University of Pennsylvania
ScholarlyCommons

Technical Reports (ESE) | Department of Electrical & Systems Engineering

6-17-2010

Feedback Control of the National Airspace System

Jerome LE NY
University of Pennsylvania, jerome.le-ny@polymtl.ca

Hamsa Balakrishnan
Massachusetts Institute of Technology, hamsa@mit.edu

This paper is posted at ScholarlyCommons. http://repository.upenn.edu/ese_reports/5
For more information, please contact repository@pobox.upenn.edu.

Feedback Control of the National Airspace System

Abstract
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.

Keywords
air traffic control, network control, transportation networks

Disciplines
Controls and Control Theory | Multi-Vehicle Systems and Air Traffic Control

This working paper is available at ScholarlyCommons: http://repository.upenn.edu/ese_reports/5

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 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.[4,37] 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.[7] Moreover, open-loop traffic flow management algorithms require weather forecasts several hours ahead of time, which are arguably beyond the limits of even state-of-the-art weather forecasting tools.[23]

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.[26,39] 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.[20,21,30,33,35] They have been shown to have reasonable predictive capabilities[35] and are compatible with today's traffic flow management system, which regulates traffic flow rates rather than planning for individual aircraft.[33] Airport operations in the

*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.

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.[2] Eulerian models that determine arrival rates at airports therefore can be used within the CDM framework. Prior queuing network models of the NAS have generally been developed with a view towards performance prediction and analysis, and not control,[8,18,19,30] which is our objective in this work.

Feedback control schemes using Eulerian models have been previously proposed, both in the context of centralized traffic flow management,[21,25] 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.[21] 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.[20,21,35] 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,[35] 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,[22] 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,[35] 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[36] that results in *distributed* feedback control laws for traffic flow planning at different facilities. This policy reflects the control structure in the current system,[16,27] in which facility (airport or sector)-level traffic planning is done locally through coordination with neighboring facilities.

At the TFM level, our model provides 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).[29] 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. It is therefore important to develop sufficiently precise models for TFM in order to minimize the impact of these lower level disturbances. 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 a few natural control policies for this model, including 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. We describe how it can then be used to mitigate the impact of weather disruptions, given such probabilistic forecasts. Finally, we present simulation results for the TFM problem over part of the Western United States, and compare the different control strategies.

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.[35]

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.

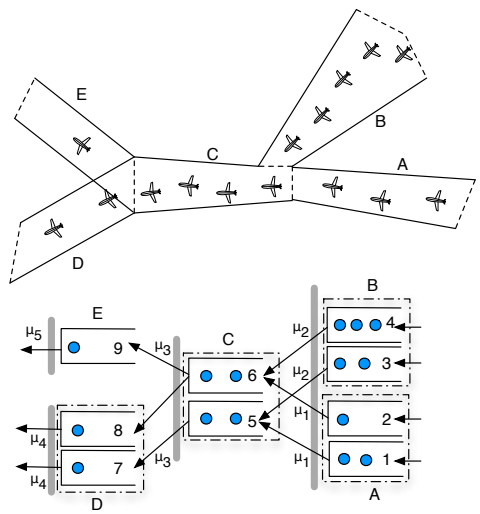

Figure 1. Five control volumes and their corresponding abstract buffer model. The situation depicted here corresponds to two routes merging and then diverging. Each control volume except E supports two flows with distinct characteristics (e.g., different destinations, as described Section II.B.1), and 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, $k \in \mathcal{N}$, is denoted $Q_i(k)$. Every buffer is associated with a control volume (see Figure 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

$$Q_i(k+1) = Q_i(k) + A_i(k) + \textstyle\sum_{j \in \mathcal{J}_i} U_j(k) - U_i(k), \tag{1}$$

$$0 \leq U_i(k) \leq D_i(k), \ \forall k \geq 0. \tag{2}$$

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). $U_i(k)$ is the number of departures from buffer i during the same period, and is controlled by the ATC. $\mathcal{J}_i$ is the set of buffers sending aircraft to buffer i. These terms will be discussed in more detail later. 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 $U_i(k)$. Calculating $D_i(k)$ requires the accurate prediction of the maximum number of aircraft from buffer i that can cross the exit boundary of the control volume during the $(k+1)^{\text{th}}$ period, a simple trajectory prediction problem for the typical values of T used in flow control (1-15 min).

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. 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. The maximum throughput curve for a given buffer or control volume can be obtained via simulation, by fitting empirical data, or a combination of the two. 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.

II.B. Additional Resource Constraints

We construct a network model of the NAS using control volumes containing buffers that follow the dynamics of Equations (1) and (2) as building blocks. 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.

