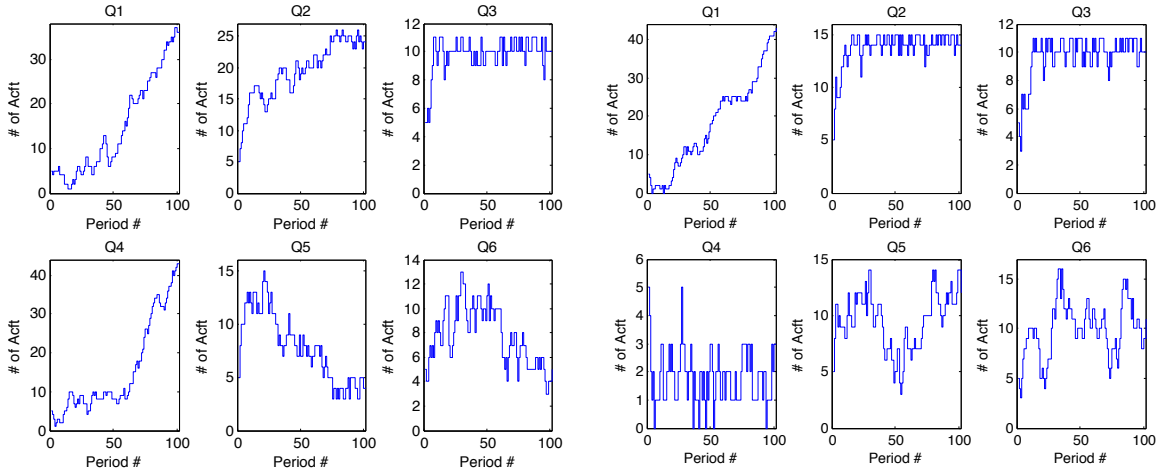


Figure 5. Small scenario with geographic capacity constraints and a local bad weather event in region D. In the absence of bad weather (nominal conditions), the FCFS policy stabilizes the system, as shown on the right of the figure (100 periods of length $T = 5$ min each shown). Queues $1-6$ start with 5 aircraft each.



(a) Sample path for the FCFS policy.

(b) Sample path for the FCFS policy with decoupled constraints in region B. Queue 4 now remains stable.

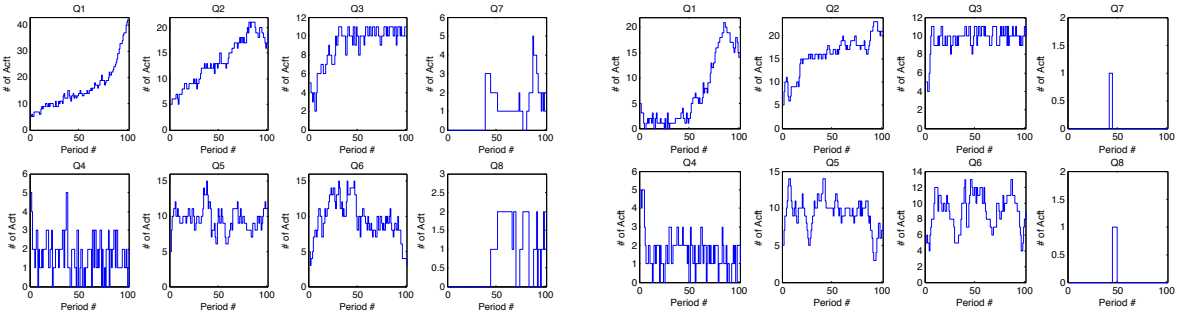
Figure 6. Same example as in Figure 5, but in the event of bad weather, showing the results of the FCFS and modified FCFS (decoupled constraints for region B) policies.

There are various solutions to this issue. Perhaps the most straightforward is to set separate capacity constraints in region B for the two flows. For example by replacing $Q_2(k) + Q_5(k) \leq 30$ with $Q_2(k) \leq 15$ and $Q_5(k) \leq 15$, we guarantee enough resources to queue 5 to ensure that the FCFS policy still stabilizes the flow from airport 1 to airport 3, see Fig. 6 (b). For this particular choice, we recover stability of the flow originating from 4. Problems with this approach for more complex scenarios include the difficulty of determining appropriate individual thresholds and

of setting many thresholds as the number of different flows in a region grows, the fact that it requires the knowledge of the various traffic flow rates, and its potential lack of flexibility and reactivity in the case of dynamically changing weather conditions.

