

Time-Delay Modeling and Optimal Controller Design of Networked Control Systems*

Feng-Li Lian¹, James Moyne^{2,*}, Dawn Tilbury¹

NSF Engineering Research Center for Reconfigurable Machining Systems

1. Department of Mechanical Engineering

2. Department of Electrical Engineering and Computer Science

The University of Michigan, Ann Arbor, MI 48109-2125¹/2122²

{fengli,moyne,tilbury}@umich.edu

Submitted for publication in the *International Journal of Control*

August 19, 2001

Abstract

In this paper we discuss the modeling and control of networked control systems (NCS) where sensors, actuators, and controllers are distributed and interconnected by a common communication network. Multiple distributed communication delays as well as multiple inputs and multiple outputs (MIMO) are considered in the modeling algorithm. In addition, the asynchronous sampling mechanisms of distributed sensors are characterized to obtain the actual time delays between sensors and the controller. Due to the characteristics of a network architecture, piecewise constant plant inputs are assumed and discrete-time models of plant and controller dynamics are adopted to analyze the stability and performance of a closed-loop NCS. The analysis result is used to verify the stability and performance of an NCS without considering the impact of multiple time delays in the controller design. In addition, the proposed NCS model is used as a foundation for optimal controller design. The proposed control algorithm utilizes the information of delayed signals and improves the control performance of a control system encountering distributed communication delays. Several simulation studies are provided to verify the control performance of the proposed controller design.

1 Introduction

A major trend in modern industrial and commercial systems is to integrate computing, communication, and control into different levels of machine/factory operations and information processes. The traditional communication architecture for control systems, which has been successfully implemented in industry for decades, is point-to-point; that is, a wire connects the central control computer with each sensor or actuator point. However, expanding physical setups and functionality are pushing the limits of the point-to-point architecture. Hence, a traditional centralized point-to-point control system is no longer suitable to meet

*This research was supported in part by the NSF Engineering Research Center for Reconfigurable Machining Systems under the grant EEC95-92125. Corresponding Author: James Moyne, Electrical Engineering and Computer Science Department, University of Michigan, 1124 EECS Building, 1301 Beal Avenue, Ann Arbor, MI48109-2122, USA

new requirements, such as modularity, decentralization of control, integrated diagnostics, quick and easy maintenance, and low cost. The introduction of common-bus network architectures can improve the efficiency, flexibility and reliability of these integrated applications through reduced wiring and distributed intelligence, and reduce installation, reconfiguration, and maintenance time and costs.

These types of distributed control systems are called networked control systems (NCS): sensors, actuators, and controllers are interconnected by one communication network. The change of communication architecture from point-to-point to common-bus, however, introduces different forms of time delay uncertainty between sensors, actuators, and controllers. These time delays come from the time sharing of the communication medium as well as the computation time required for physical signal coding and communication processing. The characteristics of time delays can be constant, bounded, or even random, depending on the network protocols adopted and the chosen hardware. It is well-known in control systems that time delays can degrade a system's performance and even cause system instability. However, though the analysis and modeling of time-delay systems has been a progressive research area, existing methodologies cannot be directly applied due to the discrete and distributed nature of the many different time delays in NCSs.

Most NCS research has focused on two areas: communication protocols and controller design. A proper message transmission protocol is necessary to guarantee the network quality of service, whereas advanced controller design is desirable to guarantee the control quality of performance. In this paper we consider the modeling of an NCS with multiple communication delays and formulate and solve the optimal control problem based on the proposed discrete-time time-delay model. In Section 2, we survey existing results on time-delay systems and networked control systems. In Section 3, we discuss the problem formulation and assumptions used in the discrete-time modeling algorithm. In Section 4, we address the delay impact on plant and controller dynamics. In Section 5, we provide an example to illustrate the stability analysis of the proposed modeling approach. Based on the proposed modeling framework, we formulate an optimal controller in Section 6. We present simulation result of the proposed controller design in Section 7. In Section 8, we address the control performance analysis of the proposed optimal controller design for a time-delay NCS. In Section 9, we summarize the results of the proposed modeling and controller design of networked control systems.

2 Previous research

In this section we discuss related research on analysis of time-delay systems and modeling of control of networked control systems.

