Control and Optimization of Air Traffic Networks

Karthik Gopalakrishnan & Hamsa Balakrishnan

Department of Aeronautics and Astronautics,
Massachusetts Institute of Technology,
Cambridge, MA 02139, USA
email: karthikg@mit.edu, hamsa@mit.edu

Annu. Rev. Control Robot. Auton. Syst. 2021. 4:1–28

https://doi.org/10.1146/annurev-control-070720-080844

Keywords

air transportation, flight delays, network models, Markov jump linear systems, optimal control

Abstract

The air transportation system connects the world through the transport of goods and people. However, operational inefficiencies such as flight delays and cancellations are prevalent, resulting in economic and environmental impacts.

In the first part of this article, we review recent advances in using network analysis techniques to model the interdependencies observed in the air transportation system, and to understand the role of airports in connecting populations, serving air traffic demand, and spreading delays. In the second part of the article, we present some of our recent work on using operational data to build dynamical system models of air traffic delay networks. We show that Markov Jump Linear System (MJLS) models capture many of the salient characteristics of these networked systems. We illustrate how these models can be validated, and then used to analyze system properties such as stability, and to design optimal control strategies that limit the propagation of disruptions in air traffic networks.

1. INTRODUCTION

Air transport connects the world: More than 4.5 billion passengers, and trade estimated to be valued at $6.7 trillion, were transported on 39 million scheduled flights in 2019 (1). Air traffic demand has grown for the greater part of the past two decades, as has its connectivity, which has nearly doubled in that time to more than 22,000 unique city-pairs worldwide (1). The growth in air traffic has been accompanied by an increase in congestion, and consequently, delays. In 2019, 19% of the flights in the United States (US) had an arrival delay of greater than 15 minutes, and another 2.5% of flights were cancelled. Flight delays are estimated to have an economic impact $30-40 billion each year in the US, and $60 billion worldwide (2, 3, 4, 5).

FAA: The Federal Aviation Administration is the government agency that provides air traffic control services in the US

NAS: The National Airspace System, refers to the airports and airspace regions in the US that are controlled by the FAA

Primary delay: The delay caused by the initial disturbance to an aircraft's schedule

Reactionary delay: The delay caused by the knock-on effects of a primary delay, caused when the delay propagates to subsequent legs of an aircraft's itinerary

Flight delays and cancellations can be initiated by a number of factors, including over-scheduling during peak hours of operation, reductions in airport and airspace capacity due to inclement weather (e.g., fog or thunderstorms), mechanical and maintenance issues on aircraft, unavailability of crew members due to illness, extreme weather events (e.g., hurricanes), airport or airspace closures (6, 7), and even power or computer outages at an airline (8, 9). While the above disruptions initiate primary delays at an airport, these delays propagate to other airports, with potentially far-reaching impacts (10).

The networked nature of the air transportation system enables a high-degree of connectivity between cities around the world, but can also cause brief disruptions at an airport to cascade throughout the system. For example, two of the most disruptive events in 2016 were an hour-long computer outage at Southwest Airlines in Dallas, and a 5-hour-long power outage (due to a fire) at Delta Airlines in Atlanta (8, 9). The impacts of these outages were system-wide, lasted several days, and have been estimated to cost Southwest and Delta Airlines, $177 and $150 million, respectively (9). It is worth noting that approximately 40% of flights in both the US and Europe are delayed not because of a direct impact, but because the aircraft is late arriving from a previous leg of its itinerary (11, 12).

Any investigation of air traffic delays should explicitly consider the network structure and its role in the dynamics of delay propagation. By aiding traffic managers, airlines, and transportation system engineers in their strategic and tactical decision-making, such a study would enable efficient, robust, safe, and environmentally-friendly air traffic operations.

1.1. Challenges in Air Traffic Networks

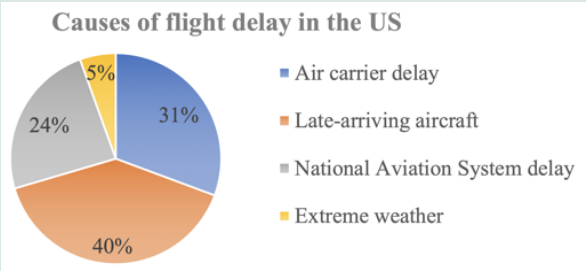
Networks are a powerful abstraction that have been successfully used to model the dynamics of a wide range of systems (14), from disease epidemics (15) and rumor propagation (16), to engineered systems such as power grids (17, 18), the Internet (19), roads (20), public transport (21), railroads (22) and air transportation (23). While each of these applications has unique characteristics, we highlight some of the major challenges that arise in the context of air traffic delay networks:

Coupling of nodal states and network topology: Air traffic networks exhibit a strong coupling between the dynamics of the nodal states and the network topology. The network topology changes with time (e.g., due to flight schedules) and is also a function of the state of the nodes of the network (e.g., airport delays). Such complex intertwining of the nodal states and the underlying topology poses technical challenges to the modeling and analysis of these systems.

Multilayer networks: Connectivity in the air transportation system has been traditionally modeled only in terms of operations, that is, flight service between an origin and a

What causes flight delays?

Data from the US Bureau of Transportation Statistics on the different causes of flight delays is presented below (11).



Late-arriving aircraft. A single aircraft flies multiple flight legs each day in order to increase its utilization. Aircraft in the US typically operate 4-6 flight legs a day; some aircraft fly as many as 10 flights in a single day (13). Although airline schedules are planned with buffers between flight legs to mitigate delay propagation, the delay of the incoming aircraft may exceed the buffer, resulting in reactionary delays.

Air carrier delays. This category includes all causes of delay that are considered to be within the control of an airline, such as maintenance or flight crew issues, passenger boarding, fueling, baggage loading, aircraft cleaning, etc.

NAS delays. These are delays due to air traffic control and traffic management initiatives, for example, to address reduced capacity due to inclement weather or heavy traffic volume, at an airport or in the airspace.

Extreme weather. This category includes severe meteorological conditions such as hurricanes, tornadoes, or blizzards, that prevent the safe operation of flights.

destination (24, 25, 26, 27). However, in reality, the air transportation system is *multi-layered*: the nodes (airports) may interact along multiple dimensions, each of which can be treated as a layer of the network (28). For example, one can represent traffic flows on one layer of the network, airport capacities on another layer, and flight delays on a third layer of the network. These layers would in turn be connected; for instance, bad weather can lead to a reduction in airport capacities, which can in turn increase the delays on the network. High delays can lead to network-wide cancellations, thereby modifying the traffic on the network. In other words, the air transportation system is a multilayer network over which delays and other disruptions emerge and spread. Limited methodological tools exist for developing analytical, or even data-driven, models for such real-world networks.

Complex and variable dynamics: Air traffic networks exhibit significant variability (e.g., in weather patterns and airline schedules) which have seasonal, weekly, and daily trends; furthermore, weather phenomena are uncertain. As a consequence, it is difficult to

analyze and model the dynamics of air traffic delays even when operational data is available, since one needs to be able to control for the changes in capacity and demand.

Airline competition: Constrained air traffic network resources are shared by aircraft operated by multiple competing airlines, each with its own utility function. These utility functions govern their scheduling and operating practices, and therefore, the dynamics of the overall system.

Data quality and privacy: Information and data sharing are a key part of any modeling, analysis, or optimization algorithms. In data-driven analysis, it is important to note that some data (e.g., the ascribing of delay causes) can be subjective, resulting in potential biases in the conclusions. Some data elements pertaining to airline operating practices and cost functions are proprietary, and the flight plans of certain aircraft may be held private, resulting in only partial information on some system states.

1.2. Outline of the Paper

In Section 2, we present an overview of the advances in the analysis of both static and dynamic aviation networks. In Section 3, we present a Markov Jump Linear System (MJLS) model of airport delays, and describe its general applicability to networked systems with switching topologies. Section 4 presents results on the stability of MJLS models of air traffic delay networks, and Section 5 presents new results on optimal control for MJLS models. Finally, Section 6 concludes with promising directions for future research.

While this paper adopts a network-centric perspective on air transportation, it is important to note that optimization and control techniques have long been an integral part of air traffic management. Successful applications of such techniques include airport surface congestion management (29, 30, 31), traffic flow management through large-scale optimization (32, 33), tactical conflict resolution (34, 35), equitable resource allocation (36, 37), and airline operations scheduling (38, 39). Developing practical, data-driven optimization and control algorithms to manage large-scale network operations will be essential to ensure safety and to enable growth in manned and unmanned air traffic operations over the coming years.

2. STRUCTURE AND EVOLUTION OF AVIATION NETWORKS

Research on network models of air transportation can be broadly classified based on the quantity of interest (e.g., flight connectivity or operational performance), the scope of the network (e.g., limited to a specific geography or airline), and temporal behavior (static or dynamic network) (23). Most prior studies have focused on static flight connectivity networks, with airports are nodes, and edges represent the number of flights or the number of passengers between pairs of airports (25, 26, 27). Prior work has also considered air traffic in different parts of the world, as well as passenger and cargo airlines (40, 41, 42). There has been limited consideration of the dynamics of delays, cancellations, airport congestion, and airspace capacity in air traffic networks (Figure 1).

We first discuss centrality in the context of static networks, and then extend these concepts to the case of dynamic networks, focusing on applications to air traffic delays.

Static network: A network whose nodes and edges, including the weights and directions, do not change over time

Dynamic network: A network whose nodes or edges (weights or directions) vary with time

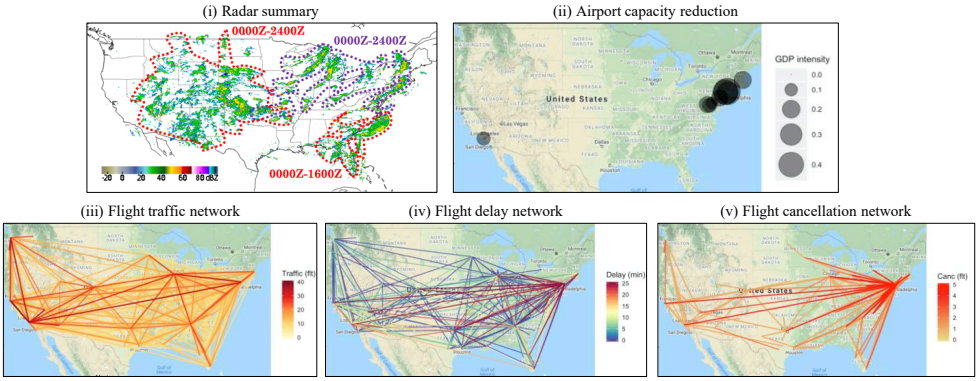


Figure 1

Different representations of the state of the air transportation system on July 1, 2016. Although networks in (iii)-(v) are directed, an average undirected network is shown for ease of visualization.

2.1. Node Centrality

Static network analysis of the air transportation system has yielded insights into its scale-free nature (23, 43), airline-specific patterns (40, 43), and its robustness to disruptions (44, 45, 46).