The MaxWeight policy can also keep the flow through queues $4-5-6$ stable, even in the absence of decoupled capacity constraints, and is therefore perhaps more flexible in this case. In the case where no routing is allowed, maintaining stability of this flow requires giving a clear preference to queue 5 with respect to queue 2 however, and this is accomplished by setting the weight matrix Ξ such that for example $\xi_4 >> \xi_5 >> \xi_1$ and $\xi_1 = \xi_2$. With such a choice, any slight imbalance $Q_4 > Q_5$ gives priority to aircraft in queue 4 over aircraft in queue 1 to take off from airport 1. Similarly, if routing is allowed, one can specify a preference for the shortest route by the choice of weights. By changing these weights dynamically, one can give preference to certain flows in the system. There is however no quantitative way of choosing the weights in general. A sample path of the queue trajectories for the MaxWeight policy with routing allowed is shown on Fig. 7 (a).

Finally, we can implement the MPC controller of section III.C, again for the scenario with reduced throughput in region D. The cost is set to be linear as in (28), with $K = 15$, $\gamma_k = \mathbf{1}_8$ the eight-dimensional all-one vector, and $\gamma_{K+1} = 50 \times \mathbf{1}_8$. The advantage of this control law is that it leaves only few parameters to adjust, e.g. K and the magnitude of the final cost, and in this case also stabilizes the flow from airport 1 to 3, see Fig. 7(b). One can specify a cost measure in the optimization problem in terms of the vectors Q and U. In this case, our cost (excluding the final cost) is the expected total number of aircraft in the system over the optimizing horizon, hence we indirectly attempt to minimize the total time spent in the system by all aircraft, which is also an indirect measure of all delays. The potential drawbacks of this controller however is that its implementation requires a lot of information to be sent to a central computing location at the beginning of each period, namely the aircraft counts in each queue, the external arrival and throughput rates, the weather state, etc.



(a) Sample path for the MaxWeight policy with routing.

(b) Sample path for the MPC policy.

Figure 7. Same example as in Figure 5, but in the event of bad weather, showing the results of the MaxWeight and MPC policies.

To help evaluate the performance of the different policies, we compute over 100 sample simulations the cost

measured by the total cumulative number of aircraft in the system during 100 periods (each of length $T = 5$min). That is, the cost for a single run is $\sum_{k=0}^{100} \mathbf{1}_8^T Q(k)$. The performance histograms for FCFS, FCFS with decoupled constraints, MW with routing and MPC are shown on Fig. 8. Statistically, there is no advantage in using the MPC policy in this simple scenario over the FCFS policy with properly decoupled constraints for Q_5 and Q_2, except for the fact that the MPC approach is more systematic and avoids the problem of choosing bad individual thresholds, which could happen in more complex scenarios. Note that even though the FCFS policy performs poorly on average as discussed above, it has the best cost in a few simulation samples, where most likely the asymptotic instability of queue 4 did not start manifesting itself significantly over the first 100 periods.

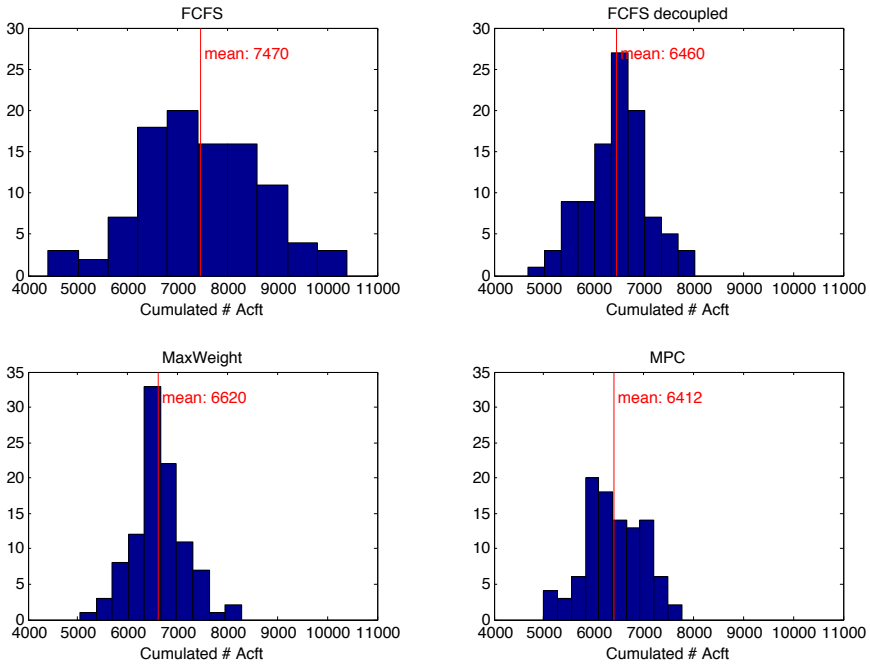


Figure 8. Performance histograms over 100 simulations, with bad weather over region D.