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

For a control volume containing m buffers, the control vector for the exit boundary is denoted by

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

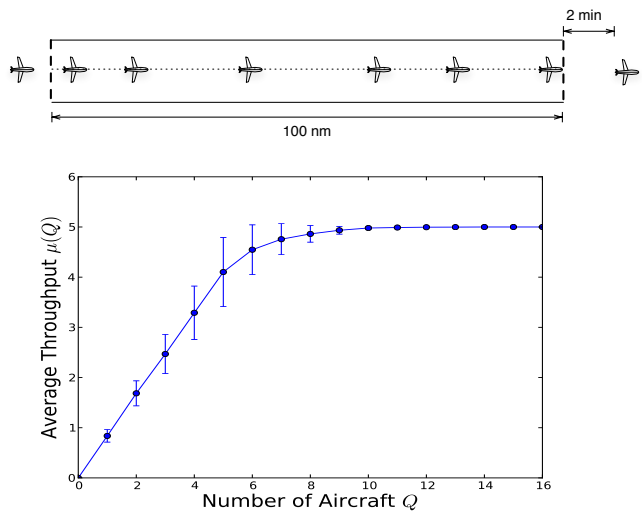

Figure 2. Maximum throughput for the exit boundary of a simple linear control volume. The empirical curve of the mean number of departures per period was obtained via discrete-event simulation, with random arrivals. The error bars show the empirical variance of the number of departures per period. In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

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

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

In general we take $c_i = 1$ for all i, and then r is simply the maximum number of aircraft that can cross the boundary per period. 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. $U^{(1)}$ 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.

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

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

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

II.B.3. Routing

We can easily can incorporate routing decisions into the proposed scheduling model by adding control variables. Allowing ATCs to make tactical routing decisions can help accommodate dynamically changing conditions in the network, such as the impact of weather on capacities.[25] In the model shown in Figure 1, aircraft from buffer 6 in control volume C can enter buffer 8 or 9. The dynamics of buffer 6 are then

$$Q_6(k+1) = Q_6(k) + A_6(k) + U_2(k) + U_4(k) - (U_{6,8}(k) + U_{6,9}(k))$$
$$U_{6,8}(k) \geq 0, U_{6,9}(k) \geq 0, U_{6,8}(k) + U_{6,9}(k) \leq D_6(k).$$

Here $U_{6,8}$ (resp. $U_{6,9}$) represents the number of aircraft routed by the ATC from buffer 6 to buffer 8 (resp. 9).

II.B.4. 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

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

where M is a matrix and S a vector. However, it is difficult to enforce such constraints explicitly in many control algorithms. In these cases, we resort to an iterative process where the control law is developed without enforcing the constraint (6), and then modified until these constraints are approximately satisfied.

II.B.5. Airport Resources

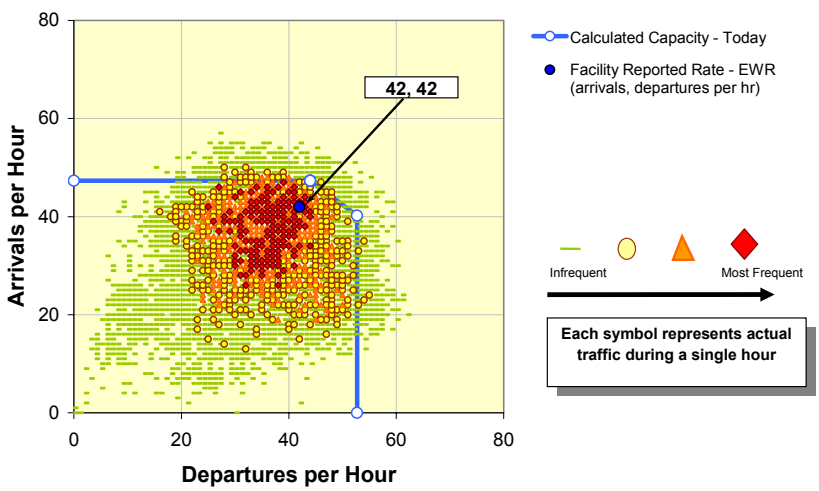

Figure 3. Capacity envelope observed at Newark airport, under optimal conditions.[38] Any pair of arrival/departure allocation rates within the polytope is achievable. This polytope corresponds to constraints of the form (5), and depends on the airport runway configuration.[13]

An airport is modeled using arrival queues (aircraft waiting to land, which are associated with air traffic flows in the close 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.[13,14] These constraints depend on runway configurations and can be determined empirically or analytically.[13,38] 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.[31,32] 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, and has the form shown in Figure 2.[28,31,32]

II.B.6. 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 using an example in Subection IV.C. 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;[17] 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

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

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

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

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

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

The matrix B consists of $+1$'s and -1's and is the incidence matrix[11] 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[22] 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

In recent years, several Eulerian models have been proposed for air traffic flow management.[20,21,30,33–35] While some previous work has focused on the capabilities of Eulerian models[30,33,35] to predict aircraft counts in control volumes,

these models are less precise than trajectory-based simulation tools such as FACET[5] 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.