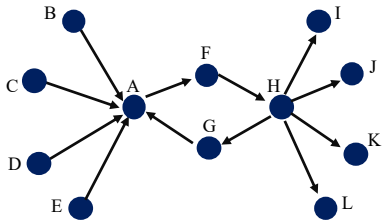
A key aspect of analyzing network structure is the quantification of the centrality of a node in a graph. We define a network (or graph) by the set of V nodes, and the edges between these nodes. The directionality and weight of edges can be represented using an adjacency matrix $A \in \mathbb{R}^{V \times V}$, where the element a_{ij} represents the weight on the edge from node i to node j. In the air traffic context, the edge (i, j) is sometimes referred to as an *OD pair* (Origin-Destination pair). The entry $a_{ij} = 0$ if there is no edge from node i to node j. The centrality, or importance of a node in a network, is context-dependant, and can be evaluated in several ways (14, 47). We describe three measures of node centrality:

Undirected network: A network whose adjacency matrix is symmetric, i.e., $A = A^T$

Unweighted network: A network whose adjacency matrix elements are binary, i.e., 0 or 1

1. Degree centrality: For a node i, the in-degree (d_i^{in}) and the out-degree (d_i^{out}) are defined as $d_i^{in} = \sum_j a_{ji}$ and $d_i^{out} = \sum_j a_{ij}$, respectively.
2. Eigenvector centrality (or eigencentrality): While the degree centrality measures the strength of association of a node with its immediate neighbors, the eigencentrality reflects the importance of a node in the context of the entire network. The eigencentrality for all nodes of an undirected network with (symmetric) adjacency matrix A_{sym} are given by the elements of the principal eigenvector (i.e., the eigenvector corresponding to the eigenvalue with the largest magnitude) of A_{sym}.
3. Hub and authority scores (48, 49, 50): The generalization of eigencentrality to directed graphs considers the importance of the inbound and outbound connections of network nodes using metrics called the hub score and authority score. These scores are computed as the principal eigenvector of AA^T and A^TA, respectively (Figure 2).

While flight connectivity networks are often represented as unweighted undirected networks, delays or cancellation networks exhibit asymmetric relationships between airports. As a result, they are better modeled as weighted, directed networks, with the edge weights representing the average flight delays (or cancellations) on a particular route. Such networks



Node(s)	In-degree	Out-degree	Hub score	Authority score
A	4	2	0	0.71
B-E	0	1	0.32	0
F	1	1	0	0
G	1	1	0.32	0.32
H	2	4	0.71	0
I-L	1	0	0	0.32

Figure 2

(Left) An example of a network, and (right) centrality metrics for its nodes.

have only recently received attention (51).

2.2. Clustering of Air Traffic Networks

Clustering has been used in the aviation context to categorize operational data in terms of flight connectivity, delays, traffic, trajectories, weather impacts, etc. Most of these efforts have involved clustering high-dimensional data (e.g., weather (52), airport capacities (53, 54, 55), or flight trajectories (56)). Their scope has been restricted to either one airport or a small group of airports, and has not considered network interactions. By contrast, we focus on network representations of operational performance, specifically delays and cancellations, and consider the problem of graph clustering (57, 51).It is worth noting the distinction between the node-clustering problem (i.e the community detection problem), and the problem of graph-clustering problem. The former (58, 59, 60, 61, 62, 51) categorizes groups of nodes in a given graph that are similar to each other, while the latter aims to identify entire graphs that are similar to each other.

The clustering of graphs can be carried out defining a feature vector, denoted $\mathbf{f}_{\mathcal{G}}$, for each graph $\mathcal{G}$, and subsequently performing a vector clustering technique (e.g., k-means clustering (63)) on these feature vectors. The degree centrality or eigencentrality can be used as the feature vector $\mathbf{f}_{\mathcal{G}}$ (59, 64). However, degree centrality does not consider global measures of centrality, while eigencentrality is only suitable for undirected graphs. One could also populate $\mathbf{f}_{\mathcal{G}}$ with the edge weights a_{ij}; however the feature vector then loses the network-centric context, and its length grows super-linearly ($\mathcal{O}(V^2)$) with the size of the network. These challenges in clustering weighted, directed networks were addressed in (51) by using a scaled hub/authority score feature vector given by :

$$\mathbf{f}_{\mathcal{G}} = \sum_{i,j} a_{ij} \begin{pmatrix} \mathbf{h} \\ \mathbf{a} \end{pmatrix} \qquad 1.$$

This feature vector accounts for network-centric features and the magnitude of the edge weights while clustering weighted, directed graphs. It can therefore help distinguish between two delay networks based on the magnitude and direction of the delays, as well as the importance of the corresponding airports in the context of the entire network.

Delay networks evolve with time. It is therefore valuable to identify characteristic network topologies. These typical or characteristic topologies can help identify anomalous network structures, comparing network topologies, and understand the evolution of the

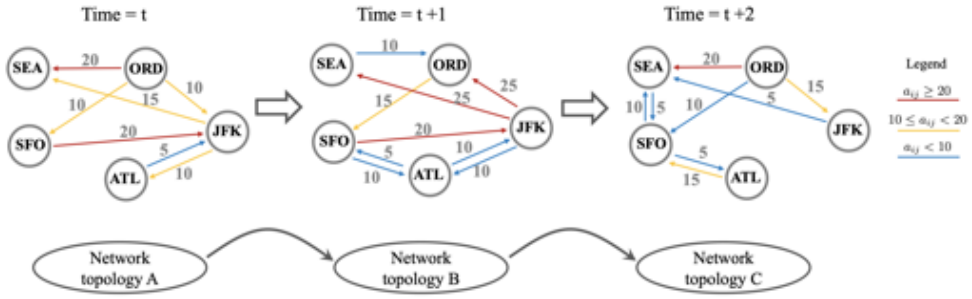


Figure 3

(Top) An example of a time-series of weighted-directed networks. (Bottom) Clustering can provide an equivalent, high-level representation of the evolution of the network.

system. Figure 3 shows a time-series of weighted directed graphs (representing delay networks), and how clustering could model the evolution of the system. This approach of using clustering to analyzing the evolution of air transportation networks will be further elaborated in Section 3.

2.3. Temporal Evolution of Air Traffic Networks

The temporal evolution of a networked system is associated with two distinct phenomena. The first, widely-studied, phenomenon is the spreading process on the nodes of a network (e.g., epidemic propagation in a population network). The second phenomenon is the evolution of the structure of the network itself, namely, changes in the number of nodes, or strength and direction of edges. Figure 3 illustrates one such example, in which the edge weights, which could represent flight delays between a pair of airports, changes with time.

Models of flight delays help characterize the temporal evolution of the air traffic delay network. Two modelling approaches are popular in this context: queuing theory and machine learning. Queuing models track aircraft as they pass through constrained resources, modeled as queues (65, 66, 67). However, they often require proprietary information on crew itineraries, and are sensitive to airport and airspace capacity estimates (68, 69, 70). Recent work on machine learning methods has significantly improved delay prediction accuracy (71, 72, 73), but lack the analytical tractability and interpretability needed for resilience and robustness analysis (74, 75).

We seek to develop a data-driven model of air traffic delay network dynamics that is analytically tractable and accounts for time-varying network topologies as well as weighted, directed graphs. While certain aspects of this problem, such as asymmetric interactions (76) and switching topologies (77), have been previously considered, we present our recent work on developing and validating a data-driven model of airport delays (78).

3. NETWORK MODEL OF AIRPORT DELAYS

We present a Markov Jump Linear System (MJLS) model for the hourly progression of airport delays (78). MJLS are a subset of hybrid systems (79), which have been successfully used to model electromechanical systems (80), epidemic propagation (81), communication

networks (82), and multi-agent interactions (83). The proposed model is constructed using operational flight delay data from the US, incorporates the time-varying network effects that drive delay propagation, and is lends itself to further analysis (Section 4 and 5).

3.1. Airport Delay Dynamics Model

3.1.1. Data and setup. An air traffic delay network comprises of nodes corresponding to airports, and weighted, directed edges corresponding to the magnitude of flight delays between the two airports. We define a delay network for each hour of the day, with the edge weight being the median departure delay for all flights that were scheduled to take off on that route during the hour. We use data for domestic air traffic in the US during the years 2011-12 to construct these time-series of weighted-directed networks (11). We consider only the 30 major airports in the US (FAA Core 30); thus the number of nodes $V = 30$. A network for each hour in the two-year period results in $731 \times 24 = 17,544$ networks, each with a distinct $V \times V$ adjacency matrix, $A(t)$. We denote the total inbound and outbound delays at an an airport i at hour t as $d_i^{\text{in}}(t)$ and $d_i^{\text{out}}(t)$ respectively. These airport delays are related to the elements of the adjacency matrix $A(t) = [a_{ij}(t)]$ as $d_i^{\text{in}}(t) = \sum_j a_{ij}(t)$ and $d_i^{\text{out}}(t) = \sum_j a_{ij}(t)$. Our objective is to develop a model for the evolution of $d_i^{\text{in}}(t)$ and $d_i^{\text{out}}(t)$. With regards to notation, we use boldface to denote vectors (e.g., $\mathbf{x}$(t) for the state vector, $\mathbf{0}$ for a vector of zeros, etc.), $\mathbb{R}$ for the set of reals, $\mathbb{N}$ for the set of natural numbers, $\mathbb{1}_{\text{condition}}$ as an indicator variable, and $\text{I}_{M\times M}$ to denote a $M \times M$ identity matrix.

Node state: The vector $\mathbf{x}(t)$, also referred to as the continuous state

Discrete mode: The current mode $m(t)$ belonging to one of the M possible options, also referred to as the mode or topology.

State of the MJLS: Combination of the node state $\mathbf{x}(t)$ and the discrete mode $m(t)$.