V.B. A Larger TFM Scenario

Finally, we illustrate the modeling of a larger TFM scenario over a portion of the western part of the United States and discuss the performance of the three control policies for this problem. The airspace modeled is primarily within the Oakland Air Route Traffic Control Center (ZOA), and we consider the airports at Los Angeles (LAX), San Francisco (SFO), Seattle (SEA), Portland (PDX), and Las Vegas (LAS). External traffic also enters the system, mainly via the major routes coming from the east towards SFO, LAX and SEA. After identifying the major routes supporting most of the traffic, we define control boundaries at which we would like to regulate traffic rates, and the corresponding control volumes. Separating traffic flows with different destinations within control volumes (see section II), we obtain

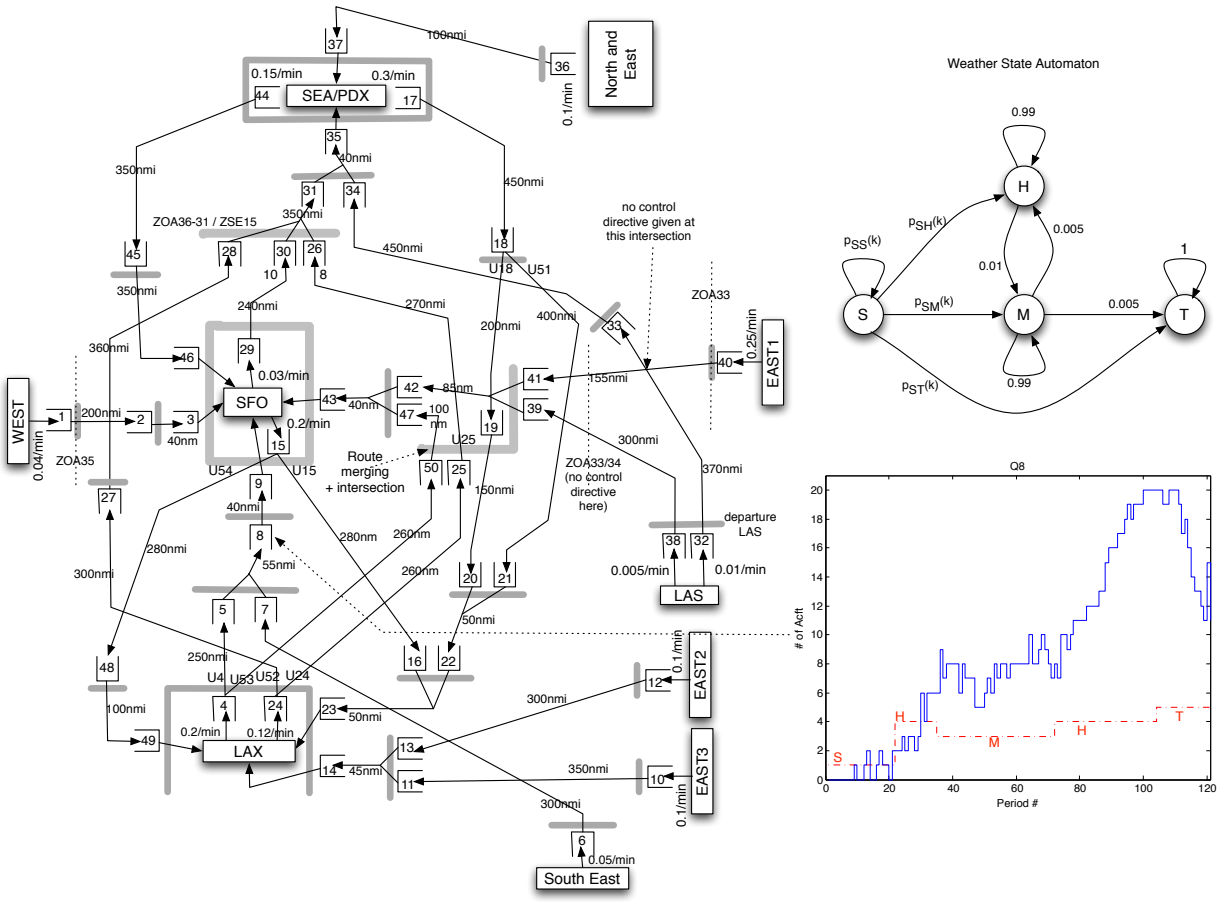


Figure 9. Network model for TFM in the simulated system. Thick gray lines correspond to control variables coupled via the linear constraints (10), (11). All queues start initially empty. We also show the automaton describing the probabilistic weather evolution as well as a sample trajectory of a queue in the system under the MaxWeight policy. The red dashed line shows the evolution of the weather state for that sample, namely $S - H - M - H - T$.