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), and makes no other assumptions on the dynamics of the system. Menon et al.[20] 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[9,10] (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,[21] and a model predictive control strategy is proposed to handle these constraints. Similar models are used by Sridhar et al.,[33] and by Sun and Bayen.[34] 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 detailed modeling example is presented in Section IV.C). 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[34] 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.[34] 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. In Sun and Bayen,[34] the authors 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

$$Q_p^i(t+1) = Q_{p-1}^i(t) - \tilde{U}_{p-1,p}^i(t) + \tilde{U}_{p,p+1}^i(t) + A_p^i(t). \tag{14}$$

These are, in fact, the exact dynamics presented by Sun et al.[34] (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 $U_{p,p+1}^i(t) \leq Q_p^i(t)$. 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[28,31,32] 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,[34] 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.[22] 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.,[2] and Mukherjee and Hansen[25] 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 the TFM problem in this setting. In contrast, we treat the various quantities as continuous in our optimization, 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 closed loop 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.[22] We note that the state and control constraints in particular, neglected in the original work of Menon et al.,[20] 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.

In this paper, we describe two natural control strategies, each with very different computation and implementation requirements, that can be used for scheduling and routing aircraft through the NAS. The first strategy, the MaxWeight policy,[36] is a distributed policy with very few implementation requirements that can be used under nominal conditions. In contrast, the second approach is a Model Predictive Control (MPC) strategy that has greater implementation requirement, and is more suitable for use in rapidly-changing capacity scenarios such as bad weather days, as discussed in Section IV.

III.A. Distributed MaxWeight Policy

The celebrated MaxWeight or maximum back pressure policy for network control,[36] which was previously proposed by the authors for air traffic control,[17] can be obtained as follows. We assume that the sector capacity constraints (9) are not present. At period k, in state $Q(k) = Q$ of the network, with right-hand sides $D(k) = D$ in (10), and R in (11), we apply the control

$$U^{MW}(Q) \in \arg\min_{U \in \mathcal{U}} \langle BU, \Xi Q \rangle = \arg\min_{U \in \mathcal{U}} Q^T \Xi BU, \tag{15}$$

where $\mathcal{U} = \{U | C_1 U \leq D, C_2 U \leq R, U \geq 0\}$, and Ξ is a diagonal weighting matrix with positive coefficients ξ_i. The choice of Ξ is left open here and allows the controller to prioritize certain flows over others.[17] For the simulation results of Section IV.C, we set Ξ to be the identity matrix.

One possible motivation for this control law is the following. To the discrete-time dynamics (8), we can associate the following continuous-time model, called the *fluid model*[22]

$$\begin{aligned} &\frac{dq}{dt} = B\zeta(t) + \alpha(t) \\ &C_1\zeta \leq \mu, C_2\zeta \leq R, \zeta \geq 0, \end{aligned}$$

with control vector ζ. Suppose we consider the quadratic Lyapunov function

$$V(q) = \frac{1}{2} q^T \Xi q. \tag{16}$$

The MaxWeight policy maximizes the rate of decrease (given by the Lie derivative) of V, since

$$\frac{d}{dt} V(q(t)) = \langle B\zeta, \Xi q \rangle + \langle \alpha, \Xi q \rangle, \tag{17}$$

and the second term of Equation (17) is independent of ζ.

III.A.1. Implementation Requirements of MaxWeight

The interesting fact about the MaxWeight policy, as given by Equation (15), 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 (15) can be written explicitly as

$$U^{MW}(Q) \in \arg\min_{U \in \mathcal{U}} \sum_{e \in \mathcal{E}} U_e(\xi_{e^+} Q_{e^+} - \xi_{e^-} Q_{e^-}), \text{ or,}$$
$$U^{MW}(Q) \in \arg\max_{U \in \mathcal{U}} \sum_{e \in \mathcal{E}} U_e(\xi_{e^-} Q_{e^-} - \xi_{e^+} Q_{e^+}), \tag{18}$$

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 (18) 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 can couple control variables that involve shared resources. The linear program (18) 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

$$\sum_{i=1}^{N} U_{e_i} \leq R, \tag{19}$$

where the control variables are 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). 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 (19). 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 (19) became active. If this constraint is not yet active and if $\pi_{e_l} \geq 0$, it sets U_{e_l} to $\min\{D_{e_l}, R - \sum_{i=1}^{l-1} U_{e_i}\}$. 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, 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 shape of the maximum throughput functions, throughout the network. 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.A.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, an 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,[22] 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 $Q(k)$ evolves as a Markov chain. The network is said to be *stable* if for each initial condition $Q(0) = x$, the average cost

$$\eta_x = \lim \sup_{n\to\infty} \frac{1}{n} \sum_{k=0}^{n-1} E[\|Q(k)\|_\infty], \tag{20}$$

is finite. In the absence of the capacity constraints (9) and the throughput constraints (10), the MaxWeight policy is known to be stabilizing for any network that is stabilizable.[22] As in the previous subsection, we continue to ignore the capacity constraint (9). The throughput constraints (10) force the flow rate out of a buffer to be small when the buffer is lightly loaded. With these low throughputs at light loads, the number of aircraft in the system can grow. At high loads, the throughputs increase and approach their saturation values, and for the network to be stable these saturation rates should be sufficiently high to keep the queues bounded.