3.1.2. Topology-dependent model. The total outbound delay at airport i at time-step $t+1$ depends on two components: the outbound delay at that airport at time t (reflecting delay-persistence) and the delay bound to i from the other airports at time t (modeling the network effect). Similarly, the outbound delay at airport i at $t+1$ depends on its outbound delay at time t as well as the network effect of other connected airports. Combining these two factors linearly using proportionality constants α_i^{out}, α_i^{in}, β_i^{out}, and β_i^{in}, we get:

$$d_i^{\text{out}}(t+1) = \alpha_i^{\text{out}} d_i^{\text{out}}(t) + \beta_i^{\text{out}} \sum_{j=1}^{V} \bar{a}_{ij}(t) d_j^{\text{in}}(t) \tag{2.}$$

$$d_i^{\text{in}}(t+1) = \alpha_i^{\text{in}} d_i^{\text{in}}(t) + \beta_i^{\text{in}} \sum_{j=1}^{V} \bar{a}_{ji}(t) d_j^{\text{out}}(t) \tag{3.}$$

where $\bar{a}_{ij}(t)$ represent the elements of the row normalized adjacency matrix $\bar{A}(t)$ (i.e., $\bar{a}_{ij}(t)$ is the fraction of the total outbound delay at node i that is destined for node j). The constants α and β, are assumed to be non-negative, and their magnitudes reflect the relative importance of the persistence and the network effect in the delay dynamics. These constants do not necessarily sum up to 1, since delay may not be conserved due to factors such as buffers in flight schedules (which help attenuate delays) or high airport connectivity (which could exacerbate delays). We denote the node-state of the system at any time t as the vector $\mathbf{x}(t)$ which comprises of all the inbound and outbound delays at all airports:

$$\mathbf{x}(t) = \begin{bmatrix} \mathbf{d}^{\text{out}}(t) \\ \mathbf{d}^{\text{in}}(t) \end{bmatrix}. \tag{4.}$$

Using $\mathbf{x}(t) \in \mathbb{R}_{\geq 0}^{N\times 1}$, where $N = 2V = 60$, Equations 2 and 3 are expressed succinctly as

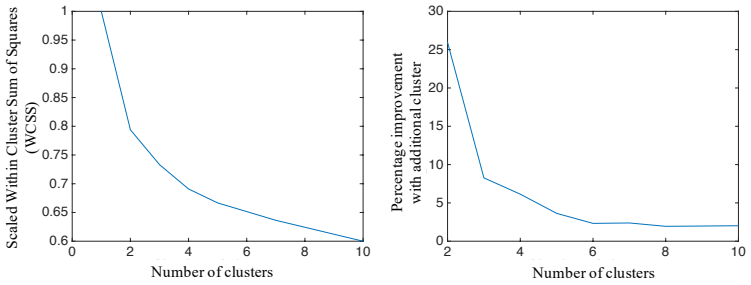


Figure 4

Plot of the within-cluster sum of squares (WCSS) distance as a function of the number of clusters (left), and the percent reduction in WCSS with each additional cluster (right). We use $k = 6$.

$$\mathbf{x}(t+1) \;=\; \Big(\mathrm{diag}([\alpha^{\mathrm{out}};\alpha^{\mathrm{in}}]) + \mathrm{diag}([\beta^{\mathrm{out}};\beta^{\mathrm{in}}])\mathcal{A}(t)\Big)\,\mathbf{x}(t) \;=\; \Gamma(t)\,\mathbf{x}(t), \qquad 5.$$

where $\mathcal{A}(t) = \begin{pmatrix} 0 & \bar{A}(t) \\ \bar{A}^{\mathrm{T}}(t) & 0 \end{pmatrix}$. Thus, Equation 5 describes the dynamics of the continuous-state $\mathbf{x}(t)$ as a function of the topology $A(t)$.

3.1.3. Clustering of network topologies. Since the evolution of the delays depends on the network topology $A(t)$ which can take many possible values, we aim to find a set of characteristic network topologies. In other words, we want to identify a limited set of matrices that are representative of $\mathcal{A}(t)$. Incidentally, the principal eigenvector of $\mathcal{A}(t)$ is the vector comprising of the hub-and authority scores of a network with adjacency matrix $\bar{A}(t)$ (48). We emphasize a subtle observation: Equation 5 is such that a centrality measure on static networks ends up being relevant in identifying characteristic patterns that govern the temporal dynamics of the system. A weighted hub-authority clustering (based on Equation 1) is subsequently performed to identify six representative network topologies (Figure 4), which are further split based on whether the overall system delay is increasing or decreasing at that time. The $M = 12$ network topologies, $\{A_1, A_2, \ldots, A_M\}$, are shown in Figure 5.

3.1.4. Parameter estimation. The parameters α and β are estimated for each discrete mode (network topology) using a linear regression. These parameters are combined with the topologies $\{A_1, \ldots, A_M\}$ via Equation 5 to identify the system dynamics matrices $\{\Gamma_1, \Gamma_2, \ldots, \Gamma_M\}$. The transitions in the network structure are assumed to be a result of Markovian jumps between $M = 12$ possible topologies. In other words, the probability of transitioning from mode $m(t)$ at time t to $m(t+1)$ at time $t+1$, $\mathbb{P}[m(t+1) = j | m(t) = i]$ is denoted $\pi_{ij}(t)$. The transition probability $\pi_{ij}(t)$ (and the corresponding $M \times M$ transition matrix Π_t) is obtained using the maximum likelihood estimator, which in this case is equal to the empirically observed frequency of transitions from mode i to j at time t. The discrete mode transitions may also be associated with resets or jumps in the state of the node. In other words, any transitions from topology i to topology j modify the node state as $\mathbf{x}(t+1) = J_{i,j}\Gamma_i\mathbf{x}(t)$. The state reset matrix $J_{i,j}$ is also estimated by linear regression.

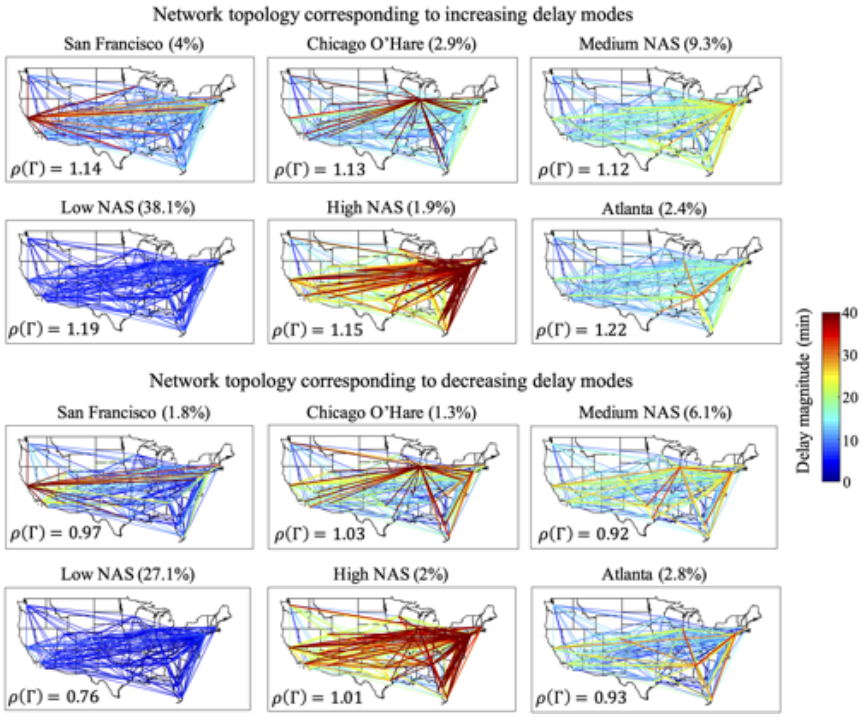


Figure 5

Characteristic network topologies (discrete modes) from 2011-12, including delay trends (increasing or decreasing). The modes are named on a qualitative basis, and the frequency of occurrence is shown in parentheses. The spectral radius (magnitude of the largest eigenvalue) of the corresponding system dynamics matrix, $\rho(\Gamma)$ is also presented.

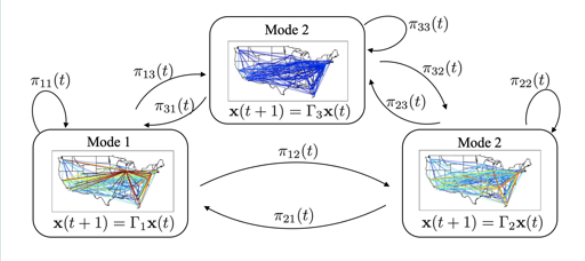
3.2. Validation and Model Performance

We consider the MJLS model whose parameters α, β, and J are estimated using data from 2011. Figure 6 highlights the potential for the model to reflect high-level temporal trends in delay dynamics. For this experiment, the MJLS model is initialized with the airport delays at 8 am EST, and the delays are propagated using Equations 6 and 7. We see that the MJLS model is able to capture the varying scale and magnitude of delays across different airports, as well as the temporal trend of increasing and decreasing delays (with a 6.4% error over the first 12 hours). There is a 1- to 2-hour lag in the ability of the model to capture decaying delays, possibly because the model, for simplicity, has just $M = 12$ discrete modes. Although not shown in this figure, the model is also found to track the variance in delay reasonably well.

The predictive capability of the MJLS model is worth noting (86). Table 1 compares a linear regression and the MJLS models for predicting delays at a 2-, 4-, and 6-hour horizons. The models use different inputs: the MJLS model uses $\mathbf{x}(t)$ and $m(t)$ as the initial condition, while the linear regression models use t, $\mathbf{x}(t)$, and a characteristic network topology label for the previous day (for potential residual delay effects). The results indicate that the MJLS model performs better than linear regression (as well as simple neural network architectures)

Markov Jump Linear System (MJLS) models

The dynamics of the inbound and outbound airport delays is described using a discrete-time switched linear system, where the evolution of the airport delays in each discrete mode is linear, and the discrete mode transitions are governed by a Markov process (84, 85). An example of a MJLS system with three discrete network topologies is shown below.



In general, the dynamics of the airport delays $\mathbf{x}(t)$ from mode $m(t)$ at time t is given by:

$$\mathbf{x}(t+1) = J_{m(t),m(t+1)}\Gamma_{m(t)}\mathbf{x}(t) \quad 6.$$

$$\pi_{ij}(t) = \mathbb{P}[m(t+1) = j | m(t) = i] \quad 7.$$

$$\text{Initial conditions} : \mathbf{x}(0) \geq \mathbf{0} \text{ and } m(0) \quad 8.$$

where the state reset matrix $J_{m(t),m(t+1)} = \mathrm{I}_{N\times N}$ if $m(t) = m(t+1)$.

Positive MJLS (PMJLS). A PMJLS is a MJLS for which the state always remains non-negative for any non-negative initial condition, i.e., $\mathbf{x}(0) \geq \mathbf{0} \implies \mathbf{x}(t) \geq \mathbf{0}\ \forall t$. For airport delay dynamics, all the elements of Γ_i and $J_{i,j}$ are non-negative and thus the system is a PMJLS, consistent with the fact that airport delays are always non-negative.

Method	Mean Prediction error (min)		
	2 hour	4 hour	6 hour
Linear Regression	7.59	9.51	10.47
MJLS model	6.75	8.83	10.09

Table 1 Mean prediction errors for the total system delay for the MJLS and a linear regression model across different prediction horizons (86). The model was developed using data from 2011, and tested on data from 2012.

while modelling airport delay propagation (86).