the queueing network shown on Fig. 9. It consists of 50 queues, with most of the control variables consisting of scheduling decisions, and with a few routing decisions available. The capacity envelopes for the airports are obtained from the 2004 Airport Capacity Benchmark Report.[45] The throughput functions for the control volumes are only approximately identified, based on the length of the control volumes, the required separation distance (5 nmi), and assuming simple linear traffic flows in the volumes, as on Fig. 2. In order to solve the MPC optimization problem as a linear program, these throughput curves are then approximated by piecewise linear functions consisting of just two pieces, as shown in Fig. 4. The departure rates at airports and other external arrival rates are shown on Fig. 9. Simulations of the scenario are run for eight hours, with sampling period $T = 4$ min, resulting in a discrete-time problem with 120 periods.

For this system, we simulate a bad weather event around San Francisco that changes the capacity envelope at SFO airport according to the data given in the 2004 Airport Capacity Benchmark Report.[45] The event also reduces the

capacity in certain other regions in a more limited fashion, in particular at the major traffic intersection East of SFO, and at LAX. We assume that after some initial period during which the system operates at optimal capacity (starting state S), the weather changes at some random time to one of 2 states: medium (M) or high (H) weather impact. In the H state, SFO allows no departures and accepts only 10 arrivals per hour; M corresponds to Marginal conditions at SFO in the capacity benchmark.[45] The Markov chain modeling the weather dynamics can jump between these 2 states for some time, but eventually reaches a final absorbing state (terminal state T), in which the capacity is again optimal. The structure of the probabilistic model is shown in Figure 9. The transition probabilities from the initial state used in the simulations are as follows. We set $P_{\mathsf{SS}}(k) = 1$ for $k \leq 13$ (i.e., no bad weather for the first 52 min). For $13 \leq k \leq 102$, we set $P_{\mathsf{SS}}(k) = 1 - p(k)$ with $p(k) := (e^{4(k/120)} - 1)/(e^4 - 1)$, $P_{\mathsf{SM}}(k) = 0.2\,p(k)$, $P_{\mathsf{SH}}(k) = 0.8\,p(k)$. Finally, $P_{\mathsf{ST}}(k) = 1$ for $k \geq 103$, a transition that is taken if the bad weather scenario is avoided. All queues except the ones receiving external arrivals (i.e., the queues $1, 4, 24, 6, 10, 12, 15, 17, 29, 32, 36, 40, 44$) have a bounded capacity of 20 aircraft. In addition, the complex intersection east of san francisco is also subject to a capacity bound

$$Q_{19}(k) + Q_{25}(k) + Q_{39}(k) + Q_{41}(k) + Q_{50}(k) \leq 20, \forall k \geq 0, \tag{33}$$

and the region around three of the SFO arrival queues is subject to the constraint

$$Q_3(k) + Q_9(k) + Q_{43}(k) \leq 15, \forall k \geq 0. \tag{34}$$

Table 1. Typical computational requirements per period to solve the linear program for the MPC heuristic. Computations were done using MATLAB and CVX[19] on a standard desktop with a $3.06GHz$ Intel Core 2 Duo processor and $4GB$ of RAM.

K	Nb variables	Nb constraints	Computation time (s)
8 (32 min)	2120	744	$\leq 1s$
23 (92 min)	6063	2227	$\leq 3s$
38 (152 min)	9995	3699	$\leq 5s$

We simulate the different policies presented above using the model of section II. FCFS does not include the possibility of routing, and only the shortest routes can be chosen by the aircraft. Routing is allowed in MaxWeight, which can take advantage of alternate paths to direct traffic out of saturated areas. Note that the computation times of the MaxWeight and FCFS policy at each period are negligible, and recall that these policies do not exploit the knowledge of the model parameters nor that of the transition probabilities in the weather automaton. We also use the CE-MPC controller in the linear programming form described in Section IV.B. The cost function we optimize in the MPC objective (28) consists of a time-invariant cost $\gamma_1 = \ldots = \gamma_K = \mathbf{1}_{50}$, $\gamma_{K+1} = 100\ \gamma_1, \tilde{\gamma}_k = 0$. At

each time period, the MPC algorithm involves solving a linear program with a few thousand variables and constraints, a computation that can be done in a few seconds, as summarized in Table 1. We solve the linear programs with Matlab and the optimization modeling package CVX,[19] without trying to improve the computational performance by employing a more sophisticated LP solver. Once the MPC control directive for the period is obtained, the rest of the period would be available for the ATC to implement it, which is compatible with the short computation time of the MPC heuristic. Setting the horizon K for MPC to $32, 60$ and 92 minutes in 3 different sets of experiments (hence $K = 8, 15$, and 23 time periods, respectively), it was found for this scenario that essentially no performance gain could be obtained by increasing the MPC horizon beyond 32 minutes. This suggests that at the levels of uncertainty considered in this scenario, open-loop planning for larger horizons based on expected average dynamics provides no useful additional information. It was observed however that increasing the horizon of the MPC controller tends to reduce the sensitivity of its performance to the choice of the performance metric of interest for the system.

