Robust Control of Arrivals into a Queuing Network

Sandeep Badrinath, *Student member, IEEE,* Hamsa Balakrishnan, *Member, IEEE*

***Abstract*—Queuing networks have been widely-used to model congestion in transportation systems. Due to their interconnected nature, delays in a queuing network can propagate as customers traverse through the network; similarly, downstream resources can be underutilized due to poor control policies. This paper considers the regulation of arrivals into a queuing network in order to maintain a desired level of occupancy (queue length) in the system. The dynamics of the queuing network is represented by a fluid-flow model, which is then used to develop a robust controller for tracking the desired queue length. The controller is based on a sliding mode control approach, with predictor-based feedback to account for propagation delays. For a single queue, we determine sufficient conditions for tracking the queue length, and bounds on the tracking error. We also present an analysis of the tracking performance for queues in tandem.**

We demonstrate our approach for the example of airport surface congestion control. The proposed robust control framework is based on a queuing network model of the airport, and is used to tactically manage aircraft departures in order to reduce congestion on the airport tarmac.

***Index Terms*—robust control, queuing networks, time-delay systems, airport surface congestion**

I. Introduction

Queuing networks have been used to model congestion in a wide range of infrastructures, including communication systems, industrial supply chains, and transportation systems [1]–[4]. A queuing network is a collection of interconnected *servers* that represent the system's capacitated resources, and *customers* who wish to utilize these resources. For example, in an urban traffic network, road intersections can be viewed as the servers, and vehicles as the customers. The demand for a resource can be close to – or even exceed – its capacity, leading to congestion and the formation of large queues; this impact could cascade further into other resources. Congestion results in higher operating costs and increased wait times. A key challenge in queuing systems is the development of control strategies that can reduce congestion, while still satisfying operational constraints. The control inputs can vary depending on the specific application: examples include the rate at which customers are sent into the system, or the capacity of servers.

A. Control of queuing networks

A variety of control frameworks, based on models of varying complexity, have been proposed to reduce congestion in queuing networks. A significant amount of early research focused on the optimal control of arrival and service rates using Markov Decision Processes (MDPs) [5], [6]. However, the Markov chain representation for queuing dynamics that was used in this MDP framework relied on very restrictive assumptions (such as Poisson arrivals, exponential service time distributions, stationarity, and no propagation delays), which may not be valid in practice. In order to better understand the problem of congestion control in queuing networks, we describe the two main contexts in which it has typically been studied: Internet congestion control and urban traffic networks.

1) Internet congestion control: With the growth of the Internet, there was much interest in the analysis of congestion control protocols for communication networks [7]. The objective of these protocols is to maintain a desired quality of service, as measured in terms of delay, throughput, packet loss, or jitter. The primary congestion control mechanism in these networks was the regulation of the sending rate at the source (e.g., the transmission control protocol), and the queue length at the routers by dropping packets (e.g., active queue management techniques) [8], [9]. Control-theoretic techniques have been used to analyze the stability of such protocols and tune parameters [10]. Internet congestion control also presents some domain-specific challenges; for example, feedback to a source can only be based on delayed packet-loss information rather than actual queue length information.

Control-theory has been used to a limited extent in the design of congestion control protocols. Prior work has considered fluid-flow models, typically non-linear delay differential equations with time-delays to account for the round-trip travel time from the source to the receiver [4]. The resulting models have allowed researchers to use standard control techniques such as PID, $\mathcal{H}_\infty$, and variable structure controllers [11], [12]. To apply these standard techniques, the models were often linearized around an equilibrium point, and in a few cases, even the time-delays were ignored [10], [11], [13]. Most models assumed that the queue length of the bottleneck server is always positive, and that the queue lengths elsewhere are zero. Consequently, the resulting approaches tend to perform poorly in practice due to the lack of robustness to time-varying traffic, delays, and capacity [12].

2) Urban traffic networks: Traffic congestion on urban roads is often represented as a queuing network [14]. The problem of congestion control through regulating traffic signals corresponds to controlling the service rates for each of the flows in the queuing network. Timed traffic lights have been shown to be inefficient under time-varying arrival rates. Furthermore, the control policy needs to account for the downstream impacts of throttling upstream flows. To overcome these challenges, adaptive traffic signaling approaches, such as optimization-based techniques, have been proposed to reduce congestion [15]–[17]. However, online optimization

S. Badrinath is a PhD candidate in the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA-02139, USA. e-mail:sandeepb@mit.edu

H. Balakrishnan is a Professor in the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA-02139, USA. e-mail:hamsa@mit.edu

This work was supported in part by NSF under grant 1739505 and NASA under grant 80NSSC19K1607.

techniques for large queuing networks are computationally intensive, while decentralized approaches and heuristics are often sub-optimal.

B. Airport surface congestion

Despite the increase in air traffic demand in recent years, there have not been significant increases in infrastructure and capacity at major airports. The imbalance between airport capacity and demand, particularly during periods of peak traffic, has led to congestion and delays. Consequently, there is a significant interest among airlines and airport operators alike in developing operational strategies to reduce congestion.

Departure metering has been widely-recognized as a promising operational approach to mitigate airport surface congestion [18]–[21]. In the absence of departure metering, pilots push back from the gate and start taxiing whenever ready, resulting in long queues during periods of high departure demand. A departure metering procedure tactically holds aircraft at their gates during periods of congestion, and releases them in an appropriate manner such the surface experiences smaller queues, while still maintaining runway throughput. The result is a reduction in the taxi-out time of flights (travel time from the departure gate to take-off) and fuel savings.

The algorithms that determine the hold-decisions for departure metering most often rely on a model of the aircraft movements on the airport surface. Queuing networks have been shown to be effective in modeling airport surface operations [3], [22]–[24]. They have enabled delay prediction on the airport surface, as well as for the entire network of airports [3], [22]. Queue-based models have been used to develop departure metering algorithms using dynamic programming techniques or various heuristics [18], [25]. The resulting control strategies have been shown to perform well in simulations as well as in field demonstrations [18]. However, most prior work assumed congestion occured only at the runway, and modeled the departure process as a single queue. This assumption is not necessarily valid at major airports with multiple congested locations. We have previously developed and validated models of traffic movement at such airports using a larger queuing network [23]. Controlling such a queuing network is challenging because the queues are inherently time-varying (due to demand and capacity fluctuations), and due to the propagation of congestion effects between the nodes of the network, and the time-delays involved. Furthermore, the problem is not amenable to traditional probabilistic models, particularly for large dynamic queuing networks with time-delays.

C. Contributions of this paper

We consider the problem of controlling the arrivals into a queuing network in order to maintain the lengths of queues in the network at desired levels. In the context of departure metering, this corresponds to determining the release times of aircraft from their gates in order to maintain a limited queue at the runway, resulting in reduced taxi-out delays, without under-utilizing the runway. More generally, we propose a feedback controller to track desired queue lengths by controlling the sending rate at the source. The proposed approach accommodates queuing networks with propagation time-delays between nodes, and arbitrary service time distributions.

We use a fluid-flow model to represent the evolution of queues in the queuing network [26], [27]. Although deterministic, the model captures the dynamic behaviour of non-stationary queues, and allows us to develop robust control strategies that account for uncertainties. Prior work on controlling queues using fluid models with techniques from nonlinear control theory did not account for time-delays, and only considered single servers [28], [29]. Time-delays can be destabilizing; traditional techniques for analyzing the stability of time-delay systems (e.g., Lyapunov-Krasovskii or Lyapunov-Razumikhin methods) are challenging and often result in very conservative results for nonlinear systems [30], [31].

In this paper, queue length tracking is achieved through a sliding mode control approach, with predictor-based feedback to compensate for the time-delays. Using Lyapunov analysis, we determine sufficient conditions for tracking the queue length, and bounds on the tracking error, for the case of a single queue. We also illustrate the performance of the controller for tandem queues using simulations. We build a queuing network model of the movement of departure traffic at Charlotte Douglas International Airport (CLT), one of the busiest airports in the world. Using this model, we develop a robust controller to determine the modified pushback times of departing flights to reduce congestion on the airport surface. Simulations indicate that our robust control methodology performs better than a heuristic that is currently being used in field trials at CLT [32].

II. Prior work: Queuing network model [23]

The exact analysis of non-stationary queuing networks is analytically challenging, causing researchers to resort to numerical simulations or approximations. In this paper, we use a fluid-flow model based on point-wise stationary approximation to represent the queuing process [23], [26], [33]. The model is a continuum approximation to the discrete queuing problem, derived by combining results from steady-state queuing theory with the flow conservation principle. The results in this paper leverage a previously-developed queuing network model [23], which we summarize for completeness. We first present the model for a single queue, and then extend it to the case of a queuing network.

A. Single queue

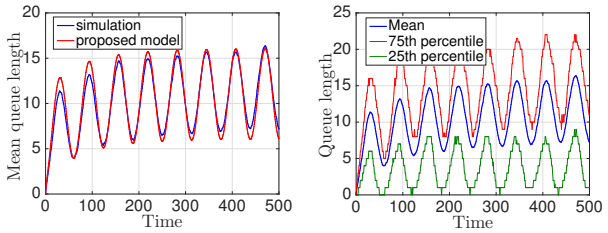
Consider a single queue with a server that has a stochastic service time distribution. We assume that the queue has infinite buffer capacity (no blocking), a common assumption in queuing network analysis. Let $x(t)$ represent the average number of customers in the queue at time t. Let $\lambda(t)$ and $\mu(t)$ denote the average arrival rate and service rate, respectively. Note that these correspond to the ensemble averages. Assuming that the arrival rates into the queue are Poisson, the evolution of $x(t)$ can be approximated by

$$\dot{x}(t) = -\mu(t)\frac{C(t)x(t)}{1 + C(t)x(t)} + \lambda(t). \tag{1}$$

Here, $Cx/(1+Cx)$ represents an approximation for the utilization factor of the server, whose value is zero for $x=0$ and one as $x \to \infty$. The parameter C depends on the coefficient of variation of the service time distribution, and is given by

$$C = \underset{C'}{\operatorname{argmin}} \int_0^{x_m} \left(\frac{y+1-\sqrt{y^2+2{C_v}^2+1}}{1-{C_v}^2} - \frac{C'y}{C'y+1} \right)^2 dy.$$

Here, x_m is the expected maximum queue length, and C_v is the coefficient of variation of the service time distribution of the server. Fig. 1(a) shows the mean queue length obtained from the analytical queue model for a $M_t/M/1$ queue, and a stochastic simulation of the queuing process. The stochastic queuing simulation involves sampling from the arrival and service time distributions, and the ensemble average queue length is computed using 3,000 independent samples. We see that the results from the analytical queue model closely match the simulation results. The model is seen to perform similarly well for other general service time distributions.



(a) Comparison between simulation and proposed model.

(b) Simulation results.

Fig. 1. Mean queue length for $M_t/M/1$ queue ($\lambda/\mu = 0.9 + 0.45\sin(0.1t)$; $\mu = 2$).

It is worth noting that the model governs the evolution of the ensemble mean queue length and not the actual queue length. Since the arrival times and service times are stochastic in nature, the actual queue length is a random variable. Probabilistic queuing models, such as a Markov chain representation of the queuing process, provide the queue length probabilities at any time using the Chapman-Kolmogrov equation. However, such models are often complex and restrictive, and are difficult to scale to controlling queue lengths in large queuing networks. Fig. 1(b) shows the ensemble mean queue length for a $M_t/M/1$ system obtained from the simulations, along with the 75th percentile and 25th percentile of the queue length at any time instant. The figure shows that the actual queue length at any time can be considered to be a small deviation from the mean queue length. Consequently, we can use a model of the mean queue length to control the actual queue length, by considering uncertainties in the model prediction. Our proposed model for the mean queue length, Eq. (1), allows us develop robust controllers to account for model uncertainties.