3.3. Other Applications of Modeling Framework

The proposed framework (Figure 7) yields a switched-system model for the temporal evolution of a continuous nodal state vector. In our application, the continuous state corresponded to airport delays, but it could be extended to packet latency at routers in a communication network, or the extent of gossip spread in a social network. In each of these settings, one could express a parametric, topology-dependant model for the nodal state, re-

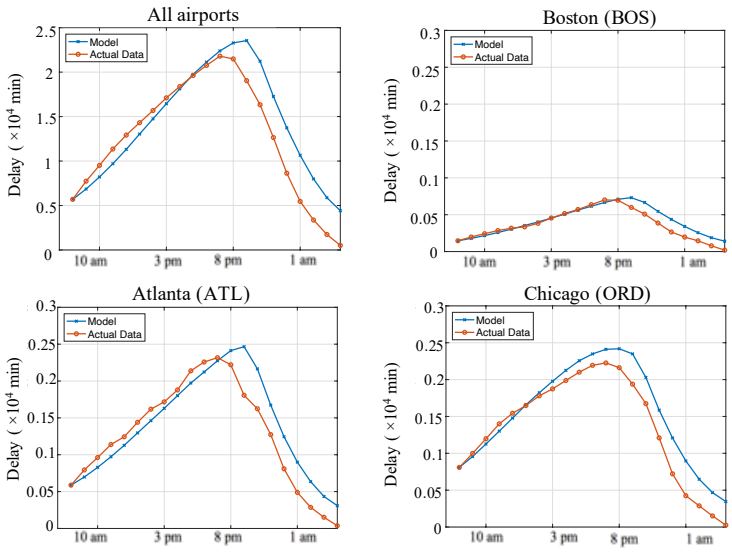


Figure 6

Comparison of actual and MJLS-predicted mean delays (sum of inbound and outbound delays) evaluated for 2011. All times are in US Eastern Standard Time (EST).

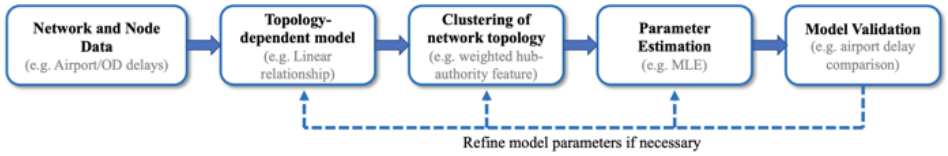


Figure 7

Overview of the steps involved in developing a switched-system network model from data. The text in parenthesis describes the adaptations of the steps for the airport delay model.

duce the model complexity by restricting the topology to a finite set by clustering (87), learn the model parameters by regression, and iteratively refine the parameters if needed. More generally, the model need not be linear or the topology transitions Markovian, presenting great flexibility.

4. STABILITY OF AIRPORT DELAY DYNAMICS

Stability analysis provides insights into the temporal and topology-specific patterns that lead to increasing or decreasing delays, and also helps further validate the model. For example, one would expect airport delays to decay overnight, as traffic demand decreases. The stability, or instability, of a system may also motivate approaches to effectively control the system, as we will explore further in Section 5.

For a discrete-time, linear, time-invariant (LTI) system, stability is defined as the convergence of the sequence of the continuous state norms to 0, and a necessary and sufficient

for stability is that the spectral radius of the system matrix is less than 1. The evolution of $\mathbf{x}(t)$ in a MJLS is a stochastic process, and consequently, relevant notions of stability have been proposed for switched systems in general (88, 89), and MJLS in particular (90, 91, 92). Stability conditions for MJLS are starkly different from those for LTI systems– stability or instability of the individual modes is neither necessary nor sufficient for the stability of the MJLS (91). In this section, we present conditions for the stability of PMJLS with continuous state resets, and interpret them in the context of air traffic delay networks.

Stability of discrete-time positive Markov Jump Linear Systems (PMJLS)

Definitions

Mean Stability (MS): A system is said to be *mean stable* if the expected value of the state tends to zero as time tends to infinity, i.e, $\lim_{k\to\infty} \mathbb{E}[\mathbf{x}(k)] = \mathbf{0}$, for any non-negative initial conditions, $\mathbf{x}(0)$, and for any $m(0)$.

Exponentially Mean Stability: A system is said to be *exponentially mean stable* if there exist positive scalars c and $r < 1$, such that $\mathbb{E}[\mathbf{x}(k)] \leq c\, r^k \|\mathbf{x}(0)\|\mathbf{1}$, for all times k, for any non-negative initial conditions, $\mathbf{x}(0)$, and for any $m(0)$.

Exponential δ-Moment Stability: A system is said to be *exponentially δ-moment stable* if there exist positive scalars c and $r < 1$ such that $\mathbb{E}[\|\mathbf{x}(k)\|^\delta] \leq c\, r^k \|\mathbf{x}(0)\|^\delta$, for all times k, for any non-negative initial conditions, $\mathbf{x}(0)$, and for any $m(0)$. When $\delta = 2$, it is known as *exponential mean-square stability*.

Almost Sure Stability (ASS): A system is said to be *almost surely stable* if the state tends to zero as time tends to infinity, with probability 1, i.e., $\mathbb{P}[\lim_{k\to\infty} \|\mathbf{x}(k)\| = 0] = 1$, for any non-negative initial conditions, $\mathbf{x}(0)$, and for any $m(0)$.

Remarks:

1. The notion of MS was specifically introduced to exploit the non-negativity property of PMJLS, in order to obtain stability criteria (93, 94).
2. The concept of ASS requires that *every* realization of the system state tends to zero, rather than just tending to zero in expectation. This definition is typically more useful in practice (95).

Relationships between the different notions of stability

For a PMJLS, these notions of stability are related as follows (93, 78):

$$\text{Mean Stable} \iff \text{Exponentially Mean Stable} \iff \text{Exponentially 1-Moment Stable} \Rightarrow \text{Almost Surely Stable}$$

While sufficient conditions, with varying degrees of conservatism, have been proposed for ASS, a necessary *and* sufficient condition remains elusive (96). As a result, a proof of MS is typically used as a sufficient condition for ASS.

4.1. Mean Stability of a PMJLS

We begin with the mathematical preliminaries needed for our main result on necessary and sufficient conditions for the mean stability (MS) of a PMJLS. We ignore the state reset matrices $J_{i,j}$ in our analysis for simplicity; however, the results can be trivially extended to incorporate them (78). The key idea involved is to define and track an augmented state vector $\mathbf{q}(t) \in \mathbb{R}^{NM\times 1}$ given by

$$\mathbf{q}(t) = \Big[\, \mathbf{q}_1(t),\ \cdots,\ \mathbf{q}_M(t) \,\Big]^\intercal, \text{ where } \mathbf{q}_i(t) = \mathbb{E}[\mathbf{x}(t)\,\mathbb{1}_{m(t)=i}]. \tag{9.}$$

Next, matrices $B_t \in \mathbb{R}^{NM\times NM}$ and $S = [s_{ij}] \in \mathbb{R}^{N\times NM}$ are defined as

$$B_t = \begin{bmatrix} \pi_{11}(t)\Gamma_1 & \dots & \pi_{M1}(t)\Gamma_M \\ \vdots & \ddots & \vdots \\ \pi_{1M}(t)\Gamma_1 & \dots & \pi_{MM}(t)\Gamma_M \end{bmatrix} \text{ and } s_{ij} = \begin{cases} 1, \text{ if } j = i + kN, \text{ for } k \in \mathbb{N} \\ 0, \text{ otherwise} \end{cases} \tag{10.}$$

Lastly, we define $C_{t_1,t_2} \in \mathbb{R}^{N\times NM}$ as

$$C_{t_1,t_2} = S(\mathrm{I}_{MN\times MN} + B_{t_1} + B_{t_1+1}B_{t_1} + \ldots + B_{t_2-1}\cdots B_{t_1}), \tag{11.}$$

where $C_{t_1,t_2} = [\mathbf{c}_{ij}^T]$, with $\mathbf{c}_{ij} \in \mathbb{R}^{N\times 1}$, $i = 1,\ldots,N$, and $j = 1\ldots,M$.

The following propositions describe the evolution of $\mathbf{q}(t)$ and $\mathbb{E}[\mathbf{x}(t)]$:

Proposition 1. *Given* $\mathbf{x}(t)$ *and* $m(t)$, $\mathbf{q}_i(t+1) = \pi_{m(t),i}(t)\Gamma_{m(t)}\mathbf{x}(t)$.

Proposition 2. *The augmented state vector evolves as* $\mathbf{q}(t+1) = B_t\mathbf{q}(t)$.

Proposition 3. *The expected state* $\mathbb{E}[\mathbf{x}(t)]$ *is equal to* $\sum_{i=1}^{M}\mathbf{q}_i(t)$.

Proposition 4. *The expected sum of future states* $\sum_{t=t_1}^{t_2}\mathbb{E}[\mathbf{x}(t)]$ *is equal to* $C_{t_1,t_2}\mathbf{q}(t_1)$.

These propositions indicate that knowledge of the current state, either in the form of $(\mathbf{x}(t), m(t))$ or $\mathbf{q}(t)$ is sufficient to propagate the augmented state and the expected state $\mathbb{E}[\mathbf{x}(t)]$ for future time. We now present the main result:

MS: Mean stable

ASS: Almost surely stable

Theorem 1 (from (78))**.** *If the transition probabilities are time-varying and periodic with length* K*, i.e.,* $\pi_{ij}(t+K) = \pi_{ij}(t)$*, then the PMJLS is Mean Stable if and only if the spectral radius of the matrix* $D_k = \mathcal{B}_{k+K-1}\mathcal{B}_{k+K-2}\cdots\mathcal{B}_k$ *is less than 1, for some* $k \in [0, K]$.

Theorem 1 prescribes the conditions required to test for MS when transition matrices are time-varying and periodic. Since traffic patterns and flight delays are cyclic over the course of a day, the transition probability matrix Π_t is periodic with $K = 24$. The following corollary considers the case when transition matrices are not time-varying.

Corollary 1. *If* $\pi_{ij}(t) = \pi_{ij}\ \forall t$*, and* $B_t = B\ \forall t$*, then a necessary and sufficient condition for the Mean Stability of the PMJLS system is that the spectral radius of* B *is less than 1.*

Theorem 1 provides a relatively straightforward test for MS based on the spectral radius of D, $\rho(D)$. Additionally, since MS implies ASS for a PMJLS, mean stability implies that airport delays $\mathbf{x}(t)$ will converge to $\mathbf{0}$ with probability 1 as $t \to \infty$. It is also worth remembering that these are sufficient and not necessary conditions, i.e., if the conditions for MS are not satisfied, it does not tell us anything about ASS. However, for the case of airport delays, we expect the system to be both MS and ASS.

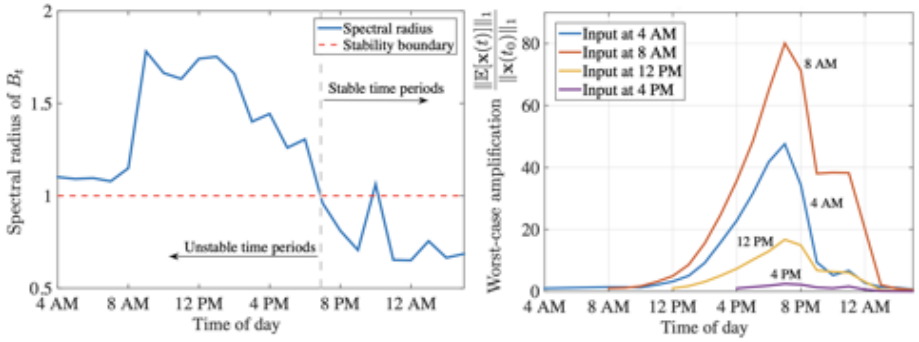


Figure 8

(Left) Worst-case delay magnification (over the next hour) of 1 unit of total delay initiated at a particular hour of the day. (Right) Worst-case amplification of 1-unit of time delay injected at four different times (4 am, 8 am, 12 pm and 4 pm), when allowed to propagate. All times are in US EST.