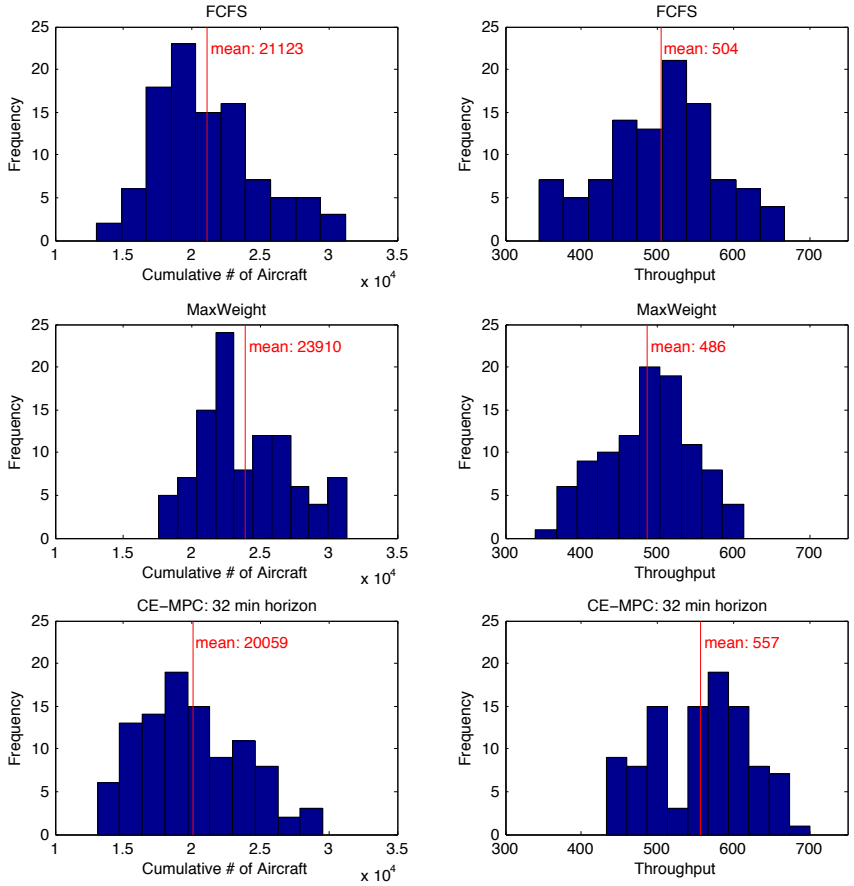


Figure 10. Histograms of the performance results for 100 simulations, for a scenario with probabilistic forecasts as described on Fig. 9. Each simulation corresponds to an eight hour interval. On the left, we show the total cumulative number of aircraft in the system during the eight hour duration of the simulation. In other words, an aircraft is counted once for every period in which it remains in the system. On the right, we show the throughput of the policies, that is, the total number of aircraft that landed during the simulation.