B. Queuing networks

The single queue model can be extended to multiple queues using the flow conservation principle: the output of one queue becomes the input to another queue, if they are connected. Let R be the routing matrix, with elements r_{ij} representing the fraction of customers joining queue j after being served by server i. Let λ_i be the exogenous input into queue i with mean service rate μ_i. The dynamics of the mean queue length for queue i, denoted x_i, is given by:

$$\dot{x}_i = -\mu_i(t)\frac{C_i(t)x_i}{1+C_i(t)x_i} + \lambda_i(t) + \sum_j \mu_j(t)\frac{C_j(t)x_j}{1+C_j(t)x_j} r_{ji}(t). \tag{2}$$

Time-delays due to propagation are a feature of many queuing networks. This propagation delay does not include the wait time in the queue. Let τ_{ij} be the propagation time (travel time) from server i to j. Then, the mean queue length is given by the following delay differential equation:

$$\begin{aligned} \dot{x}_i &= -\mu_i(t)\frac{C_i(t)x_i(t)}{1+C_i(t)x_i(t)} + \lambda_i(t) \\ &\quad + \sum_j \mu_j(t-\tau_{ji})\frac{C_j(t-\tau_{ji})x_j(t-\tau_{ji})}{1+C_j(t-\tau_{ji})x_j(t-\tau_{ji})} r_{ji}(t-\tau_{ji}). \end{aligned}$$

The model can also be extended to handle multiple classes of customers [33]. Let $i = 1, 2, .. \ l$ be the different class of customers in the system, with x_{Ti} being the number of customers of class i in the queue buffer and λ_{Ti} being the mean arrival rate of customers of class i. The evolution of total number of customers in the queue ($x_T = \sum_{i=1}^{l} x_{Ti}$) can be obtained based on Eq. (1) using:

$$\dot{x}_T = -\mu(t)\frac{C(t)x_T}{1+C(t)x_T} + \sum_{i=1}^{l} \lambda_{Ti}(t). \tag{3}$$

The effective mean service rate for each class in the queuing dynamics is assumed to be proportional to the fraction of customers of that particular class in the queue buffer, considering the same service time distribution for all customers. Then, the evolution of mean queue length of class i is given by:

$$\dot{x}_{Ti} = -\mu(t)\frac{x_{Ti}}{x_T}\frac{C(t)x_T}{1+C(t)x_T} + \lambda_{Ti}(t) \tag{4}$$

$$\dot{x}_{Ti} = -\mu(t)\frac{C(t)x_{Ti}}{1+C(t)x_T} + \lambda_{Ti}(t). \tag{5}$$

III. Tracking Controller for a Single Queue

In this section, we consider the case of a single queue served by a single server, and then present the analysis for more complex queuing networks in Sections IV and V. The objective is to maintain a desired queue length by controlling the release times into the queue. We first consider the case in which there is no travel time from the source (where the customers are released) to the point of entry into the queue. We will later consider the scenario with propagation delays.

A. Case without propagation delays

The fluid-flow model for a single queue served by a single server is given by the following equation:

$$\dot{x} = \bar{\alpha}(x,t) + u(t), \tag{6}$$

where $\bar{\alpha}(x,t) = -\mu(t)C(t)x/(C(t)x+1)$, μ is the mean service rate of the server, and the parameter C primarily depends on the coefficient of variation of the service time

distribution. We assume that the actual dynamics deviates from the model, but has a similar structure of the form:

$$\dot{x} = \alpha(x,t) + u(t), \tag{7}$$

where $\alpha(x,t)$ is an unknown function that is bounded as follows:

$$|\alpha(x,t) - \bar{\alpha}(x,t)| \leq F(x,t). \tag{8}$$

Motivated by the fact that the errors arise primarily due to uncertainties in the individual service times, we consider,

$$F(x,t) = a(C(t)x)/(C(t)x+1). \tag{9}$$

Here, a is a design parameter that needs to be chosen depending on the level of uncertainty. The objective of the control problem is to determine a sending rate ($u(t)$) in order to maintain the queue length at a desired value, $x_d(t) \geq 0$.

We first present a few standard definitions and theorems on stability properties that are used to develop the feedback controller [34], [35].

Definition 1 (Equilibrium point)**.** *The state x^* is said to be an equilibrium point of the system $\dot{x} = f(x,t)$ if: $x(t_0) = x^* \implies f(x^*,t) = 0$ for all $t \geq t_0$. Without loss of generality, one can always consider the origin to be the equilibrium point through a simple coordinate transformation.*

Definition 2 (Stability)**.** *The equilibrium point $x = 0$ is said to be stable at initial time t_0 if, for any $R > 0$, there exists $r(R,t_0) > 0$, such that if $\|x(t_0)\| < r$, then $\|x(t)\| < R$ for all $t \geq t_0$. Otherwise, the equilibrium point is said to be unstable. Additionally, the equilibrium point is said to be uniformly stable if the value of r in the preceding definition can be chosen independent of t_0.*

Definition 3 (Asymptotic stability)**.** *The equilibrium point $x = 0$ is asymptotically stable at initial time t_0, if it is stable and, in addition, there exists some $r(t_0) > 0$ such that if $\|x(t_0)\| < r$ then $\|x(t)\| \to 0$ as $t \to \infty$. Additionally, the system is globally uniformly asymptotically stable if these properties hold true for any choice of r.*

Definition 4 (Class K, KR functions)**.** *A function $\phi(x)$ belongs to class K if it is continuous, strictly increasing, and $\phi(0) = 0$. Additionally, $\phi(x)$ belongs to class KR if $\phi(x)$ belongs to class K and $\phi(x) \to \infty$ as $x \to \infty$.*

Definition 5 (Positive definite functions)**.** *A continuous function $V(x,t)$ is said to be a positive definite function if for some $\phi(.)$ of class KR, $V(0,t) = 0$ and $V(x,t) \geq \phi(|x|)$ for all $t \geq 0$. Additionally, $V(x,t)$ is called a negative definite function if $-V(x,t)$ is positive definite.*

Definition 6 (Decrescent functions)**.** *A continuous function $V(x,t)$ is said to be decrescent if there exists a function $\gamma(.)$ of class K, such that, $V(x,t) \leq \gamma(|x|)$ for all $t \geq 0$.*

Theorem III.1 (Lyapunov theorem for global asymptotic stability [35])**.** *Assume there exists a scalar function $V(x,t)$ with continuous partial derivatives such that: (a) $V(x,t)$ is positive definite and decrescent, and (b) $\dot{V}(x,t)$, which is given by $\left(\frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(x,t)\right)$, is negative definite. Then, the equilibrium point at the origin is globally uniformly asymptotically stable.*

Next, we develop a feedback controller for the sending rate such that the queue length asymptotically tracks a desired value. The control input must be non-negative since it represents the sending rate, thereby imposing an additional constraint on the model uncertainty ($F(x,t)$) to guarantee tracking. The following lemma presents the control law for tracking the queue length.

Lemma III.2. *Consider the fluid-flow model for a single queue served by a single server without any propagation delays from the source (represented by the dynamics in Eq. (7), with the bounded model uncertainties described by Eq. (8)). A control input of the form $u(t) = \max\Big(-\bar{\alpha}(x,t) + \dot{x}_d - k\,sgn(x(t) - x_d(t)), 0\Big)$, with $k = F(x,t) + \eta$ for some $\eta > 0$, guarantees asymptotic tracking of the desired queue length ($x_d(t)$) if the bounds on the model uncertainty satisfy $F(x,t) < -\big(\bar{\alpha}(x,t) - \dot{x}_d\big)$ when $x > x_d(t)$.*

Proof. A feedback law for the sending rate is determined using a sliding mode control approach [34]. A sliding variable (s) is defined in terms of the tracking error (e) as follows:

$$s = e(t) = x(t) - x_d(t) \implies \dot{s} = \alpha(x,t) + u(t) - \dot{x}_d(t).$$

Asymptotic tracking (i.e., $e(t) \to 0$ as $t \to \infty$) of the queue length is achieved by requiring the squared distance to the sliding surface ($s = 0$) decreases along all trajectories:

$$\frac{1}{2}\frac{ds^2}{dt} \leq -\eta|s|,\ \eta > 0 \implies s\dot{s} \leq -\eta|s| \tag{10}$$

The best approximation to the control input that would achieve $\dot{s} = 0$ is represented by $\hat{u}$, and is given by, $\hat{u} = -\bar{\alpha}(x,t) + \dot{x}_d$. Consider the control law of the form

$$u(t) = \hat{u} - k\,\mathrm{sgn}(s) = -\bar{\alpha}(x,t) + \dot{x}_d - k\,\mathrm{sgn}(s). \tag{11}$$

The discontinuity is added across the sliding surface to account for model uncertainties [34]. To determine the gain parameter, k, consider,

$$\begin{aligned} s\dot{s} &= s\big(\alpha(x,t) + u(t) - \dot{x}_d\big) && (12)\\ &= s\big(\alpha(x,t) - \bar{\alpha}(x,t) - k\ \mathrm{sgn}(s)\big) && (13)\\ &= s\big(\alpha(x,t) - \bar{\alpha}(x,t)\big) - k|s| && (14)\end{aligned}$$

If we chose $k \geq F(x,t) + \eta$, then the sliding condition in Eq. (10) is satisfied, and therefore the resulting control law in Eq. (11) is guaranteed to asymptotically track the desired queue length. However, the control input needs to be non-negative since it represents the sending rate from the source. Therefore, the feedback law is modified as:

$$u(t) = \max\big(\hat{u} - k\,\mathrm{sgn}(s), 0\big) \triangleq \big(\hat{u} - k\,\mathrm{sgn}(s)\big)^+. \tag{15}$$

Next, we show that the resulting closed loop dynamics asymptotically tracks the desired queue length even with the saturated control input under certain conditions.

We make use of the Lyapunov theorem (Theorem III.1) to show that the queue length tracks the desired value using the control input given in Eq. (15). Consider the Lyapunov function candidate, $V(s,t) = s^2$. Here, $V(s,t)$ is a positive

definite function and decrescent. The time derivative of V is given by

$$\dot{V} = s\dot{s} = s\big(\alpha(x,t) + \big(\hat{u} - k\ \text{sgn}(s)\big)^{+} - \dot{x}_d\big). \quad (16)$$

When the control input is not saturated ($\hat{u} - k\ \text{sgn}(s) \geq 0$), the sliding condition ensures that $\dot{V}$ is negative definite.

Next, we consider the case when the control input is saturated. Using Eq. (15), the condition for control input saturation is given by

$$\big(-\bar{\alpha}(x,t) + \dot{x}_d - (F(x,t) + \eta)\ \text{sgn}(s)\big) < 0. \quad (17)$$

The control input can saturate only when $s > 0$, if we choose $\eta > \max\Big((\bar{\alpha} - \dot{x}_d - F(x,t)), 0\Big)$. For the case of saturated control input,

$$\dot{V} \quad \leq \quad F(x,t)|s| + s\big(\bar{\alpha}(x,t) - \dot{x}_d\big). \quad (18)$$

From the above inequality, $\dot{V} \leq -\eta|s|$ if $F(x,t) = -\big(\bar{\alpha}(x,t) - \dot{x}_d\big) + \eta$ for any $\eta > 0$ and $s > 0$. Therefore, $\dot{V}$ is negative definite even with control input saturation if $F(x,t) < -\big(\bar{\alpha}(x,t) - \dot{x}_d\big)$. Hence, the equilibrium $s = 0$ (that corresponds to $x = x_d$), is asymptotically stable, guaranteeing perfect tracking.

□

Remark: The bounds on the uncertainty in the dynamics ($F(x,t)$) becomes more conservative if the desired queue length, $x_d(t)$, has larger fluctuations.

B. Case with propagation delays

Consider the case in which there is a travel time, τ, to move from the source to the server. Assume that the travel time is a known constant. The queue length is given by

$$\dot{x} = \bar{\alpha}(x,t) + u(t-\tau). \quad (19)$$

Applying the delay-free controller that we developed earlier on the time-delay system can lead to poor tracking. To illustrate this, let us consider a simple example where the desired queue length to be tracked is of the form $x_d = 8 + 2\sin(0.1t)$. The system response with the delay-free controller that we developed earlier is shown in Fig. 2. If there are no delays, the tracking is perfect with the robust controller as intended. However, the observed queue length deviates from the desired value with the introduction of time-delay, indicating the need to develop a controller that explicitly accounts for it.

1) Predictor-based feedback control: A predictor based feedback controller is used to compensate for the time-delays in the system [36]. The system is forward-complete, that is, the state trajectories are well-defined for all $t \geq 0$, for every initial condition and bounded input signal. Forward-completeness ensures that the state does not become unbounded before the control acts on the state due to delays. Let $P_t(t+\tau)$ be the predictor for the state at time $(t+\tau)$, computed at time t. The predictor, $P_t(\theta),\ \forall\theta \in (t, t+\tau]$ is obtained by integrating the model equations with input delays (Eq. (19)) forward in time with the initial condition, $P_t(t) = x(t)$, as:

$$P_t(t+\tau) = P_t(t) + \int_{t-\tau}^{t} \Big(\bar{\alpha}\big(P_t(\theta+\tau), \theta+\tau\big) + u(\theta)\Big) d\theta. \quad (20)$$

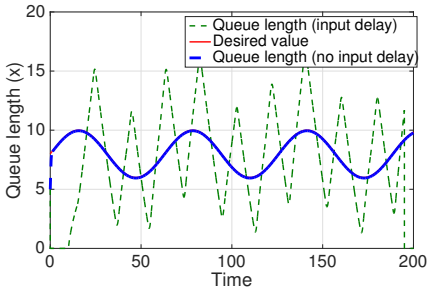


Fig. 2. Queue length obtained when the delay-free controller is applied on the time-delay system, for $x_d = 8 + 2\sin(0.1t))$, $\tau = 5,\ C = 1,\ \mu = 1$.