2.1 Analysis of time-delay systems

Modeling and control of NCSs is based on the time-delay systems analysis framework which has been studied for several decades. In general, delays occur in the transmission of signals or materials between different subsystems. Large-scale systems such as communication systems, manufacturing systems, transportation systems, power systems, and teleoperation systems are typical examples of time-delay systems [23]. There have been two approaches used to analyze the stability of time-delay systems: classical (frequency domain) and functional (time domain) approaches [7, 24, 28]. The classical approach utilizes analytical or graphical methods to find the roots of the characteristic equation of a dynamic system. Since delays appear in the characteristic equation as exponential functions, analytical techniques are developed to find solutions of an exponential polynomial. Alternatively, the Padé approximation to the time delay can be used to obtain a pure polynomial. In addition, standard graphical methods such as the root locus, Bode plot, and Nyquist diagram can be further modified to analyze delayed polynomials or transfer functions. For a discrete-time system with time delays, the system stability can also be easily verified by the root locus analysis in the z -domain [4]. However, the classical approach is only effective for single-delay cases. For multi-delay cases, the functional approach must be used.

The functional approach uses delay-differential equations to characterize systems with delays in the state, input, or both. Multiple delays [9, 12], and time-varying delays [8, 10], can be modeled in a similar setting. Based on standard algorithms from functional analysis, solutions of delay-differential equations can be constructed, and their stability can be analyzed using Lyapunov's second method (i.e., Lyapunov-Krasovskii and/or Lyapunov-Razumikhin stability theorems) [2].

Since time delays in most applications are considered non-deterministic parameters, controller design for time-delay systems typically uses either a robust or stochastic control approach. Robust \mathcal{H}_∞ controllers can be designed based on the known system structure or delay uncertainty. Uncertainties in both the system matrix and the input matrix can be included in the delay-differential dynamic equation. Moreover, using a linear matrix inequality (LMI) approach, system stability and performance can be analyzed and guaranteed [13, 17, 22].

These modeling and analysis approaches developed for time-delay systems can only be used as a foundation for analyzing the time-delay effects in NCSs. Since all devices in an NCS are distributed on one common-bus network, sample and hold and time skew among samples are typical, and result in an inherent asynchronization in the system model. Also, due to the distribution of actuators and the network communication mechanism, the system inputs are piecewise constant with delays, rather than continuous. These

properties have not been discussed in previous time-delay analyses; thus further research is needed to model and analyze NCSs.

2.2 Modeling and control of networked control systems

Research in NCSs is different from that in traditional time-delay systems. Because of the variability of network-induced time delays, the NCSs may be time-varying systems, making analysis and design more challenging. Wittenmark et al. [32] discussed several timing issues such as communication and computation delays, processor jitter, and transient errors existing in NCSs. These timing issues must be addressed when deriving a discrete-time state-space model. In their subsequent work, a discrete-time model with a single sensor-controller delay and a single controller-actuator delay was studied, and a stochastic controller design was designed [26, 27]. Nilsson studied the case with multiple sensor-controller and controller-actuator delays [25]; only the case where the total maximum network delay is less than one controller sampling period was considered.

Recently research on the analysis and modeling of NCSs has been conducted using continuous-time and discrete-time models. It is more natural to analyze an NCS from the discrete-time point of view since in a typical NCS operation, physical signals (from sensors or to actuators) are sampled and then transmitted on the network medium after a short delay. For discrete-time models, most researchers assume that the network is synchronized and the sampling rates of sensors, controllers, and actuators are the same. Halevi and Ray [11] considered the case of a single time delay for sensor-controller and controller-actuator and a single time skew between sensor and controller sampling instants. They used the augmented state to include the past delayed signals and derived a closed-loop model for NCSs.

Krtolica et al. [15] derived a discrete-time time-varying state-space representation of NCSs with random network delays by using the augmented states to include past plant and controller states. The total number of states used depends on the possible range of sensor-controller and controller-actuator delays. Since the maximum possible number of network delays is bounded, and the closed-loop system is a discrete-time model, the system matrix can be viewed as a finite automaton with finite states. The stability analysis of such a system can be further described by a Markov chain with finite state transitions.