Define

$$\delta(x) = \mathbb{E}\left[\min_{\substack{U\geq 0: C_1U\leq D,\\ C_2U\leq R}} \langle BU + A(k), \Xi Q(k)\rangle \Big| Q(k) = x\right],$$

which intuitively represents the average decrease per period of the Lyapunov function (16) due to the MaxWeight policy. Next assume that there is a positive constant K and $\epsilon > 0$ such that for all x with $\|x\|_\infty > C$, we have $\delta(x) < -\epsilon\|x\|_\infty$. This assumption generalizes a standard stability condition for networks that are not subject to the saturation constraint (10).[22]

Theorem 1 *Under the preceding assumptions, the network model (8)-(12) is stabilized by the MaxWeight policy.*

Proof Consider the Lyapunov function (16) and the Lyapunov drift under the MaxWeight policy

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

Then with our hypothesis on $\delta(x)$, it is not hard to see that

$$\mathcal{D}V(x) \leq -\epsilon\|x\|_\infty + b, \quad \forall \|x\|_\infty > C, \tag{21}$$

where

$$b = \frac{1}{2}\max_{x,u} \mathbb{E}[(Q(k+1) - Q(k))^T \Xi (Q(k+1) - Q(k))|Q(k) = x, U(k) = u].$$

Here b is a finite constant because the maximization is over the feasible control vectors, which is a bounded set. Equation (21) implies the stability of the network as a consequence of the Foster-Lyapunov criterion (Theorem 8.0.3 in Meyn[22]). ∎

III.B. Model Predictive Control

Model Predictive control (MPC)[12] 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.[21] 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 $\mathbf{Q} = \{Q_k\}_{k_0+1 \leq k \leq k_0+K+1}, \mathbf{U} = \{U_k\}_{k_0 \leq k \leq k_0+K}$

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

$$\text{subject to} \tag{23}$$

$$Q_{k_0+1} = Q(k_0) + BU_{k_0} + A(k_0)$$

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

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

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

$$C_1 U_k \leq \mu(Q_k), \;\; k_0 + 1 \leq k \leq k_0 + K \tag{26}$$

$$C_2 U_k \leq R, \;\; k_0 \leq k \leq k_0 + K, \tag{27}$$

$$U_k \geq 0, \;\; k_0 \leq k \leq k_0 + K,$$

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

$$f(\mathbf{Q}) := \sum_{k=k_0+1}^{K} \gamma_k^T Q_k + \gamma_{K+1}^T Q_{K+1}, \tag{28}$$

for some positive vectors $\{\gamma_k\}_k$. We use the certainty-equivalence heuristic[3] 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.

Upon solving the optimization problem (22), we obtain a sequence of vectors $U_{k_0}, \ldots, U_{k_0+K}$, which are real-valued. We round-off the first vector U_{k_0}, and use it as a control directive $U(k_0) = U_{k_0}$ for the current period. We discard the other vectors $U_{k_0+1}, \ldots, U_{k_0+K}$. 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.[6] 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 $m_i = 1$).

$$c_i^T U \leq a_{ij}^T Q_i + b_{ij}, \; j = 1, \ldots, m_i \tag{29}$$

$$c_i^T U \leq \mu_{i,sat}.$$

In this case, the constraints (26) are replaced by linear constraints and the program (22) becomes a linear program. The hard constraint (24) is usually removed to avoid infeasibility issues in the optimization procedure. Instead, we can try to respect capacity constraints by penalizing large sector loads in the objective function.

As mentioned in Section II.D, we use a naive rounding procedure to obtain an integer-valued control vector U at each period (making sure that the constraints (25) and (27) for $k = k_0$ are satisfied). An alternative approach would be to solve the program (22)-(27) as an integer program, producing directly an integer solution for U_{k_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

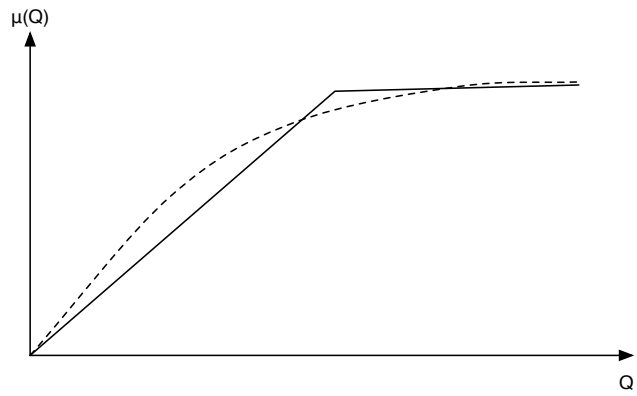

Figure 4. Simple piecewise-linear approximation of the throughput curves used in the MPC optimization step, see (29).