Using the feedback law for the delay-free system (Eq. (15)), the predictor-based feedback law for the delayed system is obtained by replacing the current state values with the corresponding predicted states, as:

$$u(t) = \big(-\bar{\alpha}(P_t(t+\tau), t+\tau) + \dot{x}_d(t+\tau) - k\ \text{sgn}(P_t(t+\tau) - x_d(t+\tau))\big)^{+} \quad (21)$$

Next, we determine the tracking guarantees with the above predictor-based feedback law. We determine bounds on the prediction errors, and use those bounds to obtain guarantees for tracking the queue length. The following lemma provides the bounds for the prediction errors.

Lemma III.3. *Consider the fluid-flow model for a single queue served by a single server with a known propagation delay (τ) to move from the source to the server (i.e., dynamics given by Eq.* (19)*, and bounded model uncertainties as in Eq.* (8)-(9)*). The error between the predicted queue length (given by Eq.* (20)*) and its actual value is bounded by*

$$|P_{t-\tau}(t) - x(t)| < a\tau. \quad (22)$$

Proof. For the ease of notation, we denote $\hat{P}(\theta) = P_{t-\tau}(\theta)$, the predictions for $x(\theta)$ computed at time $(t-\tau)$. Using Eq. (20), the dynamics of $\hat{P}(\theta)$ is as follows:

$$\dot{\hat{P}}(\theta) = \bar{\alpha}\big(\hat{P}(\theta), \theta\big) + u(\theta-\tau);\ \theta \in [t-\tau, t], \quad (23)$$

with the initial condition $\hat{P}(t-\tau) = x(t-\tau)$. Suppose that the actual queue dynamics evolves according to

$$\dot{x} = \alpha(x,t) + u(t-\tau). \quad (24)$$

Similar to the delay-free case, we assume that the function $\alpha(x,t)$ is unknown but bounded by $|\alpha(x,t) - \bar{\alpha}(x,t)| < F(x,t)$, and that the time delay (τ) is known. The error in predicting $x(t)$ at time $(t-\tau)$ is given by $\hat{e}(\theta) = e_{t-\tau}(\theta) = \hat{P}(\theta) - x(\theta)$. From Eqs. (23)-(24), the error dynamics is given by:

$$\dot{\hat{e}}(\theta) = \bar{\alpha}\big(\hat{P}(\theta), \theta\big) - \alpha(x(\theta), \theta);\ \ \theta \in [t-\tau, t], \quad (25)$$

with the initial condition $e(t-\tau) = 0$. Simplifying further, we obtain

$$\dot{\hat{e}}(\theta) = -\gamma(\theta)\hat{e}(\theta) + d(x(\theta)), \quad (26)$$

where $\gamma(\theta) = \mu(\theta)C(\theta)/\Big((C(\theta)\hat{P}(\theta)+1)(C(\theta)x(\theta)+1)\Big)$, and $d(x(\theta)) = \big(\bar{\alpha}(x(\theta),\theta) - \alpha(x(\theta),\theta)\big)$. The error dynamics is stable since $\gamma(\theta) > 0$, and the solution is given by

$$\begin{aligned}\hat{e}(t) &= \exp\Big(-\int_{t-\tau}^{t}\gamma(\theta)d\theta\Big)\hat{e}(t-\tau) \\ &\quad + \int_{t-\tau}^{t}\exp\Big(-\int_{r}^{t}\gamma(\theta)d\theta\Big)d(x(r))dr \leq a\tau,\end{aligned}$$

since $\exp\Big(-\int_r^t \gamma(\theta)d\theta\Big) \leq 1$ (since $\gamma(\theta) > 0$) and $d(x(t)) \leq F(x,t) < a$. Therefore, the error bound (D) for the prediction of $x(t)$ computed at time $t-\tau$ is given by

$$|\hat{P}(t) - x(t)| < D = a\tau. \tag{27}$$

□

The above result indicates that the prediction error is proportional to the time delay (τ) and magnitude of uncertainty in the dynamics (a). Next, we investigate the impact of the prediction error on tracking the queue length.

2) Tracking performance of the controller: Using the results from Lemma III.3, we can show that with predictor-based feedback control, the queue length always converges to a set around the desired value. To do so, we define the concept of ultimate boundedness [37].

Definition 7 (Ultimate boundedness for time-delay systems [38])**.** *The solutions $x_t(t_0,\phi)$ of $\dot{x} = f(t,x_t)$ are said to be uniformly ultimately bounded if there exists an $\eta > 0$ and a $\hat{t} = \hat{t}(\eta,\delta) > 0$ independent of t_0 such that $\|x_t(t_0,\phi)\| \leq \eta$ for all $t \geq t_0 + \hat{t}$, when $|\phi| < \delta$. Here, $x_t(t_0,\phi)$ refers to the solutions of $\dot{x} = f(t,x_t)$ with the initial condition ϕ at t_0.*

The following theorem presents the key result for tracking guarantees using predictor-based feedback control:

Theorem III.4. *Consider the fluid-flow model for a single queue served by a single server, with a known propagation delay (τ) to move from the source to the server (i.e., dynamics from Eq.* (19) *and bounded model uncertainties given by Eq.* (8)-(9)*). Then, the control input*

$$u(t) = \big(-\bar{\alpha}(P_t(t+\tau),t+\tau) + \dot{x}_d(t+\tau) - k\ sgn(P_t(t+\tau) - x_d(t+\tau))\big)^+ \tag{28}$$

with

$$P_t(t+\tau) = \int_{t-\tau}^{t}\Big(\bar{\alpha}\big(P_t(\theta+\tau),\theta+\tau\big) + u(\theta)\Big)d\theta,\ P_t(t) = x(t); \tag{29}$$

$$\text{and } k = F(x,t) + \eta + \mu C a\tau;\ \eta > \big(\bar{\alpha} - \dot{x}_d - (F+\mu C a\tau)\big)^+$$

guarantees that the queue length is uniformly ultimately bounded if $F(x,t) < (-\bar{\alpha}(x,t) + \dot{x}_d)$ when $x > x_d + a\tau$. Furthermore, the ultimate bounds are given by $\|x - x_d\| < a\tau\ \forall\ t \geq t_0 + T$, where $T = \frac{(x_0-x_d)^2 - a^2\tau^2}{ka\tau}$.

Proof. Consider the Lyapunov function candidate, $V(s,t) = s^2$, where $s = (x - x_d)$, as before. The time derivative of V is given by

$$\dot{V} = s\Big(\alpha(x,t) + \big(-\bar{\alpha}(\hat{P}(t),t) + \dot{x}_d - k\,\mathrm{sgn}(\hat{P}(t) - x_d(t))\big)^+ - \dot{x}_d\Big)$$

When $x > x_d + D$, we have $\hat{P}(t) - x_d > 0$ from Eq. (27) and $s > 0$. Then,

$$\begin{aligned}\dot{V} &= s\Big(\alpha(x,t) - \bar{\alpha}(\hat{P}(t),t) - k\Big) \\ &\leq sF(x,t) + s\Big(\bar{\alpha}(x,t) - \bar{\alpha}(\hat{P}(t),t)\Big) - sk.\end{aligned}$$

However, since $\big(\bar{\alpha}(x,t) - \bar{\alpha}(\hat{P}(t),t)\big) = \frac{-\mu C(x-P)}{(Cx+1)(CP+1)} \leq \mu CD$, we obtain

$$\dot{V} \leq sF(x,t) + s\mu CD - sk \tag{30}$$

For the case when $x < x_d - D$, we have $\hat{P}(t) - x_d < 0$ from Eq. (27) and $s < 0$. Then, similarly:

$$\dot{V} \leq |s|F(x,t) + |s|\mu CD + sk. \tag{31}$$

Eqs. (30)-(31) imply that $\dot{V} < 0$ for $|x - x_d| > D$ when the gain, k, is appropriately chosen. With the above results, we can show that queue length always converges to a set around the desired value with predictor-based feedback control, or equivalently, the trajectories of the closed-loop system are uniformly ultimately bounded.

If the gain k in Eqs. (30)-(31) is chosen such that $k = \mu CD + F(x,t) + \eta,\ \eta > 0$, then we obtain $\dot{V} \leq -\eta|s|$ for $\|s\| > D$. Therefore, for $\|s\| > D$, $\dot{V} \leq -\eta D$, which implies that the trajectory behaves as if the origin ($s = 0$) is asymptotically stable and satisfies an inequality of the form, $s^2 \leq s_0^2 - \eta D(t - t_0)$. The trajectories that start in or those that reach the set, $\{s : \|s\| \leq D\}$, will remain within that set since $\dot{V}$ is negative on the boundary of that set. Therefore, the system is uniformly ultimately bounded with an ultimate bound D, which implies $\|x - x_d\| < D\ \forall\ t \geq t_0 + T$, where $T = \frac{(x_0-x_d)^2 - D^2}{\eta D}$. Therefore, the queue length converges to a set around the desired queue length, $\{x : \|x - x_d\| < D = a\tau\}$, in finite time. This set depends on the time-delay and model uncertainty.

The above results are valid even when the controller saturates under certain conditions. From Eq. (21), the control input saturates if:

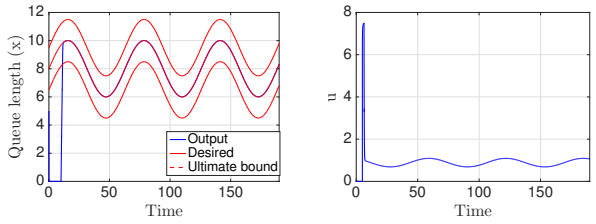
$$-\bar{\alpha}(\hat{P}(t),t) + \dot{x}_d - (F + \mu CD + \eta)\ \mathrm{sgn}(\hat{P}(t) - x_d(t)) < 0.$$

If $\eta > \max\Big(\bar{\alpha} - \dot{x}_d - (F + \mu CD), 0\Big)$, then the control input does not saturate when $x < x_d - D$ (from Eq. (27)). For the case when the controller saturates and $x > x_d + D$, we have:

$$\dot{V} \leq F(x,t)|s| + s\big(\bar{\alpha}(x,t) - \dot{x}_d\big).$$

From the above inequality, $\dot{V}$ is negative for $x > x_d + D$ if $F(x,t) \leq -\big(\bar{\alpha}(x) - \dot{x}_d\big)$. The condition on $F(x,t)$ is similar to what we had obtained earlier for the case without time-delays. Therefore, even with controller saturation, the above system is uniformly ultimately bounded with these additional conditions. □

The predictor in Eq. (29) is computed through numerical integration. To eliminate chattering (i.e., the control input switching at high frequencies at the sliding surface), the *sgn(.)* function in the control input is replaced by a saturation function, $\{\mathrm{sat}(x) = x$ if $|x| < 1$, or $\mathrm{sgn}(x)$ otherwise$\}$, as is standard practice for sliding mode controllers [34].



(a) Queue length (b) Control input

Fig. 3. Output queue length when there are no model uncertainties.

C. Numerical experiments

To illustrate the tracking performance of the proposed approach, we consider the following sinusoidal reference signal for the desired queue length: $x_d(t) = 8 + 2\sin(0.1t)$, and the parameter values $\mu = 1, C = 1, \tau = 5, a = 0.3$. The parameters represent a server with exponential service time distribution with a mean equal to one; however, the analysis can be adapted to other general service time distributions by appropriately changing the value of C.

Figure 3 shows the control input and the achieved queue length (along with the ultimate bounds) for the case with no model uncertainties. We see that one obtains perfect tracking, and that the control input is continuous (no chattering). Next, we consider the case with model uncertainty. Figure 4 shows the resulting queue length for the three cases that correspond to instances when the uncertainty in the model dynamics is either lesser than, equal to, and greater than the assumed value. In these examples, we assume that the actual mean service rate deviates from the model. The output queue length deviates from the desired value, but lies within the ultimate bound when the actual uncertainty is within the assumed range. We note that the ultimate bound $(x_d \pm a\tau)$ for the queue length that we have obtained is very tight in this example.