4.2. Implications for Airport Delay Dynamics

The airport delay MJLS model involves periodic transitions (with time period $K = 24$) between 12 discrete modes, some of which are stable, while others are unstable (Figure 5). To evaluate stability of the resulting PMJLS, we compute the spectral radius of D as $\rho(D) = 0.67 < 1$, to conclude that the system is MS, and therefore ASS. This finding is consistent with our expectation that delays will tend to decay over the course of a 24-hour period in expectation, and further validates the MJLS model. Interestingly, if we had not incorporated the time-varying transition matrix and instead considered the average matrix $\Pi_t = \Pi \; \forall t$, then $\rho(B) = 1.06$, and by Corollary 1 the system is would not be MS. This result highlights the critical role that the discrete mode transitions to 'Low NAS decreasing' (and other decaying delay modes) during the later part of the day play in stabilizing the system.

The spectral radius of B_t gives an upper bound on the amplification (i.e., the worst case magnification) of the total system delay at time t, over the next time-step (an hour, in our case). Figure 8, which plots $\rho(B_t)$ as a function of time indicates that the topology transitions are such that the delays tend to decay after 8 pm, thus resulting in low airport delays overnight. Additionally, from Proposition 2, we can also show that the worst case amplification of delays from t_1 to t_2, $\frac{\|\mathbb{E}[\mathbf{x}(t_2)]\|_1}{\|\mathbf{x}(t_1)\|_1}$, is equal to $\rho(B_{t_2-1} \dots B_{t_1+1} B_{t_1})$. Using this result, we plot the worst case amplification of a 1-unit total delay initialized at different points in time (4 am, 8 am, 12 pm and 4 pm) in Figure 8. Delays initiated earlier in the day have a greater potential to cascade throughout the system and get magnified (up to 80 times by 8 pm) as compared to those introduced during the later part of the day. Our analysis emphasizes the need to ensure disruption-free operations during the early part of the day for improved system performance – an operational practice regularly-employed by airline operations managers and air traffic controllers.

Our results illustrate the role that stability analysis plays in model validation and in identifying key aspects of the delay dynamics. Finally, we point readers towards recent work on structural (97) and finite-time (98) stability, which have yielded interesting insights.

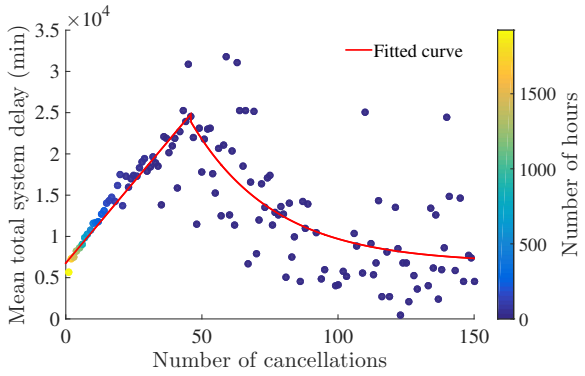


Figure 9

Mean system delay vs. number of flight cancellations, for hourly data from 2011-12 grouped by the number of flight cancellations. The color of each dot denotes the number of hours for each of the groups. While delays and cancellations both increase initially, cancellations are used to strategically reduce traffic demand when disruptions are severe, resulting in lower delays for flights that continue to operate.

5. OPTIMAL CONTROL OF AIR TRAFFIC DELAY NETWORKS

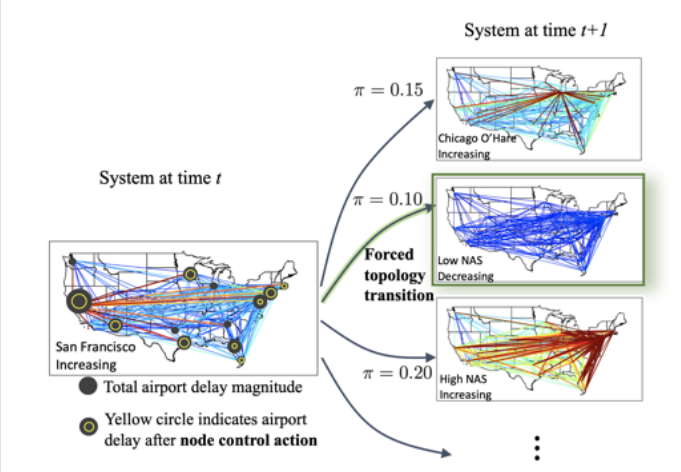
Long-term strategies for reducing air traffic delays require expensive infrastructure and technological investments. Tactical control approaches, at the time-frame of a few hours, such traffic flow management (e.g., ground delay programs or airspace flow programs resulting in flight swaps and cancellations) can be effective in mitigating congestion due to unforeseen disruptions (39). In the context of our MJLS model for air traffic delays, these actions correspond to either interventions on the nodal (continuous) state to decrease delays at airports, or interventions on the network topology to decrease the propagation of delays. Figure 9 shows an example of how flight cancellations are used to reduce demand, and consequently, reduce delays for the remaining flights. The development of control strategies for such networked systems will enable efficient, resilient and robust operations in a number of other critical infrastructures (99, 34, 100).

The control of processes on networks has been well-studied in the context of epidemiology (81, 101) and resilient infrastructures (102, 103, 104). However, controllers for systems with switching topologies have received less attention (91, 105). The most popular approach to control the nodal state of a MJLS is to use a mode-dependent state feedback controller (91, 106). Dynamic programming (107), convex programming (108), particle based approximate methods (109), and receding horizon controllers (110, 111) have also been developed. These controllers all modify the nodal state $\mathbf{x}(t)$ to achieve a desired control objective. The control of mode transitions, by which one alters the transition probability Π_t or induces specific mode transitions, has been impractical in many traditional settings, and therefore not been explored (112). However, increasing autonomy has enabled the ability to modify the structure of network interactions in many infrastructure systems. In this section, we develop controllers for a PMJLS that acts on both the continuous (node) state and the discrete mode, in order to minimize a desired objective function.

Control of Markov Jump Linear Systems

Let the continuous state of the system be $\mathbf{x}(t)$, and the discrete mode be $m(t)$ at time t. The objective of a controller is to optimally trade off the control effort with the benefits of a smaller continuous state $\mathbf{x}(t)$. Since $\mathbf{x}(t) \geq 0$ for a PMJLS, the continuous state magnitude can be represented by the 1-norm. The control action at time t minimizes the cost-to-go, and the objective function for the controller is expressed as

$$\text{Minimize} \quad \mathbb{E}[\sum_{\tau=t}^{T} \|x(\tau)\|_1] \; + \; \text{control penalty } (t) \tag{12.}$$



Node or continuous state control. The control action $\mathbf{u}(t) \in \mathbb{R}^{N\times 1}$ modifies the continuous state as $\mathbf{x}(t+1) = \Gamma_{m(t)}\mathbf{x}(t) - \mathbf{u}(t)$, while ensuring that $\mathbf{x}(t) \geq \mathbf{0}$ and $\mathbf{0} \leq \mathbf{u}(t) \leq U_{max}\mathbf{1}$. The action $\mathbf{u}(t)$ incurs a control penalty of $\beta^T(t)\mathbf{u}(t)$, where $\beta(t) \in \mathbb{R}^{N\times 1}_{\geq 0}$ is the cost per unit action on each node.

Topology or mode control. A mode control action forces a transition to a particular discrete mode at time $t+1$, instead of allowing it to be drawn from the distribution Π_t. The control penalty for forcing a transition from mode i at time t to mode j at $t+1$ is given by $\alpha_{i,j}(t) \in \mathbb{R}_{\geq 0}$. There is no penalty in the absence of a forced transition.

Integrated control. An integrated control action involves a combination of continuous state and mode control.

5.1. Optimal Control of a PMJLS

We present the optimal node, topology and integrated control actions that should be taken when the system has a continuous state $\mathbf{x}(t)$, and discrete mode $m(t)$, at time t, in order to minimize the objective function in Equation 12.

Theorem 2 (Node control)**.** *The node control action* $\mathbf{u}^*(t)$ *given by*

$$u_i^*(t) = \begin{cases} \max\{U_{max}, \sum_j [\Gamma_{m(t)}]_{ij} x_j(t)\} & \text{if } \beta_i(t) \le \eta_i \\ 0 & \text{otherwise} \end{cases} \quad 13.$$

where $\eta^T = \sum_{i,j} \pi_{m(t),j} \mathbf{c}_{ij}^T \in \mathbb{R}^{1\times N}$ *and* $C_{t+1,T}^{(\pi)} = [\mathbf{c}_{ij}^T]_{i=1:N,\ j=1:M}$ *minimizes the function in Equation 12. The minimum value is equal to*

$$(\mathbf{1}^T + \eta^T \Gamma_{m(t)})\mathbf{x}(t) + (\beta^T(t) - \eta^T)\mathbf{u}^*(t) \quad 14.$$

Proof. First, use Equation 9 and Proposition 1 to obtain $\mathbf{q}(t+1)$. Then, use Proposition 4 to simplify the expected state norm from time $t+1$ to T in Equation 12. This reduces the problem to a variable-separable linear program. □

The optimal node control policy is a threshold rule that identifies nodes at which intervention is justified, and uses the largest control action subject to the limit of U_{max}, and such that $\mathbf{x}(t+1) \ge \mathbf{0}$. Note that $\mathbf{u}^*(t) = \mathbf{0}$ if no action can decrease the objective function.

Proposition 5. *When the topology is forced to transition to mode* k *at time* $t+1$*, the value of the objective function (Equation 12) is*

$$\|\mathbf{x}(t)\|_1 + \|C_{t+1,T}\, \mathbf{q}^{(k)}(t+1)\|_1 + \alpha_{m(t),k}(t) \quad 15.$$

where $\mathbf{q}_i^{(k)}(t+1) = \begin{cases} \Gamma_{m(t)}\mathbf{x}(t), & \text{if } i = k \\ \mathbf{0}, & \text{otherwise} \end{cases}$ *for* $i \in \{1, \ldots, M\}$

Proof. Separate the terms in Equation 12, and simplify using Propositions 1 and 2. □

Theorem 3 (Topology control)**.** *The optimal topology control action chooses the mode that yields the lowest objective function value (Equation 12) from among the following options:*

(i) No topology control, with objective function value given by Proposition 4.
(ii) Topology transition to any mode $k \in \{1, \ldots, M\}$*, with objective function value given by Proposition 5.*

Proof. The cases (i) and (ii) decompose the domain of the optimization. Hence, the global minimum is the best solution among these locally optimal solutions. □

The optimal topology control action is the result of evaluating $(M+1)$ sub-problems, and then choosing the best action (Theorem 3). The enumeration of the $(M+1)$ cases is tractable in practice since the matrices $C_{t+1,T}$ can be computed offline.