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. This requirement also necessitates a model that is not too large, or a discretization that is not too fine, as our model allows. MPC can also potentially benefit from recent advances in convex optimization in solving large-scale problems.[40]

IV. Weather Management

Our discussion so far assumed that parameters such as the average capacities at control boundaries were known constants. However, if a region experiences bad weather, its capacity to accommodate traffic flows can be greatly reduced. Weather forecasts useful for detailed aircraft route planning are becoming increasingly available for TFM.[24] 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),[29] but 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.B to the situation where the capacity constraints evolve randomly in time according to a probabilistic weather forecast.

IV.A. A Model Integrating Weather Uncertainty

We 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, determined from probabilistic weather forecasts. Note that we can use such a model to represent weather scenarios of the form considered by Ball et al. or Terrab and Odoni,[2,37] 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 accomodate, and therefore depends on the weather state, and is decided by traffic management initiatives such as AFPs.[29] 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.5). These envelopes vary in shape according to the weather state.[38] Hence we replace the constraints (11) by $C_2(w(k))U(k) \leq R(w(k))$. 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.B. Namely, at period k_0, we observe the weather state $w(k_0)$, and we replace the constraints (24)-(27) by

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

$$C_1(w(k_0))U_{k_0} \leq D(k_0)$$

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

$$C_2(w(k_0))U_{k_0} \leq R(w(k_0))$$

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

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.B. In our experiments described in the next section, we did not include the state constraint (30), which can lead to potentially infeasible programs, and we assumed that the throughput function μ of the buffers was independent of the weather state, using constraint (26) rather than (31).

IV.C. Simulation Results

We consider the TFM problem over a portion of the western part of the United States. 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. The resulting queueing network is shown in Figure 5. 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.[38] The throughput functions for the control volumes are only approximately identified, based on the length of the control volumes, the required separation distance (5 nm), and assuming simple linear traffic flows in the volumes, as in Figure 2. In order to solve the MPC optimization problem as a linear program, these traffic curves are then approximated by piecewise linear functions consisting of just two pieces, as shown in Figure 4. The departure rates at airports and other external arrival rates are time-varying and oscillate around the values provided in Figure 5.

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.[38] 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, the weather changes to one of 3 states: low (L), medium (M) or high (H) weather impact. In the H state, SFO allows no departures and accepts only 10 arrivals per hour; L corresponds to IFR conditions and M to Marginal conditions.[38] The Markov chain can jump between these 3 states for some time, but eventually reaches a final absorbing state, in which the capacity is again optimal. The transition structure of the chain modeling the weather dynamics is shown in Figure 5.

We use the CE-MPC controller in the linear programming form described in Section IV.B. The time-discretization T is set to 4 minutes, simulations are run for a 6 hour duration, i.e., 90 periods. The horizon K for MPC is set to $32, 92$ and 152 minutes in 3 different sets of experiments (hence $K = 8, 23$ and 38 time periods, respectively). The cost function we wish to optimize is also used as the MPC objective (28), and consists of a time-invariant cost $\gamma_1 = \ldots = \gamma_{91}$, with coordinates $\gamma_{k,i} = 0.5$ if queue i is in a control volume on the ground (departure queues at airports), $\gamma_{k,i} = 2$ if queue i is in a terminal area (arrival queues in the airspace around airports) and $\gamma_{k,i} = 1$ otherwise. The purpose of using a higher cost in terminal areas is to reduce congestion in these regions, where the control volumes support many incoming traffic flows.

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,[15] without trying to improve the computational performance

Table 1. Typical computational requirements per period to solve the linear program for the MPC heuristic. Computations were done using MATLAB and CVX[15] 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	$\leq 5s$

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. The simulation results, obtained from 100 simulations in each set of experiments, are shown in Figure 6. We also show the performance of the MaxWeight policy for this example, which can be computed at negligible cost. Note that MaxWeight has no direct means of distinguishing different costs for aircraft waiting at different queues. In particular, the fact that the cost for a unit of ground delay is smaller than the cost of a unit of air delay cannot easily be taken into account by MaxWeight, but is the main reason (along with safety concerns) for implementing ground holding programs. Because MaxWeight uses only local information, it pushes aircraft out of airports as soon as the loads in neighboring sectors allow it. Hence it is not surprising that MPC performs better than MaxWeight for the weighted objective, by actively reducing the departure rates at airport in order to avoid air congestion. It is interesting to remark however that essentially no performance gain is 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. In Figure 7, we show the results for the same system and a scenario with no weather perturbation. In this case, increasing the planning horizon of MPC beyond 30 min has some performance impact, although not particularly large. The gap between the performance of MPC and MaxWeight for the weighted cost becomes smaller, as MaxWeight is able to clear the airspace traffic faster.