The condition that the model uncertainty is within the assumed range $(|\alpha(x,t) - \bar{\alpha}(x,t)| < F(x,t))$ to guarantee uniform ultimate boundedness is a sufficient but not necessary condition. Trajectories could lie within the ultimate bound even if the model uncertainty is greater than the assumed value, if the prediction error over the time-horizon is smaller than $a\tau$. This observation is particularly important when we apply this control framework to discrete stochastic queuing systems, wherein, the instantaneous deviation of the model might be large but the prediction errors over the horizon are small. Finally, the ultimate bound collapses to x_d when there are no time-delays. Therefore, the robust controller guarantees nearly-perfect tracking, even in the presence of model uncertainties, when there are no time-delays.

Next, the control law is tested on a discrete queuing simulator. The simulator advances in discrete time-steps, and the customers released at the source join the queue after a time-delay that corresponds to the travel time. The customers in the queue are served on a first-come-first-serve basis and the service times of the server are sampled from an exponential distribution. From an implementation point of view, we note that the control law (Eq. (28)) provides a continuous release rate into the queue, while the simulator requires as an input the number of customers that are released into the queue. Let Δt be the discrete time-step of the simulator and $u(t_n)$ be the release rate determined from the feedback law (Eq. (28)) at time-step t_n. The number of customers released into the queue at time t_n is represented by $U(t_n)$, and is given by

$$U(t_n) = \left\lfloor \sum_{i=1}^{n} u(t_i)\Delta t - \sum_{i=1}^{n-1} U(t_i) \right\rfloor,$$

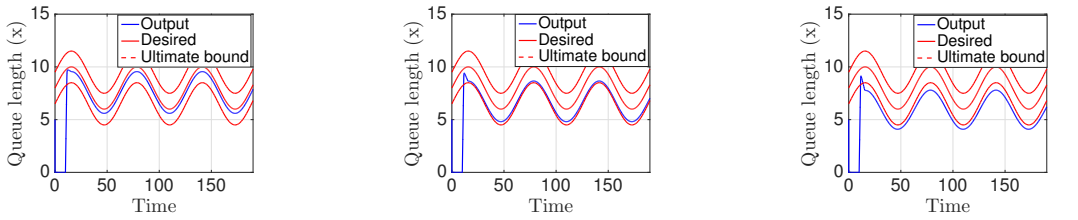
where $\lfloor . \rfloor$ represents the floor function. The objective in the simulation experiments is to release customers into the queue to track a desired queue length. The ultimate bounds obtained from the analytical model, $x_d \pm a\tau$, essentially depend on the error bounds for the state predictions $(a\tau)$. We obtain tighter empirical bounds for the predictor errors using the simulated data and model predictions. Let E be the empirical distribution of the predictor error obtained from the simulation. We then define an approximate ultimate bound from the simulation data as $[x_d - E_1, x_d - E_{99}]$, where, E_k denotes the k^{th} percentile of the error distribution. Fig. 5(a) shows the profile of the desired queue length, achieved queue length and the ultimate bounds for a single realization of the simulation. We can see that the queue length tracks the desired trajectory and stays almost within the bounds. To highlight the fact that the output trajectories mostly remain within the approximate ultimate bounds, we present multiple output trajectories in Fig. 5(b). The ensemble mean and standard deviation over multiple realizations (50 in this case) are shown in Fig. 5(c). The ensemble mean of the queue length closely follows the desired queue length, and the standard deviation of the resultant trajectories is relatively low, as desired.

IV. Control of Queues in Tandem

To illustrate the methodology for a queuing network, we first consider the simplest queuing network: two queues in tandem as shown in Fig. 6. Let x_1 and x_2 be the length of the first and second queue, respectively. Let μ_1 and μ_2 be the mean service rates of the first and second server, respectively. For the present discussion, we assume that the service time distributions are time-invariant. Let the sending rate at the source be $u(t)$. The time taken by a customer to reach the first queue from the source is denoted τ_1, and the time taken to reach the second queue after completing service in the first queue is denoted τ_2. The objective is to control the arrivals into the first queue in order to maintain a desired queue length in the second queue. For this objective to be feasible, the mean service rate of the first server is assumed to be greater than that of the second server. The input rate into the second queue depends on the out-flow rate from the first queue, which is bounded by μ_1. Moreover, the out-flow rate from the second queue depends on μ_2.

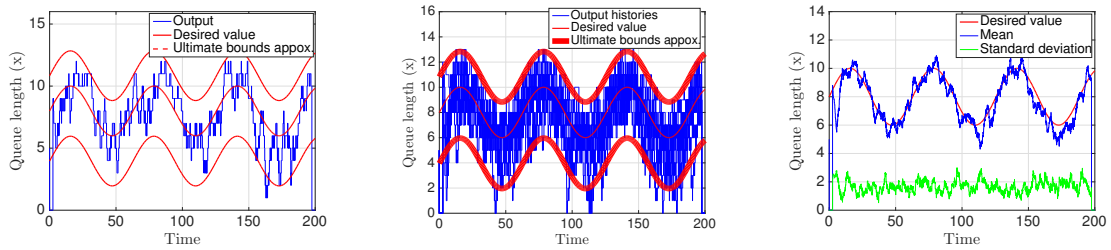
Using the fluid flow model discussed earlier, the queuing model for tandem queues with time-delays is given by the following delay differential equations with appropriate initial conditions:

$$\begin{aligned} \dot{x}_1(t) &= -\mu_1(t)\frac{C_1(t)x_1(t)}{(1+C_1(t)x_1(t))} + u(t-\tau_1) \\ \dot{x}_2(t) &= -\mu_2(t)\frac{C_2(t)x_2(t)}{(1+C_2(t)x_2(t))} + \mu_1(t-\tau_2)\frac{C_1(t-\tau_2)x_1(t-\tau_2)}{(1+C_1(t-\tau_2)x_1(t-\tau_2))} \end{aligned} \quad (32)$$



(a) $\mu_a = 1.1$; $|\alpha(x,t) - \bar{\alpha}(x,t)| < F(x,t)$ (b) $\mu_a = 1.3$; $|\alpha(x,t) - \bar{\alpha}(x,t)| = F(x,t)$ (c) $\mu_a = 1.5$; $|\alpha(x,t) - \bar{\alpha}(x,t)| > F(x,t)$

Fig. 4. Tracking accuracy when the actual service rate (μ_a) is different from the nominal service rate assumed in the model ($\mu = 1$).



(a) Queue length (single realization) (b) Queue length (multiple realizations) (c) Simulation statistics

Fig. 5. Queue length from the queuing simulation with exponential service time distribution.

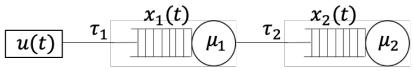


Fig. 6. Schematic of two queues in tandem.

A. Controller for dynamics without propagation delays

As we had done in the case of a single queue, we first develop a robust controller for the non-delayed system and then account for the delay terms through a predictor-based feedback. The model for the queue length without time-delay is:

$$\begin{aligned} \dot{x}_1 &= -\mu_1(t)\frac{C_1(t)x_1}{(1+C_1(t)x_1)} + u(t) \\ \dot{x}_2 &= -\mu_2(t)\frac{C_2(t)x_2}{(1+C_2(t)x_2)} + \mu_1(t)\frac{C_1(t)x_1}{(1+C_1(t)x_1)}. \end{aligned} \tag{33}$$

The objective is to track the second queue length at a desired value considering model uncertainties. The state that has to be tracked, x_2, needs to be differentiated twice to obtain the control input, leading to the following second-order differential equation:

$$\begin{aligned} &\ddot{x}_2 = \bar{\alpha}(\mathbf{x},t) + \bar{\beta}(\mathbf{x},t)u \\ &\bar{\alpha}(\mathrm{x,t}) = -\mu_1^2\frac{C_1^2x_1}{(C_1x_1+1)^3} + \mu_2^2\frac{C_2^2x_2}{(C_2x_2+1)^3} - \mu_1\frac{\mu_2C_1C_2x_1}{(C_2x_2+1)^2(C_1x_1+1)} \\ &\bar{\beta}(\mathbf{x},t) = \frac{\mu_1C_1}{(1+C_1x_1)^2}, \end{aligned}$$

where $\mathbf{x} = [x_1, x_2]^T$. We assume that the dynamics for the actual queue length is of the form $\ddot{x}_2 = \alpha(\mathbf{x},t) + \beta(\mathbf{x},t)u$, where $\alpha(\mathbf{x},t)$ and $\beta(\mathbf{x},t)$ are unknown functions. However, we assume that the error between the model and the actual dynamics is bounded. We assume the following form for the error bounds:

$$|\alpha(\mathbf{x},t) - \bar{\alpha}(\mathbf{x},t)| \le F(\mathbf{x},t), \tag{34}$$

$$\tfrac{1}{c}\bar{\beta}(\mathbf{x},t) \le \beta(\mathbf{x},t) \le c\bar{\beta}(\mathbf{x},t),\ c > 1. \tag{35}$$

Since errors arise primarily due to uncertainties in the service times, we consider the following form for $F(\mathbf{x},t)$:

$$F(\mathbf{x},t) = a_1\frac{C_1^2x_1}{(C_1x_1+1)^3} + a_2\frac{C_2^2x_2}{(C_2x_2+1)^3} + a_3\frac{C_1C_2x_1}{(C_2x_2+1)^2(C_1x_1+1)}$$

where a_i, $i = 1,2,3$ are constants that determine the level of uncertainty. The goal is to design a sliding controller to have x_2 track a desired queue length, $x_{2,d}(t)$. Assume that the desired trajectory is continuous and twice-differentiable. The sliding variable (s) is defined in terms of the tracking error, $e = x_2 - x_{2,d}$, as $s = \dot{e} + \lambda e,\ \lambda > 0$.

$$\dot{s} = \ddot{e} + \lambda\dot{e} = \ddot{x}_2 - \ddot{x}_{2,d} + \lambda\dot{e} = \alpha + \beta u - \ddot{x}_{2,d} + \lambda\dot{e} \tag{36}$$

The best approximation of the control input that would achieve $\dot{s} = 0$ is given by $\hat{u} = \bar{\beta}^{-1}(\ddot{x}_{2,d} - \lambda\dot{e} - \bar{\alpha})$. Consider the control law of the form

$$u = \hat{u} - k\ \mathrm{sgn}(s). \tag{37}$$

The gain parameter (k) is determined from the sliding condition (Eq. (10)) to guarantee asymptotic tracking. Using the definitions for s, the control law Eq. (37), and the sliding condition (Eq. (10)), we get:

$$\begin{aligned} s\dot{s} &= s\Big(\alpha + \beta(\hat{u} - k\ \mathrm{sgn}(s)) - \ddot{x}_{2,d} + \lambda\dot{e}\Big) \\ &= s\Big(\alpha + \beta(\bar{\beta}^{-1}(\ddot{x}_{2,d} - \lambda\dot{e} - \bar{\alpha}) - k\ \mathrm{sgn}(s)) - \ddot{x}_{2,d} + \lambda\dot{e}\Big) \\ &= s\Big(\alpha - \beta\bar{\beta}^{-1}\bar{\alpha}\Big) + s(-\ddot{x}_{2,d} + \lambda\dot{e})(1 - \beta\bar{\beta}^{-1}) - \beta k|s| \\ &= s\Big(\alpha - \bar{\alpha}\Big) + (\bar{\alpha} - \ddot{x}_{2,d} + \lambda\dot{e})(1 - \beta\bar{\beta}^{-1}) - \beta k|s| \\ &= s\Big(\alpha - \bar{\alpha}\Big) - \bar{\beta}\hat{u}(1 - \beta\bar{\beta}^{-1}) - \beta k|s| \end{aligned}$$

Consider $k \geq \bar{\beta}^{-1}c(F+\eta) + (c-1)|\hat{u}|$, then,

$$s\dot{s} \leq s\Big(\alpha - \bar{\alpha}\Big) - \hat{u}(\bar{\beta} - \beta) - \beta\Big(\bar{\beta}^{-1}c(F+\eta) + (c-1)|\hat{u}|\Big)|s| \quad (38)$$

$$s\dot{s} \leq s\Big(\alpha - \bar{\alpha}\Big) - \Big(\beta\bar{\beta}^{-1}c(F+\eta)\Big)|s| - \hat{u}\beta(\bar{\beta}\beta^{-1} - 1) - \beta(c-1)|\hat{u}||s| \quad (39)$$

From the assumptions on the error bounds (Eqs. (34)-(35)), we obtain the following conditions: $\beta\bar{\beta}^{-1}c \geq 1$, $(c-1) \geq (\bar{\beta}\beta^{-1} - 1)$, $(\alpha - \bar{\alpha}) \leq F$, and hence Eq. (39) implies $s\dot{s} \leq -\eta|s|$. Therefore, the gain parameter (k) if chosen as per the above condition satisfies the sliding condition (Eq. (10)) to guarantee asymptotic tracking. To reduce the effect of chattering and to account for the non-negativity of the control input, we modify the feedback law (as done in the single queue case) as follows:

$$u = \big(\hat{u} - k\ \text{sat}(s)\big).^{+}$$

As a result of saturating the control law, we lose the tracking guarantees obtained from the sliding mode controller. In the case of a single queue, we were able to obtain conditions for tracking even with controller saturation. While the analysis that was presented for the case of a single queue is difficult to extend to the case of multiple queues, we show that the feedback law performs well in numerical experiments.