Branicky et al. [1] considered a simplified NCS model where the sensor-controller and controller-actuator delays are lumped together. Their study only considered the case where the lumped time delay is less than one sampling period; the stability regions for the NCS model were investigated based on the lumped delay. In practical applications, however, sensor-controller and controller-actuator delays are different and time-varying at different networked devices due to the network transmission mechanism. Hence, a proper time

delay profile in NCSs should be characterized based on the network transmission bandwidth and control system bandwidth.

Continuous-time NCS models were considered by several researchers. Göktas et al. [5, 6] used a modified Padé approximation and considered the network delay as an uncertainty. They designed a robust controller to compensate for the uncertain delay in an ATM network. Kim et al. [14] used a Lyapunov approach to obtain the maximum allowable delay bound for the stability of a network delayed system. A scheduling algorithm for determining the sampling rate and allocating bandwidth was also provided. Walsh et al. [30, 31] also adopted the Lyapunov approach on a continuous-time model to obtain the maximum allowable transfer interval and to analyze the stability of the closed-loop system. They further analyzed the impact of different scheduling algorithms on the maximum allowable transfer interval. However, only a conservative delay bound was obtained. The impact of delay variance on control performance is discussed in these works, but is not formally characterized.

In practical applications, however, sensor-controller and controller-actuator delays are different and time-varying at different networked devices due to the network transmission mechanism. An NCS should be modeled based on the characterization of network-induced delays and the consideration of network and control parameters. Furthermore, controllers should be designed based on the NCS model, taking into account the delay information. Therefore, in this paper, a discrete-time model of NCSs with multiple inputs and multiple outputs (MIMO) and multiple distributed communication delays is derived. In addition, the asynchronous sampling mechanisms of distributed sensors are characterized to obtain the actual time delays between sensors and the controller. Based on the proposed NCS model, the stability and performance of a closed-loop system with a standard controller are analyzed, and a linear quadratic regulator (LQR) optimal control is formulated to compensate for the multiple time delays.

3 Problem formulation and assumptions

Consider the block diagram of a networked control system with a single controller, but multiple sensors and multiple actuators as shown in Fig. 1. There are N states (x), M inputs (u), and R outputs (y) in the **plant dynamics model**, and Q states (z), R inputs (w), and M outputs (v), in the **controller dynamics model**, i.e., M actuators, R sensors and one controller, where N , M , R , and Q are positive constant integers. We use $s_1, s_2, \dots, s_r, \dots, s_R$ and $a_1, a_2, \dots, a_m, \dots, a_M$ to represent the sensor-controller and controller-actuator delays, respectively. The variables w_r and u_m are the delayed y_r and v_m signals, respectively. The relationships between these variables will be addressed later. In the following discussion, we will present the system models

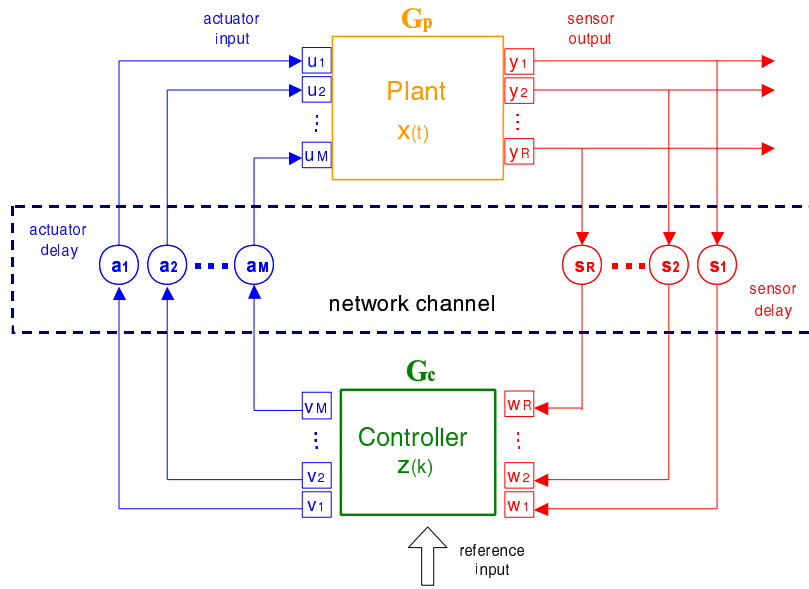


Figure 1: The block diagram of a networked control system.