On the right of Figures 6 and 7, we also show the total number of aircraft in the system during the simulations, independently of their position (ground, terminal airspace or non-terminal airspace). This corresponds to the objective (28) with an all-one cost vector at each step. For this metric (which, note, is different from the cost function optimized by MPC), we see that MaxWeight outperforms the MPC solution with a 30 min planning horizon. Better performance is obtained for the MPC solution with larger planning horizons. This suggests that the different MPC policies, despite having similar total weighted cost, can behave differently with respect to other metrics, by distributing aircraft differently in the system. It appears that increasing the planning horizon makes the MPC solution more robust to changes in the objective function.

V. 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 two 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. Future work will consider the interface between the TFM level and the lower tactical level, 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.

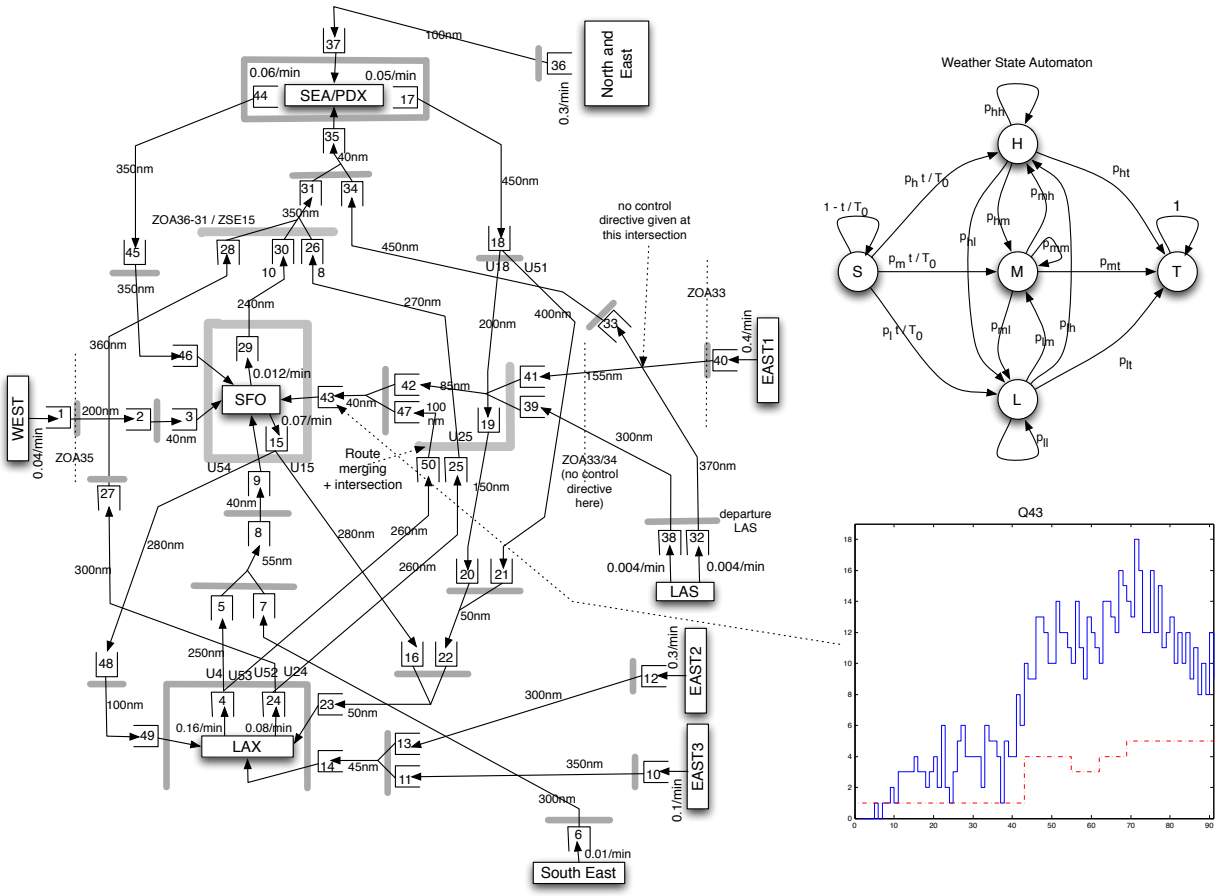

Figure 5. Network model for TFM in the simulated system. Thick gray lines correspond to control variables coupled via the linear constraints (10), (11). All queues start initially empty. We also show the automaton describing the probabilistic weather evolution as well as a sample trajectory of a queue in the system. The red dashed line shows the evolution of the weather state for that sample, namely $S - H - M - H - T$.

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] 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.

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

[4] 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.

[5] 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.

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

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

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

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

[10] C.F. Daganzo. The cell transmission model, part ii: Network traffic. In *Transportation Research*, volume 29B, pages 79–93, 1995.