1) Evaluating controller performance: To evaluate the performance of the feedback controller, we consider the case in which the service time of the servers follows an exponential distribution. We first illustrate the performance of the controller by applying the feedback law to the model equations with uncertain service rates. The model parameters used to develop the feedback control law are $\mu_1 = 2$, $\mu_2 = 1$, $C_1 = C_2 = 1$. Here, to show the performance under model uncertainty, the actual service rates are considered to be 1.2 times the nominal service rates that were used to derive the feedback controller. The state derivative that is required for the feedback law is estimated using the model equations. Figure 7 shows the control input and the resulting queue length for the case in which the desired queue length is sinusoidal ($x_{2,d} = 8 + 2\sin(0.1t)$). The controller is seen to offer good tracking even though the assumed service rate differs from the actual service rate.

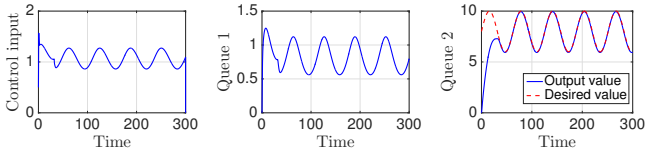
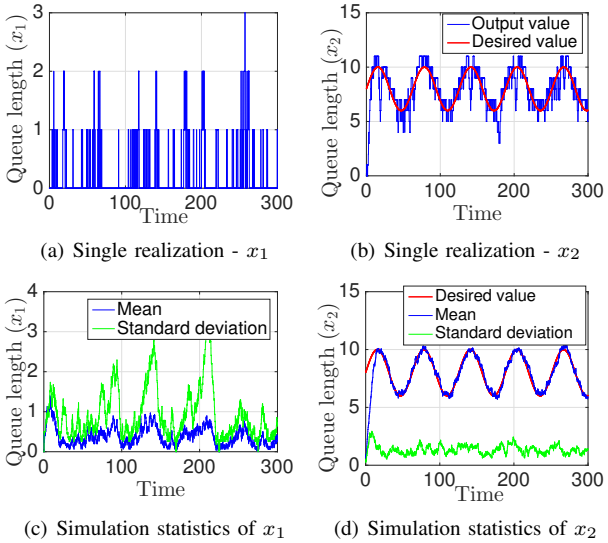


Fig. 7. Controller performance for tracking the length of the second queue.

Next, the feedback controller is tested on a stochastic queuing simulator. The continuous sending rate obtained from the control law is converted into discrete sending times in the same way as done earlier for the single queue case. In addition, we drop the state dependence of the nominal value of the control gain for better transient performance ($\bar{\beta}(x) = \bar{\beta} = 3$). Figures 8(a)-8(b) show a single realization of the first and second queue length obtained from the simulations. We see that the length of the second queue remains close to the desired



(a) Single realization - x_1 (b) Single realization - x_2

(c) Simulation statistics of x_1 (d) Simulation statistics of x_2

Fig. 8. Single realization and statistics over multiple realizations of the queue length. Simulation parameters: $\mu_1 = 2$, $\mu_2 = 1$, $\lambda = 23$, $\eta = 0.5$, $\Delta t = 0.1, x_{2,d} = 8 + 2\sin(0.1t)$

value. The mean and standard deviation of the queue length calculated from 30 realizations are shown in Figs. 8(c)-8(d).

2) Comparison to exact solutions: The solution obtained using the sliding mode controller is compared with an exact solution obtained using a Markov Decision Process (MDP). We can represent the queuing process as a Markov chain (Fig. 9) for the case in which the service time is exponentially distributed and there are no time-delays ($\tau_1 = \tau_2 = 0$). The state of the Markov chain is $X = (i, j)$, with i being the length of the first queue and j being the length of the second queue. The control input, u, determines the number of customers released to the first queue. The transition probability (P) of the Markov chain conditioned on the control input, $u = k$, is as follows:

$$p_{(i,j)\to(l,m)|u=k} = \begin{cases} \mu_1\Delta t\delta_i, \text{if } (l,m) = (i-1+k, j+1) \\ \qquad\qquad \&(i \neq 0 | k > 0) \\ \mu_2\Delta t\delta_j, \text{if } (l,m) = (i+k, j-1)\&(j \neq 0) \\ 1 - (\mu_1\delta_i + \mu_2\delta_j)\Delta t, \text{if } (l,m) = (i+k, j) \\ 0, \text{ otherwise.} \end{cases}$$

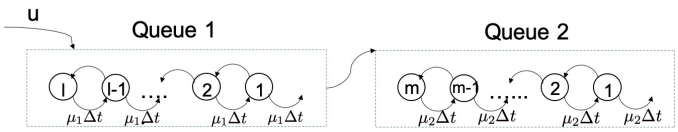


Fig. 9. Markov chain representation for tandem queue

Here, $\delta_m = 0$ if $m = 0$, or 1 otherwise. The aim of the control problem is to track the length of the second queue at some desired value (x_d). Here, we consider the desired value to be a constant to obtain a static policy. The cost function ($C(X,u)$) for the MDP is considered to be the following:

$$C(X,u) = (j - x_d)^2.$$

The cost function penalizes the deviation from the desired queue length. The $s-horizon$ cost under a stationary control policy, π, is given by:

$$J_s(X,\pi) = E_\pi\Big[\sum_{n=0}^{s-1}\gamma^n C(X_n,u_n)|X_o = X\Big],$$

where E_π denotes the expectation over the path of the process under policy π, γ is the discount factor, and (X_n, u_n) denotes the state and control input pair at time $n\Delta t$. The $infinite-horizon$ cost under policy π is given by

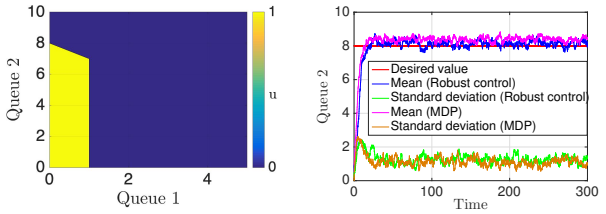
$$J(X,\pi) = \lim_{s\to\infty} J_s(X,\pi),$$

and the optimal stationary policy π^* satisfies the Bellman equation [39]:

$$J^*(X) = \min_u \Big[C(X,u) + \gamma\sum_{X'} P_{X\to X'}J^*(X')\Big].$$

The optimal policy, that is the solution to the above equation, is obtained using value iteration. Fig. 10(a) shows the control policy as a function of the queue length for $x_d = 8,\ \mu_1 = 2,\ \mu_2 = 1$. The resulting control policy is to release a customer only when there is not more than one customer in the first queue fewer than the desired value in the second queue. The mean and standard deviation of the second queue length obtained from the stochastic simulations (over 30 sample realizations) with the MDP control policy and the sliding mode controller are shown in Fig. 10(b). The mean queue length obtained from both the control methodologies are close to the desired value, while the standard deviation of the queue length obtained from the sliding controller is slightly more than the optimal solution obtained from the MDP.

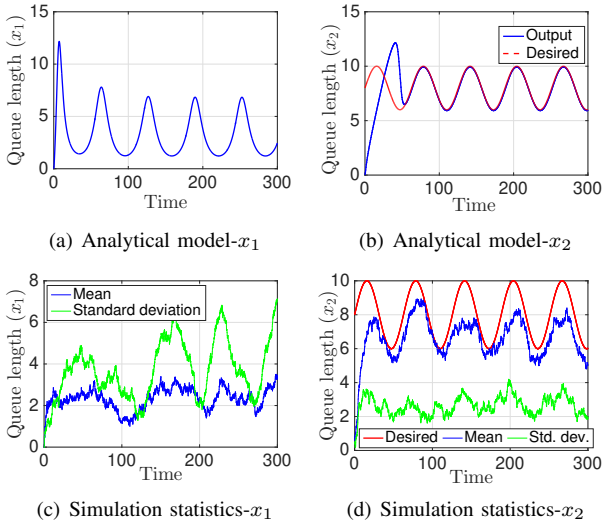
The main drawback of an MDP-based approach is its computational complexity, since the Markov chain representation for queuing networks results in a large state space. Additionally, one may have to include additional state variables to represent a queuing system that is inherently non-Markovian (e.g., non-exponential service time distribution and time-delays) as a Markov process, further increasing the size of the state space. Although there are efficient approximate techniques for solving large-scale MDPs, their computational complexity is significantly higher than that of a robust control approach, which involves the numerical integration of the system dynamics, and an algebraic evaluation of the control law.



(a) Control input as a function of the queue length (b) Mean and variance of the queue length

Fig. 10. Control policy and queue length obtained from simulation ($x_d = 8,\ \mu_1 = 2,\ \mu_2 = 1,\ \Delta t = 0.1,\ \gamma = 0.99$).

3) Comparable service rates: In the discussion so far, we considered the mean service rate of the first server to be significantly greater than that of the second server. As a result, the length of the first queue is negligible compared to the second queue. We now consider the case when the mean service rate of the second server is only slightly larger than that of the first one, and there is consequently significant queuing in the first queue. Fig. 11 shows the queue length obtained with the feedback controller for the analytical model as well as for the discrete queuing simulations (50 trials). The length of the first queue is longer than in the previous case, as expected. We also notice perfect tracking for the analytical model. However, there is some tracking error even for the mean queue length in the simulations. This is primarily due to the fact that when the first server has high utilization, then the outflow rate from the first queue (i.e., the inflow rate into the second queue) is determined by the service rate of the first server, making it more difficult to track the second queue length.



(a) Analytical model-x_1 (b) Analytical model-x_2

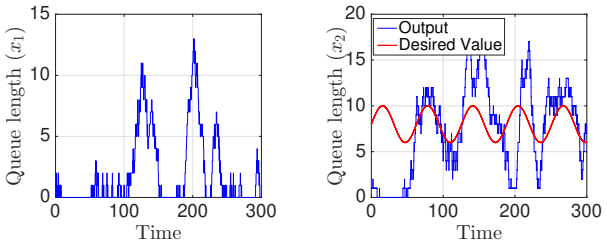
(c) Simulation statistics-x_1 (d) Simulation statistics-x_2

Fig. 11. Queue length for tandem queues without time-delays: $x_{2,d} = 8 + 2\sin(0.1t)$, $\mu_1 = 1.25$, $\mu_2 = 1$, $\lambda = 0.3$, $\eta = 0.5$, $\Delta t = 0.1$.

4) Feedback control without information on the length of the first queue: Although the second queue is the primary bottleneck, controlling the second queue without considering the first queue length information in the feedback leads to poor tracking performance. This effect is more pronounced when the two service rates are comparable. Fig. 12 shows the queue length obtained when only the second queue length is considered in the feedback using the controller developed for a single queue. In this case, the mean service rate of the first server is 1.25 and the mean service rate of the second server is 1. We see, in particular, that the standard deviation of the second queue length is higher in Fig. 12 than in Fig. 11.