Proposition 6. *The optimal node control action* $\mathbf{u}^*(t)$ *conditioned on a forced topology transition to mode* k *at* $t+1$ *(i.e.,* $m(t+1) = k$*) is obtained by setting* $\eta^T = \sum_i \mathbf{c}_{ik}^T$ *in Theorem 2.*

Proof. Use Equation 9 and Proposition 1 to obtain an appropriate $\mathbf{q}(t+1)$. The rest of the proof follows that for Theorem 2. □

Theorem 4 (Integrated control)**.** *The optimal integrated (i.e., node and topology) control action at time* t *is the one that results in the lowest objective function value from among the following options:*

Control strategy	Delay cost	Number of control actions		
		Node	Topology	Both
Node controller	0.79	2.02	-	-
Topology controller	0.85	-	3.22	-
Integrated controller	0.70	1.91	2.26	0.07

Table 2 Controller performance averaged over $\alpha \in [0, 0.25]$ and $\beta \in [0, 200]$.

(i) No forced topology transition, and optimal node control as in Theorem 2.

(ii) Forced topology transition to some mode $k \in \{1, \ldots, M\}$, and a conditionally optimal node control action, as in Proposition 6.

Proof. Similar to the proof for Theorem 3. □

The integrated controller prescribes a potential topology transition and a control $\mathbf{u}^*(t)$ on the node state as described in Theorem 4. It is important to note that the integrated controller controller does not simply choose between the node and topology controllers, but rather chooses the optimal solution (i.e., the minimizer of Equation 12) over the space of all possible combinations of node and topology control actions.

5.2. Evaluation of Controller Performance

We aim to minimize airport delays over the course of a 24-hour operational day (4 am EST to 3 am EST). The initial conditions for the PMJLS model, $\mathbf{x}(0)$ and $m(0)$, are drawn from historical data (2011-12). We study the performance of the controllers in simulation, for varying values of node and topology control action penalties. In our numerical experiments, $\mathbf{x}(0)$ is normalized, $U_{max} = 100$, and the control penalties are parametrized as $\beta(t) = \beta\mathbf{1}$ and $\alpha_{ij}(t) = \frac{\alpha(25-t)}{\pi_{ij}(t)}$, where α and β are non-negative scalars. The node control penalty, β, is the unit cost for reducing the delay at an airport. The topology transition penalty is proportional to the time remaining during that day, since actions are more challenging to implement early in the day, and affect a higher volume of downstream traffic. Furthermore, the topology control costs are assumed to be inversely proportional to the transition probabilities, to reflect the property that rare transitions are more difficult to induce than the ones that occur naturally with a high probability.

For a set of parameters (α, β) and a control policy, we measure the effectiveness of a strategy in terms of its delay cost:

$$\text{Delay cost } (\alpha, \beta, \text{control policy}) = \frac{\sum_{\tau=1}^{24} \mathbb{E}\left[\|\mathbf{x}(\tau)\|_1 + \text{control penalty}(\tau)\right]}{\sum_{\tau=1}^{24} \mathbb{E}\left[\|\mathbf{x}(\tau)\|_1 \mid \text{no control action}\right]} \qquad 16.$$

The delay cost (or simply the cost) should be less than or equal to 1 for any reasonable controller; lower the delay cost, the better the controller performance. Finally, we note that the evolution of $\mathbf{x}(t)$ is stochastic and thus the expectation in Equation 16 is estimated empirically using $1,000$ simulations for each value of α and β.

5.3. Comparison of Node, Topology and Integrated Control

From Figure 10 (left and center) we observe that smaller values of α and β result in lower delay costs for the topology and node controller, respectively. This finding is consistent with our expectation that the effectiveness of a controller would decrease if the penalty for

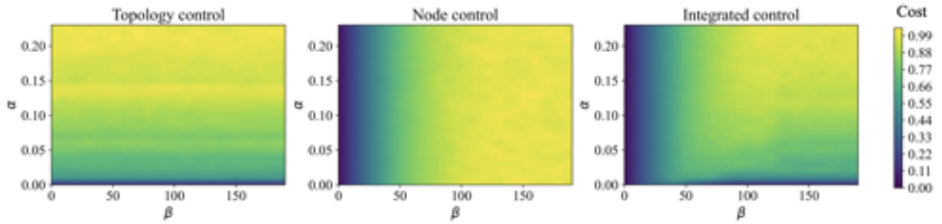


Figure 10

Delay cost contours for varying α and β, for (left) topology control, (center) node control, and (right) integrated control. The yellow regions denote the highest delay costs. Smaller penalties (values of α and β) result in lower costs, and that integrated control obtains lower costs over a larger range of penalties.

an intervention is high. The integrated controller jointly optimizes over the entire space of node and topology actions to obtain greater benefits over a wider range of α and β values (Figure 10, right).

Table 2 quantifies the average performance of the controllers over the entire parameter space. The integrated controller has the lowest average cost, and hence the best overall performance. Two major factors contribute to the lower costs: Firstly, the integrated controller can switch between node-only and topology-only control. This observation is qualitatively confirmed from Figure 10, where we see that the integrated controller captures the low cost regions of both the node and topology controllers. Secondly, the integrated controller can uniquely combine node and topology control actions to obtain lower costs than either of them can. For instance, $\alpha = 0.13$ and $\beta = 120$ results in node, topology, and integrated control costs of 0.98, 0.99, and 0.94, respectively. An analysis of the number of control actions (Table 2) yields similar insights: Although the integrated controller mostly chooses either a node-only or topology-only action, it also chooses to do both of them on occasion, in order to obtain lower costs.

Remark 1. *Node control can achieve significant delay cost reductions, even with a small number of control actions.*

Small penalties for control actions (i.e., small values of α and β) result in lower costs (Figure 10). However, lower delay costs are not necessarily associated with more control interventions, as shown in Figure 11. Here, we see that smaller control penalties can be associated with either a small or a large number of control actions, depending on the nature of the actions. In particular, the number of node control actions performed for small values of β is just 1, as the first action $\mathbf{u}(1)$ can be used to reduce the delays very effectively.

Remark 2. *The node, topology and integrated control policies are optimal one-shot controllers.*

These three controllers prescribe the optimal action to be taken at time t assuming that no other action will be taken at any future time, i.e., the objective function in Equation 12 optimizes for the best one-shot control action that could be performed at time t.

Remark 3. *The node, topology, and integrated controllers minimize the cost-to-go (Equation 12), and not the delay cost (Equation 16).*

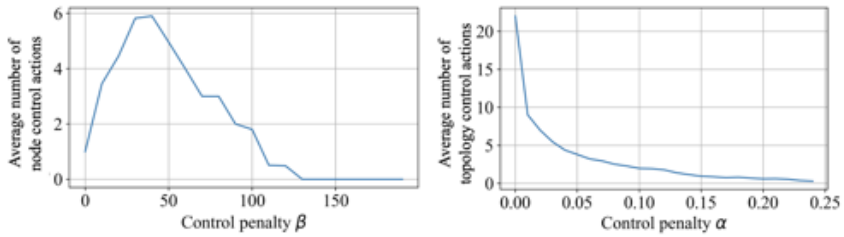


Figure 11

Number of control actions for (left) optimal node control, and (right) optimal topology control, for varying values of control action penalty, in our air traffic delay network example.

It is therefore possible that other controllers with potentially lower delay costs may exist. Furthermore, the delay cost of an integrated controller may not always be lower than that of a topology or node controller, even though the integrated controller is the optimal one-shot controller that minimizes the cost-to-go. For instance, when $\alpha = 0$ and $\beta = 20$, the node, topology and integrated control costs are 0.35, 0.11, and 0.33 respectively. In this case, the integrated controller, which is determined using one-shot optimization, is unable to incorporate the fact that mode transitions are free at all future times, and any actions on nodes, even though attractive currently, might increase the delay cost in the long run.

Remark 4. *Integrated controllers are a good choice for practical implementation.*

Real-world implementation occur in settings with uncertain dynamics, temporal and spatial constraints on the control actions, and with a rolling time horizon. One-shot controllers that minimize the cost-to-go are therefore a good choice in practice, since they are quite robust to model and control action uncertainty. Our numerical simulations suggest that the integrated controller not only performs the best on average over a wide range of parameters (Table 2), but is also never the worst-performing controller of the three alternatives.

5.4. Partial Controllability

The ability to control the continuous nodal state or topology transitions may be limited in practice. We consider two examples of partial controllability: The first one restricts the ability to reduce delays (i.e., node control) to only a subset of airports, for example, due to regulatory constraints. The second scenario affords limited temporal flexibility in implementing control actions, for example, due to coordination efforts between multiple ATC facilities. For each of these cases, we estimate the delay costs to evaluate the system-wide impact of these restrictions. These scenarios also help evaluate the resilience of system (by identifying critical airports, and control action timings), and help quantify the ability of the system to regulate its performance under degraded operational conditions.

Figure 12 plots the effect of limiting node action to only one airport. The top four airports in the US which have the largest individual ability to reduce total system delays (for $\alpha = 0.1$ and $\beta = 10$) are Atlanta (ATL), Chicago O'Hare (ORD), San Francisco (SFO), and Los Angeles (LAX). In other words, if we were forced to focus our delay reduction efforts at only one airport, choosing from within this set would be most effective.

The delay cost when restricting control actions to particular times of the day is shown

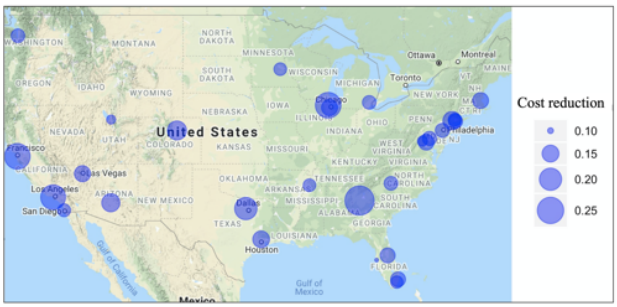


Figure 12

The size of the circle at an airport is proportional to the reduction in delay costs when the node control component of an integrated controller (with $\alpha = 0.1$ and $\beta = 10$) is limited to only that airport.