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

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

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

[14]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.

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

[16]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.

[17]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.

[18]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.

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

[20]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.

[21]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.

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

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

[24]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.

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

[26]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.

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

[28]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.

[29]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.

[30]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.

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

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

[33]B. Sridhar, T. Soni, K. Sheth, and G.B. Chatterji. Aggregate flow model for air-traffic management. In *Journal of Guidance, Control, and Dynamics*, volume 26, pages 992–997, 2006.

[34]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.

[35]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.

[36]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.

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

[38]U.S. Department of Transportation. Airport capacity benchmark 2004. Technical report, 2004.

[39]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.

[40]Y. Wang and S. Boyd. Fast model predictive control using online optimization. *IEEE Transactions on Control Systems Technology*, 2008. To appear.

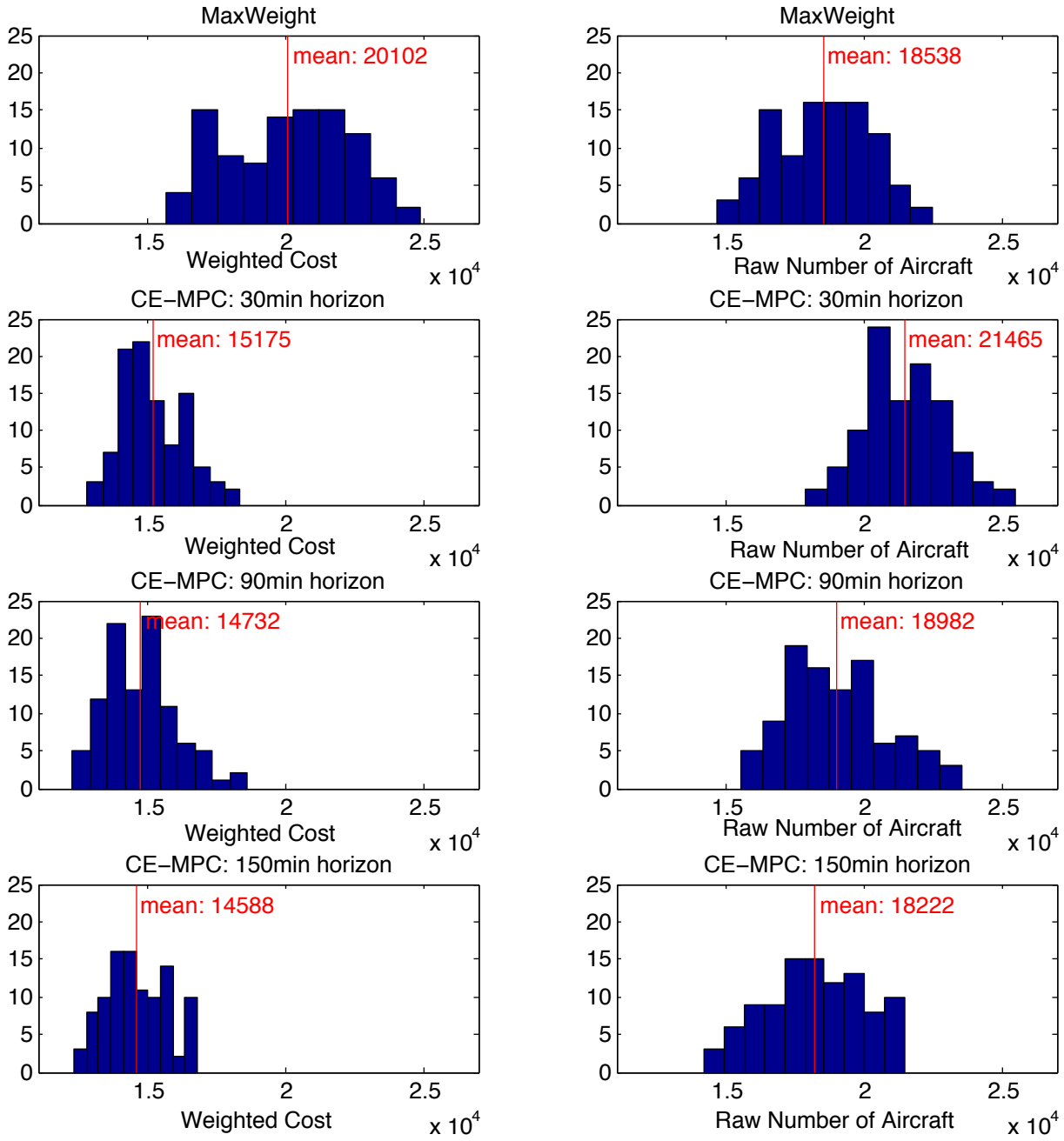

Figure 6. Histograms of the performance results for 100 simulations, for a scenario with probabilistic forecasts as described on Fig. 5. Each simulation corresponds to a 6 hour interval. On the left, the cost at each period during a simulation is 0.5 per aircraft on the ground, 2 per aircraft in the terminal airspace arriving at airports, and 1 for other airborne aircraft. This is also the cost used in the MPC optimization step (see objective (28)). The performance of MaxWeight (with $\xi_i = 1$ for all i) is shown as well, although this policy does not take into account the weather forecasts, the knowledge of the future arrival rates, or the difference in cost between ground and airborne delays. On the right, we show the total cumulative number of aircraft in the system during the 6 hour period of the simulation (in other words, an aircraft is counted once for every period in which it remains in the system). Under this metric, the performance of the MaxWeight policy is close to the best performance achievable by the MPC solution. Note however that the performance of the MPC solution could be improved for this metric by changing the cost function (28), taking the vectors γ_k to be all-one vectors.

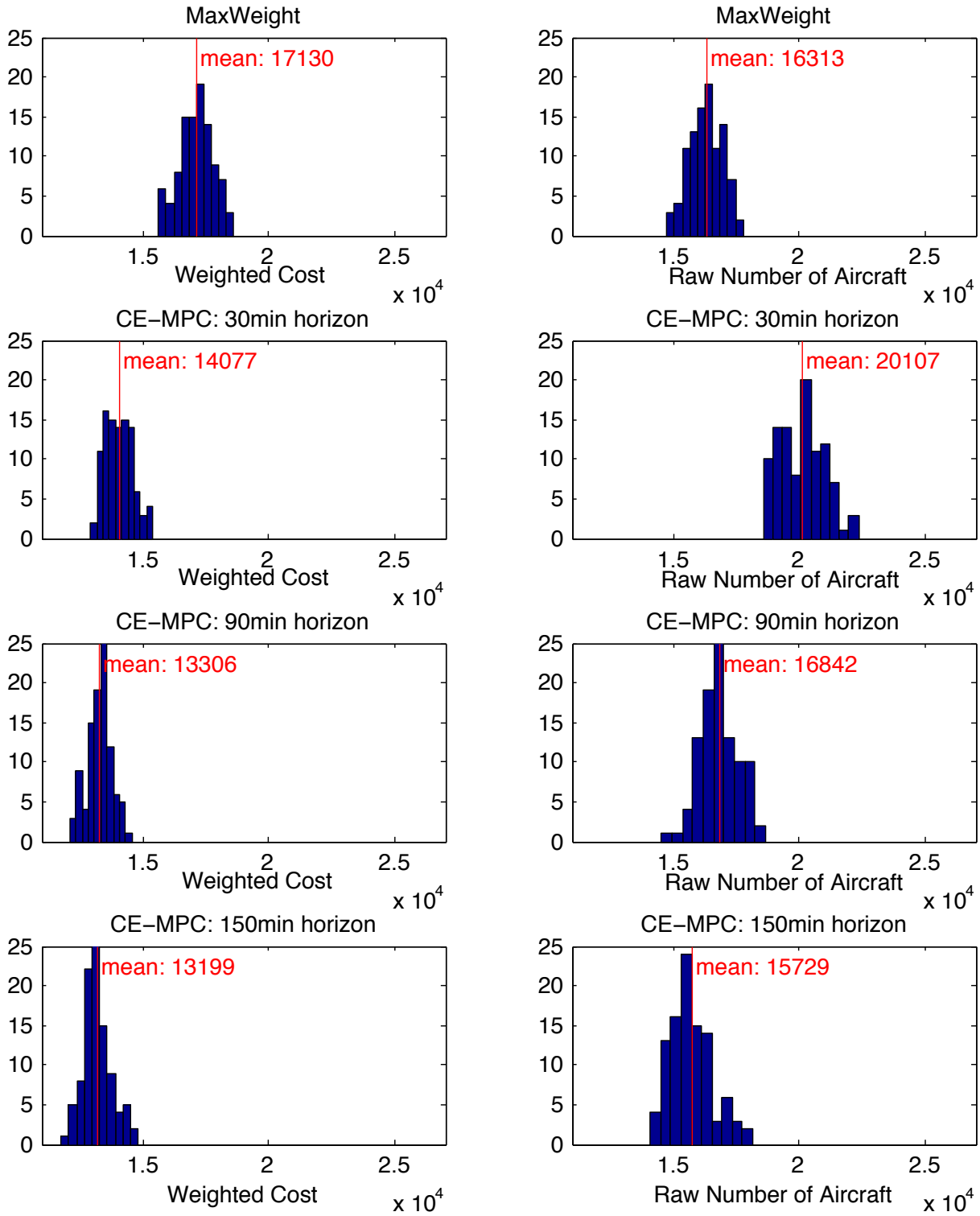

Figure 7. Simulation results as for Figure 6, for a scenario with no reduction in capacity (no weather event). With the reduced uncertainty, increasing the MPC planning horizon from 30 min to 1.5 hour improves the performance more significantly for the weighted objective compared to Figure 6.