B. Controller for tandem queues with time-delays

Eq. (32) presents the model for tandem queues with time-delays. The system has delays in the state and the control input, and is forward complete. We can use a predictor-based feedback controller for tracking the queue length. Using the feedback controller for the non-delayed dynamics presented



(a) Single realization - x_1 (b) Single realization - x_2

(c) Simulation statistics - x_1 (d) Simulation statistics - x_2

Fig. 12. Queue length for tandem queue obtained by using a controller without considering the first queue length information in the feedback.

earlier (Eq. 37), we can write the predictor-based feedback controller for the time-delayed system as follows:

$$u(t) = \left(\bar{\beta}^{-1}(\mathbf{x_p})[\hat{u}(\mathbf{x_p}) - k(\mathbf{x_p})\ \mathrm{sgn}(s(\mathbf{x_p}))]\right)^+$$

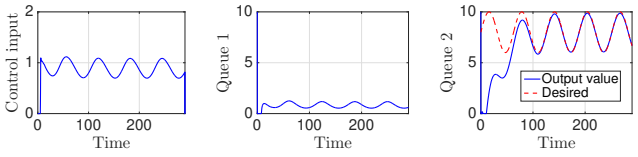
Here, $\mathbf{x_p}(t) = (P_{1,t}(t+\tau_1), P_{2,t}(t+\tau_1+\tau_2))$, with $P_{i,t}(.)$ defined as follows:

$$P_{1,t}(t+\tau_1) = x_1(t) + \int_{t-\tau_1}^{t} \left(-\mu_1 \frac{C_1 P_{1,t}(\theta+\tau_1)}{(1+C_1 P_{1,t}(\theta+\tau_1))} + u(\theta) \right) d\theta,$$

$$P_{2,t}(t+\tau_1+\tau_2) = x_2(t) + \int_{t-\tau_2-\tau_1}^{t} \left(-\mu_2 \frac{C_2 P_{2,t}(\theta+\tau_1+\tau_2)}{(1+C_2 P_{2,t}(\theta+\tau_1+\tau_2))} \right) d\theta + \int_{t-\tau_2}^{t} \left(\mu_1 \frac{C_1 P_{1,t}(\theta+\tau_1)}{(1+C_1 P_{1,t}(\theta+\tau_1))} \right) d\theta + \int_{t-\tau_1}^{t} \left(\mu_1 \frac{C_1 x_1(\theta)}{(1+C_1 x_1(\theta))} \right) d\theta$$

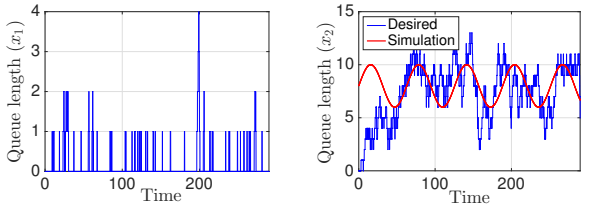
with appropriate initial conditions for the state predictors, $P_{1,t}(t) = x_1(t)$ and $P_{2,t}(t) = x_2(t)$. If the server parameters are time-varying, the integrands in the above equations need to be modified to account for this factor.

The performance of the predictor-based control law for tracking the second queue length is shown in Fig. 13. We consider the following model parameters: $\mu_1 = 2$, $\mu_2 = 1$, $\tau_1 = \tau_2 = 2.5$, and $C_1 = C_2 = 1$. The control parameters (λ, η) are picked to avoid overshoot. We see that output queue length tracks the desired sinusoidal value after a small initial transient. Fig. 14 shows the tracking performance obtained using discrete stochastic queuing simulations (statistics computed over 30 independent realizations). The ensemble mean queue length of the second queue is seen to closely match the desired value. However, as expected, the queue length has higher variability than in the case without time-delays.

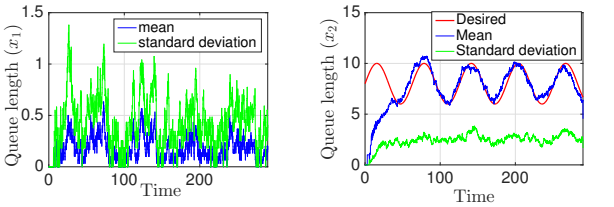


(a) Control input (b) Queue 1 (c) Queue 2

Fig. 13. Tandem queues with time-delays ($\mu_1 = 2$, $\mu_2 = 1, \tau_1 = \tau_2 = 2.5, \eta = 0.1, \lambda = 0.04, x_{2,d} = 8 + 2\sin(0.1t)$).



(a) Single realization of first queue (b) Single realization of the second queue



(c) Statistics over multiple realization- first queue (d) Statistics over multiple realization- second queue

Fig. 14. Queue length for tandem queue with time-delays.

V. Application to Airport Surface Operations

We apply the robust controller to a queuing network model of airport surface traffic movements.

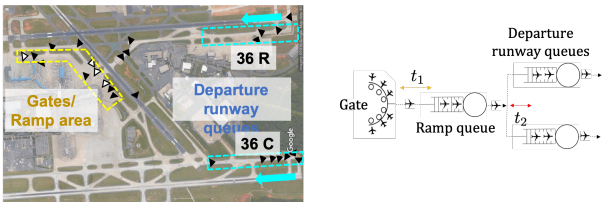
A. Background

Congestion at airports manifests as long queues of aircraft on the surface (Fig. 15(a)), resulting in excessive fuel burn for the aircraft waiting in these queues. Departure metering is widely-considered a promising approach to mitigating surface congestion. The key idea behind departure metering is to tactically hold the departures at the gate and to release them appropriately so that they pass through smaller queues during taxi, thereby reducing taxi-out times and fuel consumption.

B. Model for the airport surface traffic

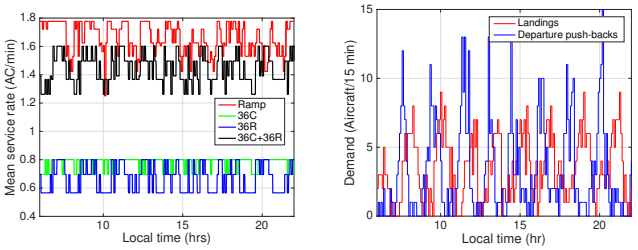
The traffic movement (departures and arrivals) in congested airports can be represented using a queuing network model. The analysis in this paper uses a queuing network model that we previously developed for Charlotte Douglas International airport (CLT) [23]. Fig. 15(a) shows a snapshot of surface traffic at CLT. The black triangles represent aircraft taxiing-out (departures), and the white triangles represent the flights taxiing-in (arrivals). Departing aircraft form long queues in the ramp area (close to the airport terminals) as well as near the departure runways. The movement of departures can be represented as a queuing network as shown in Fig. 15(b). The

departures pass through a ramp queue and one of the two runway queues (36C or 36R), before they takeoff.



(a) Layout of CLT with a focus on departure operations. (b) Queuing representation for the departure operations at CLT.

Fig. 15. Airport layout and queuing representation.



(a) Mean service rate of ramp and runway servers (b) Departure and arrival demand

Fig. 16. Variation of the mean service and demand for a typical good weather day (07/12/2015).

The corresponding queuing model for the departure process is given by the following set of delay differential equations:

$$x_s(t) = x_{s_1}(t) + x_{s_2}(t)$$
$$\dot{x}_{s_i}(t) = -\mu_s(t)\frac{C_s(t)x_{s_i}(t)}{1 + C_s x_s(t)} + u_{r_i}(t - t_1), \quad i = 1, 2;$$
$$\dot{x}_{r_i}(t) = -\frac{\mu_{r_i}(t)C_{r_i}x_{r_i}(t)}{1 + C_{r_i}x_{r_i}(t)} + \frac{\mu_s(t - t_2)C_s(t - t_2)x_{s_i}(t - t_2)}{1 + C_s(t - t_2)x_s(t - t_2)}.$$

Here, x_{s_i} represents the number of aircraft in the ramp queue headed to runway i, x_{r_i} represents the number of aircraft in the i^{th} departure runway queue, (C_s, μ_s) are the server parameters of the ramp server, (C_{r_i}, μ_{r_i}) are server parameters of the i^{th} departure runway server, t_1 is the average unimpeded time from the gate to the spot (exit of the ramp), t_2 is the average unimpeded time from the spot to the departure runway, and u_{r_i} is the pushback rate to the i^{th} departure runway. The pushback rate is the number aircraft in a given time interval (5-min, in this paper) that pushback from the gate and enter the system.

The service time distributions were obtained from operational data. For the runway server, the service time distribution depends on the number arrivals landing on the runway and the airport weather (instrument or visual meteorological condition). For the ramp server, the service time distribution depends on the number of arrivals (flights taxiing-in) in the ramp queue [40].

We assume that the service time distributions are piecewise constant over 5-min intervals. Fig. 16(a) illustrates the variation of the mean service rate for the ramp and runway servers on a typical good weather day. The observed fluctuation in the mean service rate is because of the variation in the arrival traffic level over the course of the day. The mean service rate for 36R varies more than 36C because it handles more number of arrivals. The service rate of the ramp server is slightly more that the sum of the service rates of the two runway servers, confirming that the runway is the critical bottleneck. The arrival and departure demand at the airport also fluctuate through the day (Fig. 16(b)), highlighting the need for a dynamic policy to reduce congestion.

The queue lengths can be predicted by integrating the model equations with appropriate initial conditions. The taxi-out time can be estimated as the sum of the unimpeded time plus the wait time in these queues. A detailed model validation has been presented in our earlier paper [23]: The mean taxi-out time prediction error is -1.4 min, and the mean absolute error is 4.4 min, for a mean taxi-out time of 20.1 min.

C. Departure metering to control runway queue lengths

In the absence of departure metering, pilots pushback whenever they are ready to leave. There are times during the day when many aircraft pushback around the same time, resulting in the formation of large queues, and leading to high taxi-out times. The objective of departure metering is to tactically control departure pushbacks in order to reduce taxi-out time without underutilizing the runway. The runway is said to be underutilized if the departures are held longer than necessary at the gate, causing avoidable take-off delays and a decrease in airport throughput. While having a queue length of zero will result in departing aircraft taking off without waiting, it will almost certainly lead to under-utilization of the runway, because of uncertainties. We therefore try to maintain a (small) target runway queue length.

We present a methodology, based on the discussion so far, for determining the pushback times to achieve desired runway queue lengths. The queuing model for the airport surface leads to a multi-input-multi-output system, unlike the single-input-single-output systems discussed earlier. The outputs to be tracked are the two runway queue lengths $(x_{r_1},\ x_{r_2})$ and the inputs correspond to the pushback rate to each runway (u_{r_1}, u_{r_2}). The inputs do not directly appear in the equations of the output dynamics. Differentiating the outputs twice yields the following second-order differential equations (for the non-delayed dynamics):

$$\ddot{\mathbf{y}} = \bar{\alpha}(\mathbf{x}, t) + \bar{\beta}(\mathbf{x}, t)\mathbf{u}$$

Here, $\mathbf{x} = [x_{s_1}, x_{s_2}, x_{r_1}, x_{r_2}]^T$ and $\mathbf{u} = [u_{r_1}, u_{r_2}]^T$. The functions $\bar{\alpha}(\mathbf{x}, t) \in R^2$ and $\bar{\beta}(\mathbf{x}, t) \in R^{2\times 2}$ are given by

$$\bar{\alpha}(\mathbf{x}, t) = \begin{bmatrix} \frac{\mu_{r_1}^2 C_{r_1} x_{r_1}}{(1+C_{r_1}x_{r_1})^2} - \frac{\mu_{r_1}\mu_s C_s x_{s_1}}{1+C_s x_s} - \frac{\mu_s^2 C_s^2 x_{s_1}}{(1+C_s x_s)^2} \\ \frac{\mu_{r_2}^2 C_{r_2} x_{r_2}}{(1+C_{r_2}x_{r_2})^2} - \frac{\mu_{r_2}\mu_s C_s x_{s_2}}{1+C_s x_s} - \frac{\mu_s^2 C_s^2 x_{s_2}}{(1+C_s x_s)^2} \end{bmatrix},$$

$$\bar{\beta}(\mathbf{x}, t) = \begin{bmatrix} \mu_s \frac{C_s(1+C_s x_{s_2})}{(1+C_s x_s)^2} & -\mu_s \frac{C_s^2 x_{s_1}}{(1+C_s x_s)^2} \\ -\mu_s \frac{C_s^2 x_{s_2}}{(1+C_s x_s)^2} & \mu_s \frac{C_s(1+C_s x_{s_1})}{(1+C_s x_s)^2} \end{bmatrix}.$$