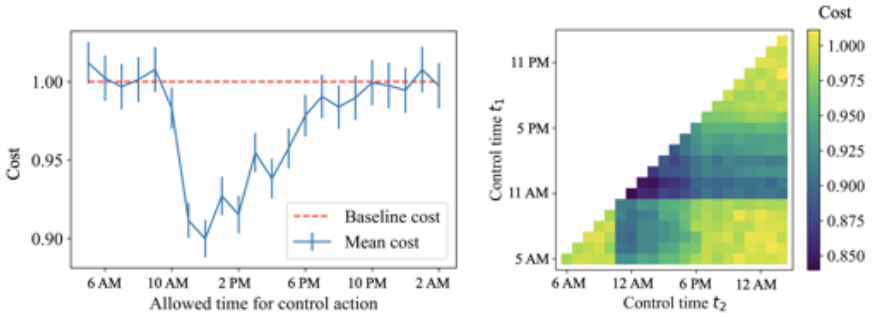


Figure 13

Delay costs (with the 99% confidence intervals) when integrated control actions are limited to (left) once in a day, at time t, and (right) twice in a day, at t_1 and t_2, where $t_1 < t_2$. Parameter values: $\alpha = 0.1$ and $\beta = 10$. All times are in EST.

in Figure 13. The ability to reduce delays varies significantly based on the chosen time(s). In particular, Figure 13 (left) shows that if we are allowed to perform an integrated control action only once during a day, doing it early in the morning or late at night is not effective. In the former case, delays have not significantly built up for the control to be effective (and the topology control cost is higher), whereas in the latter case, most of the day's traffic and delays have already been realized, and very little is left to be controlled. Figure 13 (left) suggests that around noon Eastern Time is the ideal time to implement the solution of an integrated controller, in order to decrease delays by almost 10%. This notion of restricting the timing of control actions can be extended to two time-periods, t_1 and t_2, as shown in Figure 13 (right): the results suggest that 11 am and 12 pm Eastern Time are the most effective times to apply control actions to reduce delay costs.

6. CONCLUSIONS

This paper illustrated the application of data-driven modeling, optimization, and control algorithms in ensuring the efficiency and robustness of a large-scale infrastructure. We showed that the dynamics of air traffic delays are well-represented by positive Markov Jump Linear System models, and that these models can then be used for the analysis and optimal control of the air transportation system. The research presented in this paper also reveals several promising directions for further investigation, including a comprehensive analysis of system resilience, the extension of these results to multilayer networks, the development of optimal recovery algorithms for networked systems, and the practical implementation of these approaches to enable safe and efficient air traffic operations.

SUMMARY POINTS

1. The analysis of airport delays using network models offers unique insights into the dynamics of delay propagation, thereby improving predictability and resilience to disruptions.
2. Markov Jump Linear Systems (MJLS) are a powerful abstraction to model the dynamics of networked systems with time-varying topologies.
3. Spreading processes on complex networks (e.g., delay propagation in the air traffic network) can be controlled by modifying the nodal state as well as the underlying network structure. This observation motivates the development of a novel class of MJLS controllers, and its application to air traffic delay networks.

FUTURE ISSUES

1. The problem of evaluating the robustness and resilience of systems with time-varying network topologies remains largely unexplored.
2. Multilayer networks promise to deliver more accurate and comprehensive representations of air transportation system dynamics (e.g., with separate and interacting network layers for delays, cancellations, traffic flows, and capacity); however, data-driven modeling, analysis and control of such systems remains an open challenge.
3. The true potential of the proposed techniques can only be achieved through real-world implementation, which requires bridging the gap between theory and practice in the control of complex, dynamic networks.

DISCLOSURE STATEMENT

The authors are not aware of any affiliations, memberships, funding, or financial holdings that might be perceived as affecting the objectivity of this review.

ACKNOWLEDGMENTS

We thank Richard Jordan for collaborating on the MJLS model development, and Zebulon Hanley processing the airport delay data. This work was supported in part by NSF (Award

No. 1739505) and NASA (Grant No. 80NSSC19K1607). Posted with permission from the *Annual Review of Control, Robotics, and Autonomous Systems*, Volume 4 © 2021 by Annual Reviews, `http://www.annualreviews.org/`.

LITERATURE CITED

1. International Air Transport Association (IATA). 2019. Economic Performance of the Airline Industry: 2019 End-Year Report
2. Ball M, Barnhart C, Dresner M, Hansen M, Neels K, et al. 2010. Total delay impact study
3. Joint Economic Committee, US Senate. 2008. Your Flight has Been Delayed Again: Flight Delays Cost Passengers, Airlines, and the US Economy Billions
4. Bratu S, Barnhart C. 2005. An Analysis of Passenger Delays Using Flight Operations and Passenger Booking Data. *Air Traffic Control Quarterly* 13:1–27
5. Amadeus, T2RL. 2016. Shaping the future ofAirline Disruption Management (IROPS)
6. Pejovic T, Noland RB, Williams V, Toumi R. 2009. A tentative analysis of the impacts of an airport closure. *Journal of Air Transport Management* 15:241–248
7. Madhani A, Mutzabaugh B, Spain W. 2014. FAA contractor charged with fire that halted flights
8. CNN. 2016. Southwest cancels 1,150 flights in 24 hours. money.cnn.com/2016/07/21/news/companies/southwest-airlines-flight-cancellations/index.html
9. CNN. 2016. Delta: 5-hour computer outage cost us $150 million. http://money.cnn.com/2016/09/07/technology/delta-computer-outage-cost/
10. Kafle N, Zou B. 2016. Modeling flight delay propagation: A new analytical-econometric approach. *Transportation Research Part B: Methodological* 93:520–542
11. Bureau of Transportation Statistics. 2020. Airline On-Time Statistics and Delay Causes
12. Eurocontrol Central Office of Delay Analysis (CODA). 2020. All-causes delay and cancellations to air transport in Europe: Annual report for 2019
13. Balakrishnan H. 2016. Control and optimization algorithms for air transportation systems. *Annual Reviews in Control* 41:39–46
14. Newman MEJ. 2004. Fast algorithm for detecting community structure in networks. *Physical Review E*
15. Salathe M, Kazandjieva M, Lee JW, Levis P, Feldman MW, Jones JH. 2010. A high resolution human contact network for infectious disease transmission. *Proceedings of the National Academy of Sciences* 107:22020–22025
16. Dietz K. 1967. Epidemics and rumours: A survey. *Journal of the Royal Statistical Society. Series A (General)* 130:505–528
17. Albert R, Albert I, Nakarado G. 2004. Structural vulnerability of the North American power grid. *Physical Review E* 69:025103
18. Crucitti P, Latora V, Marchiori M. 2004. A topological analysis of the Italian electric power grid. *Physica A: Statistical Mechanics and its Applications* 338:92–97
19. Crovella M, Krishnamurthy B. 2006. *Internet Measurement: Infrastructure, Traffic and Applications*
20. Kalapala V, Sanwalani V, Clauset A, Moore C. 2006. Scale invariance in road networks. *Physical Review E* 73:026130
21. von Ferber C, T. Holovatch YH, Palchykov V. 2009. Public transport networks: empirical analysis and modeling. *The European Physical Journal B* 68:261–275
22. Manitz J. 2014. Statistical Inference for Propagation Processes on Complex Networks. Ph.D. thesis, Georg-August-Universität Göttingen
23. Zanin M, Lillo F. 2013. Modelling the air transport with complex networks: A short review. *The European Physical Journal Special Topics* 215:5–21

24. Spiers G, Wei P, Sun D. 2012. *Algebraic Connectivity Maximization of the Air Transportation Network*. In *American Control Conference*
25. Wei P, Spiers G, Sun D. 2014. Algebraic connectivity maximization for air transportation networks. *IEEE Transactions on Intelligent Transportation Systems* :685–698
26. Lordan O, Sallan JM, Simo P, Gonzalez-Prieto D. 2014. Robustness of the air transport network. *Transportation Research Part E* :155–163
27. Nicolaides C, Cueto-Felgueroso L, Gonzalez MC, Juanes R. 2012. A Metric of Influential Spreading during Contagion Dynamics through the Air Transportation Network. *PLoS One* 7
28. Mucha PJ, Richardson T, Macon K, Porter MA, Onnela JP. 2010. Community Structure in Time-Dependent, Multiscale, and Multiplex Networks. *Science* 328:876–878
29. Balakrishnan H, Jung Y. 2007. *A framework for coordinated surface operations planning at Dallas-Fort Worth International Airport*. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, pp. 6553
30. Balakrishnan H, Chandran BG. 2010. Algorithms for scheduling runway operations under constrained position shifting. *Operations Research* 58:1650–1665
31. Jacquillat A, Odoni AR. 2018. A roadmap toward airport demand and capacity management. *Transportation Research Part A: Policy and Practice* 114:168–185
32. Bertsimas D, Patterson SS. 1998. The air traffic flow management problem with enroute capacities. *Operations Research* 46:406–422
33. Balakrishnan H, Chandran BG. 2017. *A Distributed Framework for Traffic Flow Management in the Presence of Unmanned Aircraft*. In *USA/Europe Air Traffic Management R&D Seminar*
34. Hu F, Yeung CH, Yang S, Wang W, Zeng A. 2016. Recovery of infrastructure networks after localised attacks. *Scientific reports* 6:24522
35. Cafieri S, Durand N. 2014. Aircraft deconfliction with speed regulation: new models from mixed-integer optimization. *Journal of Global Optimization* 58:613–629
36. Vossen T, Ball M. 2006. Optimization and mediated bartering models for ground delay programs. *Naval research logistics (NRL)* 53:75–90
37. Ball M, Donohue G, Hoffman K. 2006. Auctions for the safe, efficient, and equitable allocation of airspace system resources. *Combinatorial auctions* 1
38. Barnhart C, Belobaba P, Odoni AR. 2003. Applications of operations research in the air transport industry. *Transportation science* 37:368–391
39. Ball M, Barnhart C, Nemhauser G, Odoni A. 2007. Air transportation: Irregular operations and control. *Handbooks in operations research and management science* 14:1–67
40. Wang J, Mo H, Wang F, Jin F. 2011. Exploring the network structure and nodal centrality of china's air transport network: A complex network approach. *Journal of Transport Geography* 19:712–721
41. Neal Z. 2014. The devil is in the details: Differences in air traffic networks by scale, species, and season. *Social networks* 38:63–73
42. Boonekamp T, Burghouwt G. 2017. Measuring connectivity in the air freight industry. *Journal of Air Transport Management* 61:81–94
43. Guimera R, Mossa S, Turtschi A, Amaral LN. 2005. The worldwide air transportation network: Anomalous centrality, community structure, and cities' global roles. *Proceedings of the National Academy of Sciences* 102:7794–7799
44. Dunn S, Wilkinson SM. 2016. Increasing the resilience of air traffic networks using a network graph theory approach. *Transportation Research Part E: Logistics and Transportation Review* 90:39–50
45. Cardillo A, Zanin M, Gómez-Gardenes J, Romance M, del Amo AJG, Boccaletti S. 2013. Modeling the multi-layer nature of the european air transport network: Resilience and passengers re-scheduling under random failures. *The European Physical Journal Special Topics* 215:23–33