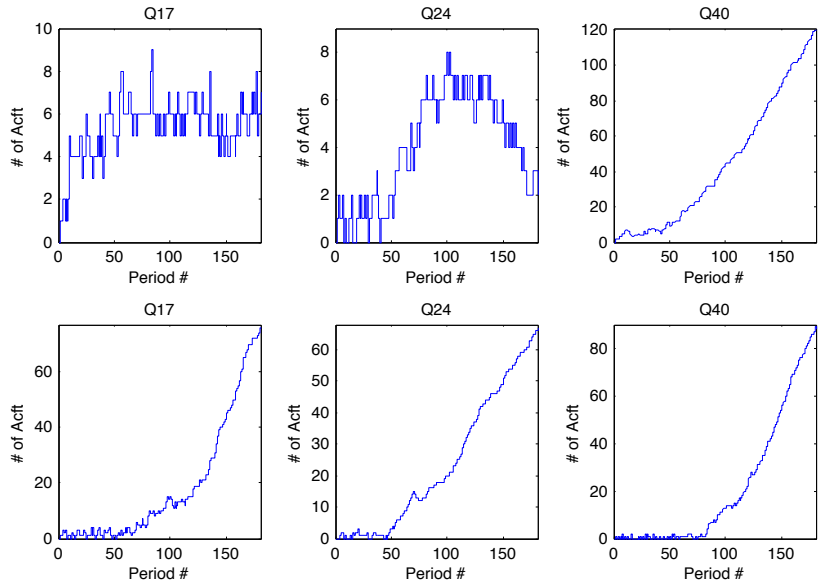


Figure 11. Sample path of the trajectories of initial queues for the flows starting East, South, and North, over a period of 12 hours (180 periods) with a bad weather event simulated according to the automaton shown on Fig. 9. The top line shows the MaxWeight policy and the bottom line the FCFS policy. No policy can stabilize queue 40 due to the lack of landing capacity at SFO. In addition, the congestion at the intersection East of SFO created by the FCFS policy prevents the other flows to cross and destabilizes them. The MaxWeight policy does not exhibit this issue, although it might increase the congestion on the East flow even more.

Simulation results obtained from 100 simulations in each set of experiments are shown in Figure 10. The FCFS policy has good throughput performance under nominal conditions, and hence is often an adequate low complexity policy to use. Here we define throughput as the the total number of aircraft landing during the simulation. It is purely local and sends aircraft forward as soon as possible however, which has detrimental effects in the case of bad weather reducing throughput through control boundaries. In our scenario, the landing rate at SFO during bad weather (state M or H) is not sufficient to accommodate all incoming flows, in particular the large one coming from the East direction. As a result, queue 40 is unstable under any policy. A drawback of the FCFS policy however is that it also quickly saturates the intersection East of SFO, blocking the North/South traffic between SEA/PDX and LAX. That is, we find that under the FCFS policy the departure queues $Q24$ and $Q17$ are unstable, see Fig. 11, even though the other policies can maintain these queues stable. This is the case of the MaxWeight policy for example. The results reported on Fig. 10 however show a worst performance for MaxWeight than for FCFS for this problem, and it was found in this case that MaxWeight suffers from a somewhat lacking throughput at the relevant time scale and its better stability properties compared to FCFS were significant only for longer horizons. Overall, the conclusions that MaxWeight can mitigate the coupling effects between flows when compared to FCFS at basically the same implementation cost, already illustrated in subsection V.A, make this policy still worth considering at least locally for regions supporting complex flow patterns.

Again the MPC policy was found to perform best, but requires the dynamical model of the buffers (throughput functions) as well as the knowledge of the transition probabilities of the weather automaton. In particular it stabilizes the North/South flow, by controlling the departure rates at initial queues to avoid saturating the intersection. Note also that in general, airborne delays are more costly than ground delays,[11] which is the main reason (along with safety concerns) for implementing ground holding programs. It is straightforward to take this aspect into account in the MPC approach, by adjusting the coefficients γ_i accordingly for the different queues to the cost of a delay of one time period at the associated geographic region. In contrast, FCFS and MaxWeight have no direct means of distinguishing different costs for aircraft waiting at different queues and need to be complemented by additional rules to avoid large airborne queues. MaxWeight tends to behave somewhat better than FCFS in this regard however, at least before hard capacity constraints on the sector aircraft counts are reached, because it requires a positive "pressure" between the ground departure queues and the first airborne queue before ordering aircraft to take-off. Finally, note that the MPC controller could have been optimized to maximize the throughput, by replacing the cost mentioned above for example by $100\, Q_{K+1} + \sum_{k=0}^{K} \tilde{\gamma}_k^T\, U_k$, where $\tilde{\gamma}_{k,i} = -1$ if $U_{k,i}$ controls the output of an arrival queue at an airport (i.e. the queues $3, 9, 14, 23, 43, 46, 49, 35, 37$) and $\tilde{\gamma}_{k,i} = 0$ otherwise.

VI. Discussion and Conclusions

VI.A. Summary on Modeling and Control for TFM

Let us summarize the methodology proposed to optimize TFM strategies over a given region of interest and mention some future work needed to put this framework into practice. We start by defining control boundaries, with control volumes defined as a byproduct. The geometry of these boundaries and volumes is strongly influenced by the preferences of the air traffic controllers at the tactical level. Namely, control boundaries are placed where it is most convenient for ATCs to impose rate restrictions, e.g. at sector boundaries and usual metering points. Also, a sampling period is defined, loosely depending on the size of the control volumes and on the capability of the ATCs to execute most flow rate directives over the time period. Once this is done, the control structure is hierarchical, with the TFM level issuing flow directives to the ATC level at the beginning of each period. These directives specify the number of aircraft of different types (e.g. different destinations) allowed to cross each boundary during the period. At the TFM level, the different types of aircraft are organized in different buffers. The ATCs try to execute the TFM directives during the time period, and at the end of the period report the new aircraft count in each volume to the TFM level, which might differ from the expected one if some directives could not be implemented. Deviations with respect to the directives is treated as a disturbance and taken into account by the feedback form of the policies.