The terms containing the derivatives of the server parameters in the queue model are ignored (note that the service time

distribution is considered to be a piecewise constant). As done earlier, we shall assume that the actual dynamics is of the following form $\ddot{\mathbf{y}} = \alpha(\mathbf{x},t) + \beta(\mathbf{x},t)\mathbf{u}$. Here, $\alpha(.)$ and $\beta(.)$ are unknown functions, with the following error bounds:

$$|\alpha_i(\mathbf{x},t) - \bar{\alpha}_i(\mathbf{x},t)| < F_i(\mathbf{x},t),\ i = 1,2.$$
$$\beta(\mathbf{x},t) = (\mathbf{I}+\Delta)\bar{\beta}(\mathbf{x},t),\quad |\Delta_{ij}| < D_{ij};\ i,j = 1,2.$$

Motivated by the fact that the uncertainties arise primarily in the service times in the actual system, we consider the following form for $\mathbf{F}$:

$$\mathbf{F}(\mathbf{x},t) = \begin{bmatrix} a_1 \frac{C_{r_1} x_{r_1}}{(1+C_{r_1}x_{r_1})^2} + a_2 \frac{C_s x_{s_1}}{1+C_s x_s} + a_3 \frac{C_s^2 x_{s_1}}{(1+C_s x_s)^2} \\ a_4 \frac{C_{r2} x_{r_2}}{(1+C_{r_2}x_{r_2})^2} + a_5 \frac{C_s x_{s_2}}{1+C_s x_s} + a_6 \frac{C_s^2 x_{s_2}}{(1+C_s x_s)^2} \end{bmatrix}.$$

The a_is and D_{ij}s are design parameters that need to be picked appropriately depending on the level of uncertainty in the system. The goal is to track the runway queue lengths at desired values ($x_{r_{id}}$, for each runway i). The sliding variable ($\mathbf{s} \in \mathbb{R}^2$) is defined in terms of the tracking error ($\mathbf{e}$) as follows:

$$s_i = \dot{e}_i + \lambda e_i;\ e_i = x_{r_i} - x_{r_{id}};\ i = 1,2.$$

Perfect tracking is guaranteed for the model dynamics if the sliding variable satisfies the following sliding condition (note similarity to Eq. (10)):

$$\frac{1}{2}\frac{d}{dt}s_i^2 \le -\eta_i|s_i|,\ \eta_i > 0 \implies s_i\dot{s}_i \le -\eta_i|s_i| \tag{40}$$

Consider the control input of the form

$$\mathbf{u} = \bar{\beta}^{-1}(\bar{\mathbf{u}} - \mathbf{k} \odot \text{sgn}(\mathbf{s})), \tag{41}$$

where $\bar{\mathbf{u}} = (-\bar{\alpha}(\mathbf{x},t) + \ddot{x}_{d,r_{id}} - \lambda\dot{e}_i)$ and $\odot$ represents element-wise multiplication operation. The value of $\mathbf{k}$ is obtained from the sliding condition to achieve perfect tracking (Eq. (40)), along with constraints on the error bounds. After some algebraic manipulation, we get:

$$F_i(\mathbf{x},t) + \sum_{j=1}^{2} D_{ij}|\bar{u}_j(\mathbf{x},t)| - \sum_{j=1,j\neq i}^{2} D_{ij}k_j + \eta_i \le (1-D_{ii})k_i.$$

A particular value of $\mathbf{k}$ is chosen by solving the following linear equation:

$$(1-D_{ii})k_i + \sum_{j=1,j\neq i}^{2} D_{ij}k_j = F_i + \sum_{j=1}^{2} D_{ij}|\bar{u}_{rj}(\mathbf{x},t)| + \eta_i.$$

The control law in Eq. (41) is guaranteed to track the desired queue length if the value for $\mathbf{k}$ is chosen such that it satisfies the above equation. However, the control input needs to be saturated at zero since the pushback rate cannot be negative. The time delays in the original model dynamics is handled using a predictor-based feedback. We consider the following substitutions in the delay-free feedback law to account for time-delays: $x_{s_i}(t) \to P_{s_i,t}(t+t_1)$, $(\mu_s(t), C_s(t)) \to (\mu_s(t+t_1), C_s(t+t_1))$, $x_{r_i}(t) \to P_{r_i,t}(t+t_1+t_2)$, $x_{r_i,d}(t) \to x_{r_i,d}(t+t_1+t_2)$, $(\mu_{r_i}(t), C_{r_i}(t)) \to (\mu_{r_i}(t+t_1+t_2), C_{r_i}(t+t_1+t_2))$. Here, $P_{y,t}(t+z)$ refers to the prediction of state $y(t+z)$ computed at time t. The derivatives of the states present in the control input are also transformed in a similar way. We use the delay differential equations to compute the predictions of the derivatives given the predictions of the states. Guarantees for tracking are no longer possible, since we have saturated the control input and considered a predictor-based feedback to account for the time-delays. Nevertheless, we show through numerical simulations that the control law performs well.

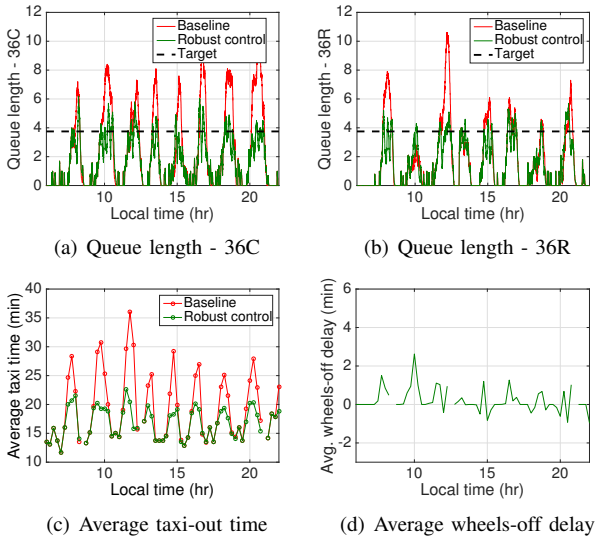
Pushback rate decisions need to be converted into flight-specific hold decisions. The day is divided into 5-min intervals. At the beginning of each interval, t, the pushback rate is determined for $t + T_p$, where T_p is the planning horizon. A planning horizon is included to improve predictability in the system. The planning horizon introduces additional input delay in the dynamics. The number of aircraft that can be released during each 5-min window (n) is determined from the pushback rate. The first n aircraft in the 5-min window are released as per the control decision, and the remaining aircraft are postponed to the beginning of the next time window, awaiting decision for release. This approach specifies the flights that need to be released in the $[t+T_p, t+T_p+5]$ time-window, and postpones the remaining flights to the next time-window.

D. Evaluating the performance of departure metering

1) Simulation environment: The performance of the departure metering algorithm is evaluated using simulations of the airport surface. The simulator is based on the discrete version of the queuing network model for the airport. The empirical service time distributions are functions of the airport weather, fleet mix and traffic. The service times are randomly sampled from the empirical distributions and the simulations are repeated multiple times to obtain consistent statistics (a Monte Carlo simulation with 10 runs). The simulator is validated by comparing the baseline taxi-out time predictions (with no departure metering) with the historical operational data for 6,474 flights from the test set.

2) Benefits of departure metering: We apply the controller developed earlier to compute the release rate for departure flights to maintain a certain target queue length at the runway. Figs. 17(a)-17(b) show the mean queue length for the two runways over multiple realizations of the simulation obtained using the pushback control strategy as well as the baseline case. We see that the queue length with departure metering is close to the desired target value during periods of high demand (when there is a large queue in the baseline case). The queue lengths are significantly lower than the target value during periods of low demand. The figure clearly shows the banking strategy adopted by airlines at their hubs.

The target queue length is set to 3.75 for each runway queue; this value is determined such that the runway does not get starved (under-utilized). Runway under-utilization can occur due to low values of target queue length, and result in unnecessary gate-holds and takeoff delays. On the other hand, higher values of target queue length lead to decreased benefits in terms of taxi-out time savings. The optimal target queue length is determined through a parametric simulation analysis that ensures an average wheels-off delay of close to zero,



(a) Queue length - 36C (b) Queue length - 36R

(c) Average taxi-out time (d) Average wheels-off delay

Fig. 17. Average queue length, taxi-out time and wheels-off delay from the simulations of the departure metering strategy for a typical day (May 7, 2015).

while still maximizing taxi-out time reduction. Figure 17(c) shows the average taxi-out time of flights in the baseline and departure metering scenarios. We see that the peaks in the taxi-out time are reduced with departure metering. The fuel savings from departure metering are approximately 10.2 kg/min at CLT. Moreover, the departure metering strategy effectively holds the flights at the gate, and does not result in significant wheels-off delays (Fig. 17(d)).

3) Comparison to a heuristic departure metering strategy: We compare the performance of the proposed robust control approach with a heuristic that models NASA's Airspace Technology Demonstration - 2 (ATD-2) logic. ATD-2 has been tested in field trials at CLT since November 2017. The departure metering logic in ATD-2 computes a gate-hold time for each flight based on its predicted taxi-out time. The gate-hold time assigned to each flight is the predicted wait time in queue for that flight minus a pre-specified excess queue time buffer [32]. We use the queuing model presented earlier to obtain the taxi-out time predictions for each flight. The underlying idea is to transfer the predicted wait time in the queues to a gate-hold time, thereby saving fuel. The excess queue time buffer helps accommodate errors in the taxi-out time prediction. The optimal buffer is chosen such that it yields the maximum reduction in taxi-out time, while ensuring that there is no significant change in wheels-off time. The optimal excess queue time buffer for a 20-min planning horizon was determined to be 7 min for CLT [40].

Tab. I shows some key statistics comparing simulations of the two departure metering approaches over three days of operations (6 AM-10 PM local time). The reduction in taxi-out time is higher with the robust control approach compared to the heuristic based on the ATD-2 logic, and results in no significant wheels-off delay. It is worth noting that a smaller fraction of flights are held for a larger duration with the robust controller compared to the heuristic.

The maximum computation time for a 5-min decision window is less than 30 ms for both the approaches, allowing for practical implementation. The computation time slightly varies by time window because of varying demand and queue length (example, for the robust control approach, the average computational time for a 5-min window is 2 ms whereas the maximum value is 28 ms). These computations were performed with MATLAB on a computer with 2.8 GHz Intel Core i7 processor.

TABLE I
COMPARISON OF SIMULATIONS OF DEPARTURE METERING APPROACHES FOR CLT.

Mean statistics	ATD-2-based heuristic	Robust control
Taxi-out reduction	2.6 min	2.89 min
Hold time	2.71 min	2.97 min
Wheels-off delay	0.10 min	0.08 min
Proportion of flights held	63%	35%
Hold time of flights held	4.33 min	8.40 min

VI. DISCUSSIONS

In this section, we present a few extensions and promising directions for future research.

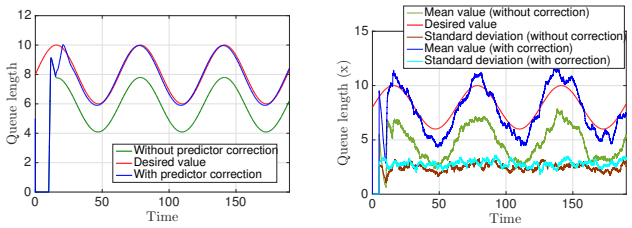
A. Correcting for the prediction errors

The model uncertainties lead to tracking errors for time-delay systems while using a predictor-based feedback controller (as seen earlier for tracking a single queue in Fig. 4). We propose a heuristic, based on the classical Smith Predictor linear time-delay systems [41], to improve the tracking performance in the presence of uncertainties. The Smith predictor provides feedback control not just using the predicted state, but also accounts for errors between the output and model predictions. Similarly, we account for the state prediction errors by estimating a correction factor online using past predictions and observations. Instead of using model predictions as inputs to the feedback law, we use corrected predictions based on past observations. Let $\hat{P}_k$ be the corrected value of the state predictor, P_k, at time t_k (note a change in convention for the subscript, k here refers to the time index. We will illustrate the method for discrete version of the dynamics as used in the implementation of the controller). We assume $\hat{P}_k = w_k P_k$. The correction factor, w_k, is determined online using weighted recursive least squares based on past model predictions and current state observations. The recursive algorithm to obtain w_k is as follows:

$$\begin{aligned} \mathrm{K}_k &= \mathrm{Q}_{k-1} P_k (\lambda_f + Q_{k-1} P_k^2)^{-1} \\ \mathrm{Q}_k &= (\mathrm{Q}_{k-1} - K_k P_k Q_{k-1})/\lambda_f \\ \mathrm{w}_k &= \mathrm{w}_{k-1} - K_k (w_k P_k - x_k) \end{aligned}$$

Here, $\lambda_f \in (0, 1]$ is an exponential discount factor; a smaller value corresponds to higher weighting of recent prediction errors compared to the past errors. Since the actual prediction error can be determined only after waiting for the system delay, the correction factor (w_k) corresponds to past predictions. However, we ignore this lag. This method is effective for handling prediction errors that have a fixed bias or are slowly varying. To illustrate the improvement in tracking performance, we consider queue length tracking for a single

queue. The mean service rate of the of the actual dynamics is considered to be 1.5 times the service rate used in the model to design the controller. Fig. 18 shows the queue length obtained with a corrected estimate for the predictor and it is compared with the value obtained without the correction. The results are shown both for the analytical model (Fig.18(a)) and the simulation (Fig.18(b)). We see significant improvements in the tracking performance compared to the case where naïve state predictions were used (Fig. 4(c)). A higher value of the forgetting (discount) factor is used in the simulation ($\lambda_f = 1$) than in the analytical model ($\lambda_f = 0.98$), to account for the stochasticity in the simulations. An alternative approach to deal with model uncertainty is to use a robust adaptive control approach to correct for the model parameters [34]. However, one would still have to deal with delayed prediction errors.