46. Janić M. 2015. Reprint of "modelling the resilience, friability and costs of an air transport network affected by a large-scale disruptive event". *Transportation Research Part A: Policy and Practice* 81:77–92
47. Boccaletti S, Latora V, Moreno Y, Chavez M, Hwang DU. 2006. Complex networks: Structure and dynamics. *Physics Reports* 424:175–308
48. Kleinberg J. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM* :604–632
49. Benzi M, Estrada E, Klymko C. 2013. Ranking hubs and authorities using matrix functions. *Linear Algebra and its Applications* :2447–2474
50. Deguchi T, Takahashi K, Takayasu H, Takayasu M. 2014. Hubs and Authorities in the World Trade Network Using a Weighted HITS Algorithm. *PLoS One* 9
51. Gopalakrishnan K, Balakrishnan H, Jordan R. 2016. *Clusters and communities in air traffic delay networks*. In *American Control Conference (ACC), 2016*, pp. 3782–3788. IEEE
52. Grabbe S, Sridhar B, Mukherjee A. 2014. *Clustering days with similar airport weather conditions*. In *14th AIAA Aviation Technology, Integration, and Operations Conference*, pp. 2712
53. Kuhn KD. 2016. A methodology for identifying similar days in air traffic flow management initiative planning. *Transportation Research Part C: Emerging Technologies* 69:1–15
54. Gorripaty S, Liu Y, Hansen M, Pozdnukhov A. 2017. Identifying similar days for air traffic management. *Journal of Air Transport Management* 65:144–155
55. Bloem M, Bambos N. 2015. Ground delay program analytics with behavioral cloning and inverse reinforcement learning. *Journal of Aerospace Information Systems*
56. Murça MCR, Hansman RJ. 2018. Identification, characterization, and prediction of traffic flow patterns in multi-airport systems. *IEEE Transactions on Intelligent Transportation Systems* 20:1683–1696
57. Rebollo JJ, Balakrishnan H. 2014. Characterization and prediction of air traffic delays. *Transportation Research Part C* :231–241
58. Schaeffer SE. 2007. Graph clustering. *Computer Science Review I* :27–64
59. Luo B, Wilson R, Hancock E. 2003. *Spectral clustering of graphs*. In *10th International Conference on Computer Analysis of Images and Patterns, CAIP*
60. Champin PA, Solnon C. 2003. *Measuring the similarity of labeled graphs*. In *Proceedings of the 5th international conference on Case-based reasoning (ICCBR'03): Research and Development*, pp. 80–95
61. Zager LA, Verghese GC. 2008. Graph similarity scoring and matching. *Applied Mathematics Letters* :86–94
62. Bonacich P. 2008. Factoring and weighting approaches to status scores and clique identification. *Journal of Mathematical Sociology* :113–120
63. Likas A, Vlassis N, Verbeek JJ. 2003. The global k-means clustering algorithm. *Pattern recognition* 36:451–461
64. Bonacich P. 1972. Factoring and weighting approaches to status scores and clique identification. *Journal of mathematical sociology* 2:113–120
65. Pyrgiotis N, Malone KM, Odoni A. 2013. Modelling delay propagation within an airport network. *Transportation Research Part C: Emerging Technologies* 27:60–75
66. Fleurquin P, Ramasco J, Eguiluz V. 2013. Systemic delay propagation in the US airport network. *Scientific Reports* :1159
67. Janić M. 2005. Modeling the large scale disruptions of an airline network. *Journal of transportation engineering* 131:249–260
68. Jacquillat A, Odoni AR. 2017. A roadmap toward airport demand and capacity management. *Transportation Research Part A: Policy and Practice*
69. Kicinger R, Chen JT, Steiner M, Pinto J. 2016. Airport capacity prediction with explicit consideration of weather forecast uncertainty. *Journal of Air Transportation* 24:18–28
70. Ramanujam V, Balakrishnan H. 2009. *Estimation of arrival-departure capacity tradeoffs in*

multi-airport systems. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pp. 2534–2540. IEEE

71. Choi S, Kim YJ, Briceno S, Mavris D. 2016. *Prediction of weather-induced airline delays based on machine learning algorithms*. In *Digital Avionics Systems Conference (DASC), 2016 IEEE/AIAA 35th*, pp. 1–6. IEEE
72. Xu N, Laskey KB, Donohue G, Chen CH. 2005. *Estimation of Delay Propagation in the National Aviation System Using Bayesian Networks*. In *6th USA/Europe Air Traffic Management Research and Development Seminar*
73. Kim YJ, Choi S, Briceno S, Mavris D. 2016. *A deep learning approach to flight delay prediction*. In *Digital Avionics Systems Conference (DASC), 2016 IEEE/AIAA 35th*, pp. 1–6. IEEE
74. Acemoglu D, Carvalho VM, Ozdaglar A, Tahbaz-Salehi A. 2012. The network origins of aggregate fluctuations. *Econometrica* 80:1977–2016
75. Herman I, Martinec D, Hurak Z, Sebek M. 2015. Nonzero Bound on Fiedler Eigenvalue Causes Exponential Growth of H-Infinity Norm of Vehicular Platoon. *IEEE Transactions on Automatic Control* 60:2248–2253
76. Sarkar T, Roozbehani M, Dahleh MA. 2019. Asymptotic network robustness. *IEEE Transactions on Control of Network Systems* 6:812–821
77. Ogura M, Preciado VM. 2015. *Disease Spread over Randomly Switched Large-Scale Networks*. In *Proceedings of the American Control Conference*
78. Gopalakrishnan K, Balakrishnan H, Jordan R. 2016. *Stability of networked systems with switching topologies*. In *Decision and Control (CDC), 2016 IEEE 55th Conference on*, pp. 2601–2608. IEEE
79. Liberzon D. 2012. *Switching in systems and control*. Springer Science & Business Media
80. Vargas AN, Costa EF, Val JB. 2013. On the control of markov jump linear systems with no mode observation: application to a dc motor device. *International Journal of Robust and Nonlinear Control* 23:1136–1150
81. Nowzari C, Preciado VM, Pappas GJ. 2016. Analysis and control of epidemics: A survey of spreading processes on complex networks. *IEEE Control Systems* 36:26–46
82. Xiao L, Hassibi A, How JP. 2000. *Control with random communication delays via a discrete-time jump system approach*. In *American Control Conference, 2000. Proceedings of the 2000*, vol. 3, pp. 2199–2204. IEEE
83. Zhang Y, Tian YP. 2009. Consentability and protocol design of multi-agent systems with stochastic switching topology. *Automatica* 45:1195–1201
84. Costa O, Fragoso M. 1993. Stability Results for Discrete-Time Linear Systems with Markovian Jumping Parameters. *Journal of Mathematical Analysis and Applications* 179:154–178
85. Costa O, Fragoso M, Marques R. 2005. *Discrete-Time Markov Jump Linear Systems*. USA: Springer-Verlag, 1st ed.
86. Gopalakrishnan K, Balakrishnan H. 2017. *A Comparative Analysis of Models for Predicting Delays in Air Traffic Networks*. In *USA/Europe Air Traffic Management Seminar*
87. Du Z, Ozay N, Balzano L. 2019. *Mode clustering for markov jump systems*. In *2019 IEEE 8th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pp. 126–130. IEEE
88. Liberzon D, Morse AS. 1999. Basic problems in stability and design of switched systems. *IEEE control systems magazine* 19:59–70
89. Lin H, Antsaklis PJ. 2009. Stability and stabilizability of switched linear systems: a survey of recent results. *IEEE Transactions on Automatic control* 54:308–322
90. Feng X, Loparo KA, Ji Y, Chizeck HJ. 1992. Stochastic stability properties of jump linear systems. *IEEE transactions on Automatic Control* 37:38–53
91. Costa OLV, Fragoso MD, Marques RP. 2006. *Discrete-time Markov jump linear systems*. Springer Science & Business Media

92. Boukas EK. 2007. *Stochastic switching systems: analysis and design*. Springer Science & Business Media
93. Bolzern P, Colaneri P, Nicolao GD. 2014. Stochastic stability of positive markov jump linear systems. *Automatica* 50:1181–1187
94. Guo Y. 2016. Stabilization of positive markov jump systems. *Journal of the Franklin Institute* 353:3428–3440
95. Kozin F. 1969. A survey of stability of stochastic systems. *Automatica* 5:95–112
96. Bolzern P, Colaneri P, De Nicolao G. 2006. On almost sure stability of continuous-time markov jump linear systems. *Automatica* 42:983–988
97. Cavalcanti J, Balakrishnan H. 2020. Sign-stability of positive markov jump linear systems. *Automatica* 111:108638
98. Cavalcanti J, Balakrishnan H. 2017. *Finite-time behavior of switching networks*. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 6505–6512. IEEE
99. Vugrin ED, Turnquist MA, Brown NJ. 2014. Optimal recovery sequencing for enhanced resilience and service restoration in transportation networks. *International Journal of Critical Infrastructures* 10:218–246
100. Chen L, Miller-Hooks E. 2012. Resilience: an indicator of recovery capability in intermodal freight transport. *Transportation Science* 46:109–123
101. Ho C, Kochenderfer MJ, Mehta V, Caceres RS. 2015. *Control of Epidemics on Graphs*. In *Proceedings of the Conference on Decision and Control*
102. Zhang X, Miller-Hooks E, Denny K. 2015. Assessing the role of network topology in transportation network resilience. *Journal of Transport Geography* 46:35–45
103. Nguyen TD, Cai X, Ouyang Y, Housh M. 2016. Modelling infrastructure interdependencies, resiliency and sustainability. *International Journal of Critical Infrastructures* 12:4–36
104. Ganin AA, Massaro E, Gutfraind A, Steen N, Keisler JM, et al. 2016. Operational resilience: concepts, design and analysis. *Scientific reports* 6
105. Watkins NJ, Nowzari C, Pappas GJ. 2017. Inference, prediction, and control of networked epidemics. *arXiv preprint arXiv:1703.07409*
106. Matei I, Martins NC, Baras JS. 2008. Optimal linear quadratic regulator for markovian jump linear systems, in the presence of one time-step delayed mode observations. *IFAC Proceedings Volumes* 41:8056–8061
107. Borrelli F, Baotić M, Bemporad A, Morari M. 2005. Dynamic programming for constrained optimal control of discrete-time linear hybrid systems. *Automatica* 41:1709–1721
108. Costa OLV, Assumpção Filho E, Boukas EK, Marques R. 1999. Constrained quadratic state feedback control of discrete-time markovian jump linear systems. *Automatica* 35:617–626
109. Blackmore L, Bektassov A, Ono M, Williams BC. 2007. *Robust, optimal predictive control of jump markov linear systems using particles*. In *International Workshop on Hybrid Systems: Computation and Control*, pp. 104–117. Springer
110. Park BG, Lee JW, Kwon WH. 1999. Robust one-step receding horizon control for constrained systems. *International Journal of Robust and Nonlinear Control* 9:381–395
111. Park BG, Kwon WH. 2002. Robust one-step receding horizon control of discrete-time markovian jump uncertain systems. *Automatica* 38:1229–1235
112. Yewen W, Zhu J, Xie W. 2015. *Optimal control strategy for discrete-time MJLS with controllable Markov chain and Gaussian white noise*. In *Control Conference (CCC), 2015 34th Chinese*, pp. 2188–2193. IEEE