Sensors, actuators, and controllers in an NCS are distributed and interconnected by communication networks. NCSs are flexible, reconfigurable, and efficient. The information of these devices can be easily shared by other subsystems. However, time-delays between sensor-controller and controller-actuator are unavoidable because of the sharing of the communication medium.

in continuous time and discrete time. In this paper time is denoted by t for the continuous-time domain and k for the discrete-time domain.

In Fig. 1, the continuous-time, state-space model of the linear time-invariant plant dynamics G_p can be described by the following standard form:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}_p \mathbf{x}(t) + \mathbf{B}_p \mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{C}_p \mathbf{x}(t),\end{aligned}\tag{1}$$

where $\mathbf{x}(t) \in \mathbb{R}^N$, $\mathbf{u}(t) \in \mathbb{R}^M$, $\mathbf{y}(t) \in \mathbb{R}^R$ and the constant matrices \mathbf{A}_p , \mathbf{B}_p , and \mathbf{C}_p are of compatible dimensions. Since the controller is implemented at one digital computer, the controller is designed in discrete time with a sampling time T and the state-space model of the controller dynamics G_c can be expressed as follows:

$$\begin{aligned}\mathbf{z}_{k+1} &= \mathbf{F} \mathbf{z}_k + \mathbf{G} \mathbf{w}_k, \\ \mathbf{v}_k &= \mathbf{H} \mathbf{z}_k + \mathbf{J} \mathbf{w}_k,\end{aligned}\tag{2}$$

where $\mathbf{z}_k \triangleq \mathbf{z}(k) = \mathbf{z}(kT) \in \mathbb{R}^Q$, $\mathbf{w}_k \triangleq \mathbf{w}(k) = \mathbf{w}(kT) \in \mathbb{R}^R$, $\mathbf{v}_k \triangleq \mathbf{v}(k) = \mathbf{v}(kT) \in \mathbb{R}^M$ and the matrices \mathbf{F} , \mathbf{G} , \mathbf{H} , and \mathbf{J} are of compatible dimensions. Note that \mathbf{w}_k is a delayed version of the sensor output $\mathbf{y}(t)$ at some sampling instant, and, similarly, $\mathbf{u}(t)$ is a delayed version of the controller output \mathbf{v}_k .

In practical applications of networked control systems, devices are distributed and have their own processing units and timing functions. Hence, synchronization of all devices is extremely difficult. In this paper, we assume that the network is not synchronized; each device may have a different time skew when related to the controller sampling instants. We also assume that the sensor and controller sampling times are the same, and that actuators respond to actuation commands immediately after receiving the information from the controller. The detailed assumptions and notations used in this paper are described as follows and are illustrated in Fig. 2.