(a) Analytical model (b) Statistics from simulation

Fig. 18. Correcting for the prediction errors using recursive least squares ($\mu = 1,\ \mu_a = 1.5,\ C = C_a = 1$).

B. Multiple arrival sources into the queuing network

This paper, motivated by the congestion control on the airport surface, focused on controlling the sending rate from a single source into a queuing network. We could extend the analysis to the control of queuing networks with multiple sources as given in the general expression for the dynamics in Eq. (2). For example, in the tandem queue system that we had considered, instead of having a single source feeding customers into the first queue, we could have two sources feeding into each of the two queues. The model queuing dynamics for the non-delayed system would be of the form, $\dot{\mathbf{x}} = \bar{\alpha}(\mathbf{x}, t) + \mathbf{u}$, where, $\mathbf{x} = [x_1, x_2]^T$; $\mathbf{u} = [u_1, u_2]^T$ and $\alpha(\mathbf{x}, t)$ as obtained from Eq. (2). We can compute the control inputs using the sliding mode approach to track the queue length for the two queues with the sliding variable, $\mathbf{s} = \mathbf{x} - \mathbf{x_d}$. Similar to the control input for a single queue, the control input for tracking would be of the form $\mathbf{u}(t) = (-\bar{\alpha}(\mathbf{x}, t) + \dot{\mathbf{x_d}} - \mathbf{k} \odot \text{sgn}(\mathbf{s}))^+$.

C. Runway assignment for flights

In the results presented earlier in this paper, the runway assignment for flights was fixed as per historical schedules. However, there are instances when there is large queuing on one of the runways while the other is empty, indicating an inefficient runway assignment. Some fights might have some flexibility with regards to runway assignment, while many might not (e.g., because of constraints on the departure fix or runway length requirement). The same framework could be utilized to decide on the runway assignments, when possible. Instead of using the runway-specific demands to determine the release rates, one can use the total demand to determine release rates, while respecting the runway assignment constraints for any flight. This approach will ensure that the runways are equally loaded and utilised, increasing the overall airport throughput.

VII. Conclusions

This paper presented an approach to control the release time of customers into a queuing network, to maintain the length of the queues at desired values. The proposed approach was based on a sliding mode controller, with predictor-based feedback to account for time-delays. For the case of a single queue and server, we were able to determine error bounds for tracking the queue length in the presence of model uncertainties. Using a tandem queuing system as an example, we showed that controlling the queue length with just the bottleneck queue can result in poor tracking, and that it is important to include information of all queue lengths in the feedback. The robust controller was found to have similar tracking performance compared to an exact solution obtained using a dynamic programming approach for the delay-free case, while being more scalable and flexible than dynamic programming in handling large, dynamic networks with time-delays and general service time distributions.

Using a validated queuing network model of Charlotte airport (CLT), we applied our robust control framework to tactically decide the pushback times (release times from gates) for departure flights in order to maintain the runway queue length at a desired value. This approach resulted in a significant reduction in taxi-out times without a loss in airport throughput. Moreover, stochastic simulations of airport operations indicated that our proposed approach performs better in terms of taxi-out time reduction (by about 11%) than an alternative heuristic that is currently under consideration.

References

[1] M. H. Veatch and L. M. Wein, "Optimal control of a two-station tandem production/inventory system," *Operations Research*, vol. 42, no. 2, pp. 337–350, 1994.

[2] C. Wang, "Urban transportation networks: analytical modeling of spatial dependencies and calibration techniques for stochastic traffic simulators," Ph.D. dissertation, Massachusetts Institute of Technology, 2013.

[3] N. Pyrgiotis, K. M. Malone, and A. Odoni, "Modelling delay propagation within an airport network," *Transportation Research Part C: Emerging Technologies*, vol. 27, pp. 60–75, 2013.

[4] V. Misra, W.-B. Gong, and D. Towsley, "Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED," in *ACM SIGCOMM Computer Communication Review*, vol. 30, no. 4. ACM, 2000, pp. 151–160.

[5] S. Stidham and R. Weber, "A survey of Markov decision models for control of networks of queues," *Queueing systems*, vol. 13, no. 1-3, pp. 291–314, 1993.

[6] S. Stidham Jr, "Analysis, design, and control of queueing systems," *Operations Research*, vol. 50, no. 1, pp. 197–216, 2002.

[7] M. Polese, F. Chiariotti, E. Bonetto, F. Rigotto, A. Zanella, and M. Zorzi, "A survey on recent advances in transport layer protocols," *arXiv preprint arXiv:1810.03884*, 2018.

[8] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on networking*, no. 4, pp. 397–413, 1993.

[9] W.-c. Feng, K. G. Shin, D. D. Kandlur, and D. Saha, "The blue active queue management algorithms," *IEEE/ACM Transactions on Networking (ToN)*, vol. 10, no. 4, pp. 513–528, 2002.
[10] S. H. Low, F. Paganini, and J. C. Doyle, "Internet congestion control," *IEEE control systems magazine*, vol. 22, no. 1, pp. 28–43, 2002.
[11] C. V. Hollot, V. Misra, D. Towsley, and W. Gong, "Analysis and design of controllers for AQM routers supporting TCP flows," *IEEE Transactions on automatic control*, vol. 47, no. 6, pp. 945–959, 2002.
[12] G. Kahe, A. H. Jahangir, and B. Ebrahimi, "AQM controller design for TCP networks based on a new control strategy," *Telecommunication Systems*, vol. 57, no. 4, pp. 295–311, 2014.
[13] Q. Chen and O. W. Yang, "Robust controller design for AQM router," *IEEE Transactions on Automatic Control*, vol. 52, no. 5, pp. 938–943, 2007.
[14] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang, "Review of road traffic control strategies," *Proceedings of the IEEE*, vol. 91, no. 12, pp. 2043–2067, 2003.
[15] J. Gregoire, X. Qian, E. Frazzoli, A. De La Fortelle, and T. Wongpiromsarn, "Capacity-aware backpressure traffic signal control," *IEEE Transactions on Control of Network Systems*, vol. 2, no. 2, pp. 164–173, 2015.
[16] L. Chong and C. Osorio, "A simulation-based optimization algorithm for dynamic large-scale urban transportation problems," *Transportation Science*, vol. 52, no. 3, pp. 637–656, 2017.
[17] R. Sanchez-Iborra and M.-D. Cano, "On the similarities between urban traffic management and communication networks: Application of the random early detection algorithm for self-regulating intersections," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 4, pp. 48–61, 2017.
[18] I. Simaiakis, H. Khadilkar, H. Balakrishnan, T. G. Reynolds, and R. J. Hansman, "Demonstration of reduced airport congestion through pushback rate control," *Transportation Research Part A: Policy and Practice*, vol. 66, pp. 251–267, 2014.
[19] Federal Aviation Administration, "TFDM Overview," 2018, https://www.faa.gov/.
[20] Eurocontrol, "Airport CDM implementation manual," 2017.
[21] H. Chen and S. Solak, "Lower cost departures for airlines: Optimal policies under departure metering," *Transportation Research Part C: Emerging Technologies*, vol. 111, pp. 531–546, 2020.
[22] I. Simaiakis and H. Balakrishnan, "A queuing model of the airport departure process," *Transportation Science*, vol. 50, no. 1, pp. 94–109, 2015.
[23] S. Badrinath, M. Z. Li, and H. Balakrishnan, "Integrated surface–airspace model of airport departures," *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 5, pp. 1049–1063, 2019.
[24] Y. Wan, C. Taylor, S. Roy, C. Wanke, and Y. Zhou, "Dynamic queuing network model for flow contingency management," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1380–1392, 2013.
[25] A. Jacquillat, "A queuing model of airport congestion and policy implications at JFK and EWR," Ph.D. dissertation, Massachusetts Institute of Technology, 2012.
[26] W.-P. Wang, D. Tipper, and S. Banerjee, "A simple approximation for modeling nonstationary queues," in *INFOCOM'96. Fifteenth Annual*
[32] S. Verma, W. J. Coupe, H. Lee, I. Robeson, Y. Jung, S. Sharma, V. L. Dulchinos, and L. Stevens, "Tactical surface metering procedures and information needs for Charlotte Douglas International Airport," in *International Conference on Applied Human Factors and Ergonomics*. Springer, 2018, pp. 157–169.
Joint Conference of the IEEE Computer Societies, vol. 1. IEEE, 1996, pp. 255–262.
[27] S. Badrinath and H. Balakrishnan, "Control of a non-stationary tandem queue model of the airport surface," in *American Control Conference (ACC), 2017*. IEEE, 2017, pp. 655–661.
[28] K. Bouyoucef and K. Khorasani, "Robust feedback linearization-based congestion control using a fluid flow model," in *American Control Conference, 2006*. IEEE, 2006, pp. 6–pp.
[29] A. Pitsillides, P. Ioannou, M. Lestas, and L. Rossides, "Adaptive nonlinear congestion controller for a differentiated-services framework," *IEEE/ACM Transactions on Networking (TON)*, vol. 13, no. 1, pp. 94–107, 2005.
[30] K. Gu, V. Kharitonov, and J. Chen, *Stability of Time-delay Systems*. Birkhauser, Boston, 2003.
[31] E. Fridman, "Tutorial on Lyapunov-based methods for time-delay systems," *European Journal of Control*, vol. 20, no. 6, pp. 271–283, 2014.
[33] D. Tipper and M. K. Sundareshan, "Numerical methods for modeling computer networks under nonstationary conditions," *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 9, pp. 1682–1695, 1990.
[34] J.-J. E. Slotine, W. Li *et al.*, *Applied nonlinear control*. Prentice hall Englewood Cliffs, NJ, 1991, vol. 199, no. 1.
[35] S. Sastry, *Nonlinear systems: analysis, stability, and control*. Springer Science & Business Media, 2013, vol. 10.
[36] I. Karafyllis and M. Krstic, *Predictor feedback for delay systems: Implementations and approximations*. Springer, 2017.
[37] M. Corless and G. Leitmann, "Continuous state feedback guaranteeing uniform ultimate boundedness for uncertain dynamic systems," *IEEE Transactions on Automatic Control*, vol. 26, no. 5, pp. 1139–1144, 1981.
[38] A. Thowsen, "Uniform ultimate boundedness of the solutions of uncertain dynamic delay systems with state-dependent and memoryless feedback control," *International Journal of control*, vol. 37, no. 5, pp. 1135–1143, 1983.
[39] D. P. Bertsekas, *Dynamic programming and optimal control*. Athena scientific Belmont, MA, 1995, vol. 1, no. 2.
[40] S. Badrinath, H. Balakrishnan, E. Clemons, and T. G. Reynolds, "Evaluating the impact of uncertainty on surface operations," in *2018 Aviation Technology, Integration, and Operations Conference*. AIAA, 2018.
[41] N. Abe and K. Yamanaka, "Smith predictor control and internal model control-a tutorial," in *SICE 2003 Annual Conference (IEEE Cat. No. 03TH8734)*, vol. 2. IEEE, 2003, pp. 1383–1387.

Sandeep Badrinath is a PhD candidate in the Department of Aeronautics and Astronautics at the Massachusetts Institute of Technology. His research focuses on system modeling, design of control and optimization algorithms, with applications to airport operations and air traffic management.

Hamsa Balakrishnan is the Associate Department Head and a Professor of Aeronautics and Astronautics at the Massachusetts Institute of Technology. Her research is in the design, analysis, and implementation of control and optimization algorithms for air transportation systems.