How much information the TFM level needs to issue directives at the beginning of each period depends on the particular control strategy implemented. For the simplest strategies, e.g. FCFS and MaxWeight, the necessary information is minimal, and no dynamic model of the system is necessary. Namely, the TFM level requires a prediction of the maximum number of aircraft of each type that the ATCs can send through a control boundary during the period (the vector $D(k)$), the aircraft count in the volumes at the beginning of the period (the vector $Q(k)$), and the knowledge of the rate constraints (11) and capacity constraints (9). For these strategies, the separation of the controller into two layers is in fact rather unnecessary as the policies require no or very few computations.

We have shown examples of undesirable congestion effects exhibited by myopic policies such as FCFS and MaxWeight however, in particular due to coupled capacity constraints. Hence we presented an MPC strategy that takes a more global view of the TFM optimization problem. However, it requires much more information and computations for its implementation. Departure rates at airports and the linear rate constraints (11) are relatively easy to obtain.[17,45] MPC also requires the dynamic throughput model for each volume in the form of the functions $\mu(q)$. Finally, the MPC procedure can incorporate and plan using weather forecasts in the form of stochastic predictions of future volume capacities and rate constraints at control boundaries.

VI.B. Conclusion

This paper presented an improved Eulerian model that can be used to develop closed-loop control policies for the NAS and that takes into account all air traffic resources, including airport capacity envelopes. The model is very flexible and provides decision support to air traffic controllers to control traffic flow rates at the control boundaries of their choice. We present simulation results for the performance of natural control strategies for the system, with different information requirements. We believe that the proposed model is particularly useful in developing planning strategies during extreme weather events, where coordination of NAS resources at distant locations is necessary. Future work will consider the interface between the TFM level and the lower tactical level in more details, in order to properly identify the maximum throughput curves required by our model, and to validate the developed control laws using more precise trajectory-based simulations.

Acknowledgments

This work was supported by NSF Contract ECCS-0745237 and NASA Contract NNA06CN24A.

References

[1] H. Arneson and C. Langbort. Distributed sliding mode control design for a class of positive compartmental systems. In *Proceedings of the American Control Conference*, 2008.

[2] S. Atkins. Estimating departure queues to study runway efficiency. *AIAA Journal of Guidance, Control and Dynamics*, 25(4):651–657, July-August 2002.

[3] M. O. Ball, R. Hoffman, A. R. Odoni, and R. Rifkin. A stochastic integer program with dual network structure and its application to the ground-holding problem. *Operations Research*, 51(1):167–171, Jan.-Feb. 2003.

[4] A. M. Bayen. *Computational Control of Networks of Dynamical Systems: Application to the National Airspace System*. PhD thesis, Stanford University, 2003.

[5] D. P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 3rd edition, 2005.

[6] D. Bertsimas, G. Lulli, and A. Odoni. The air traffic flow management problem: An integer optimization approach. In *Integer Programming and Combinatorial Optimization*, pages 34–46, 2008.

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

[8] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[9] Bureau of Transportation Statistics. Understanding the reporting of causes of flight delays and cancellations. http://www.bts.gov/help/aviation/html/understanding.html.

[10] B. G. Chandran. Predicting airspace congestion using approximate queueing models. Master's thesis, University of Maryland at College Park, 2002.

[11] A. Cook, G. Tanner, and S. Anderson. Evaluating the true cost to airlines of one minute of airborne or ground delay. Technical report, University of Westminster, May 2004.

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

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

[14] R. de Neufville and A. Odoni. *Airport Systems: Planning, Design and Management*. McGraw-Hill, 2003.

[15] R. Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer, 2005.

[16] C. E. García, D. M. Prett, and M. Morari. Model predictive control: Theory and practice–a survey. *Automatica*, 25(3):335 – 348, 1989.

[17] E. P. Gilbo. Airport capacity: Representation, estimation, optimization. *IEEE Transactions on Control Systems Technology*, 1(3):144–154, September 1993.

[18] E. P. Gilbo. Optimizing airport capacity utilization in air traffic flow management subject to constraints at arrival and departure fixes. *IEEE Transactions on Control Systems Technology*, 5(5):490–503, September 1997.

[19] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming. http://stanford.edu/ boyd/cvx.

[20] A. Haraldsdottir, R. W. Schwab, A. Shakarian, G. Wood, and R. S. Krishnamachari. ATM Operational Concepts and Technical Performance Requirements. In Lucio Bianco, Paolo Dell'Olmo, and Amedeo R. Odoni, editors, *New Concepts and Methods in Air Traffic Management*, pages 63–74. Springer, 2001.

[21] J. Le Ny and H. Balakrishnan. Distributed feedback control for an Eulerian model of the national airspace system. In *Proceedings of the American Control Conference*, 2009.

[22]D. Long, D. Lee, J. Johnson, E. Gaier, and P. Kostiuk. Modeling air traffic management technologies with a queueing network model of the national airspace system. Technical Report CR-1999-208988, NASA, 1999.

[23]K. M. Malone. *Dynamic queueing systems: Behavior and approximations for individual queues and for networks*. PhD thesis, Massachusetts Institute of Technology, 1995.

[24]P. K. Menon, G. D. Sweriduk, and K. D. Bilimoria. New approach for modeling, analysis, and control of air traffic flow. *Journal of Guidance, Control, and Dynamics*, 27(5):737–744, 2004.

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

[26]S. Meyn. *Control Techniques for Complex Networks*. Cambridge University Press, 2008.

[27]D. Michalek and H. Balakrishnan. Building a stochastic terminal airspace capacity forecast from convective weather forecasts. In *Proceedings of the Aviation, Range and Aerospace Meteorology Special Symposium on Weather-Air Traffic Management Integration, American Meteorological Society 89th Annual Meeting*, January 2009.

[28]D. Michalek and H. Balakrishnan. Identification of robust routes using convective weather forecasts. In *Eighth USA/Europe Air Traffic Management Research and Development Seminar (ATM2009)*, 2009.

[29]A. Mukherjee and M. Hansen. A dynamic rerouting model for air traffic flow management. *Transportation Research Part B: Methodological*, 43(1):159 – 171, 2009.

[30]A. Nemirovski and A. Shapiro. Convex approximations of chance constrained programs. *SIAM Journal on Optimization*, 17(4):969–996, 2006.

[31]F. Neuman and H. Erzberger. Analysis of delay reducing and fuel saving sequencing and spacing algorithms for arrival spacing. NASA Technical Memorandum 103880, 1991.

[32]A. Nilim, L. El Ghaoui, and V. Duong. Multi-aircraft routing and traffic flow management under uncertainty. In *Proceedings of the 5th US/Europe Air Traffic Management R&D Seminar*, pages 23–27, Budapest, Hungary, 2003.

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

[34]N. Pujet, B. Delcaire, and E. Feron. Input-output modeling and control of the departure process of congested airports. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, 1999.

[35]Michael Robinson, R. DeLaura, B. Martin, J. E. Evans, and M. E. Weber. Initial studies of an objective model to forecast achievable airspace flow program throughput from current and forecast weather information. In *Aviation, Range and Aerospace Meteorology Special Symposium on Weather-Air Traffic Management Integration, AMS Annual Meeting*, Phoenix, AZ, January 2009.

[36]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, 2003.

[37]T. I. Seidman. First come, first served can be unstable! *IEEE Transactions on Automatic Control*, 39(10):2166–2170, October 1994.

[38]R. A. Shumsky. *Dynamic Statistical Models for the Prediction of Aircraft Take-Off Times*. PhD thesis, Massachusetts Institute of Technology, 1995.

[39]I. Simaiakis and H. Balakrishnan. Queuing models of airport departure processes for emissions reduction. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2009.

[40]B. Sridhar, T. Soni, K. Sheth, and G. B. Chatterji. Aggregate flow model for air-traffic management. *Journal of Guidance, Control, and Dynamics*, 26(4):992–997, 2006.

[41]D. Sun and A. M. Bayen. Multicommodity Eulerian-Lagrangian large-capacity cell transmission model for en route traffic. *Journal of Guidance, Control, and Dynamics*, 31(3), 2008.

[42]D. Sun, I. S. Strub, and A. Bayen. Comparison of the performance of four Eulerian network flow models for strategic air traffic management. *Networks and Heterogeneous Media*, 2(4):569–594, December 2007.

[43]L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control*, 37(12):1936–1948, December 1992.

[44]M. Terrab and A. R. Odoni. Strategic flow management for air traffic control. *Operations Research*, 41(1):138–152, Jan.-Feb. 1993.

[45]U.S. Department of Transportation. Airport capacity benchmark 2004, 2004.

[46]P. B. M. Vranas, D. Bertsimas, and A. R. Odoni. Dynamic ground-holding policies for a network of airports. *Transportation Science*, 28(4):273–291, November 1994.