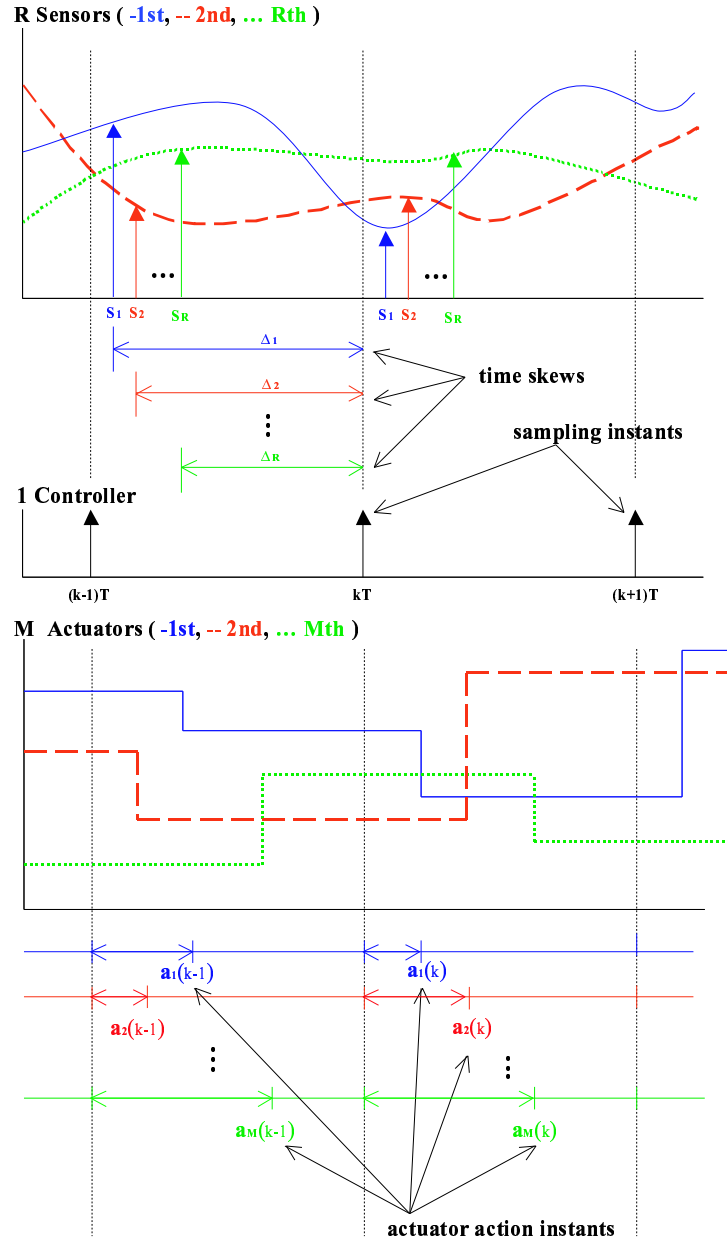


Figure 2: The timing diagram for sensors, controller, and actuators in an NCS.

1. The **buffer length** at the controller for each sensor and each actuator is equal to one. That is, the controller only uses the newest sensor messages and never sends a stale actuator command.
2. As shown in Fig. 2, the **periods** of all R sensors and one controller are identical and equal to T , but there may be R different **time skews**, denoted as $\Delta_r, r = 1, \dots, R$, among these sensor sampling instants. The definition of Δ_r is the time difference between the sampling instant of the r th sensor and the sampling instant of the controller. We assume that Δ_r 's are constant.
3. There are two types of time delays: **processing delay** and **communication delay**. Processing delays occur at the controller, sensors, and actuators, and are denoted as $p^c(k), p_r^s(k)$, and $p_m^a(k)$, respectively. Communication delays between the sensors and the controller and between the controller and the actuators are denoted as $c_r^s(k), c_m^a(k)$, respectively. In this paper, we add the processing delays to the communication delays. The combined time delays are defined as follows. The combined sensor processing-communication delay is $c_r^s(k) + p_r^s(k)$ and the combined actuator-controller processing-communication delay is $a_m(k) = p^c(k) + c_m^a(k) + p_m^a(k)$. Hence, in the following discussion, we only consider the combined time delays. We also assume that these delays are bounded by one sampling period, that is, the cases of vacant sampling and message rejection are not considered in this paper.
4. As shown in Fig. 2, the **sensor-controller delay** $s_r(k)$ depends on both the time skew Δ_r and the sampling period T . If $c_r^s(k) + p_r^s(k) \leq \Delta_r$, then $s_r(k) = \Delta_r$. Otherwise, $s_r(k) = T + \Delta_r$. Here, we also assume that the time delay is bounded by the sampling period; that is, $\max\{s_r(k), r = 1, \dots, R\} \leq 2T$. Note that this assumption is true for deterministic network protocols such as token passing or priority-based under normal traffic load. However, for those networks with a stochastic medium access control mechanism, this assumption might not be true.
5. Since the **controller-actuator delay** is $a_m(k)$, the k th controller sampling interval $[(k-1)T, kT)$, can be formulated as $[0, T) = [0, a_m(k)) \cup [a_m(k), T)$. Note that if the actuation delay is longer than one sampling period, then $a_m(k) = T$. This longer delay may be due to the blocking of message transmission or using too small of a sampling period. The first situation can be improved by further considering network parameters in system design. For the second situation, a larger T should be chosen. However, the control performance may degrade due to long sampling period. On other hand, a smaller sampling period increases the control performance as well as the system complexity due to the high network traffic load.
6. For any variable θ , we will use $\bar{\theta} = \theta/T$ to denote its value in terms of sampling period. In addition, we split $\bar{\theta}$ into its integer and fractional parts as $\bar{\theta} = \hat{\theta} + \tilde{\theta}$, where $\hat{\theta} \in \mathbb{Z}^+$ and $0 \leq \tilde{\theta} < 1$.

4 Delay impact on plant and controller dynamics

One of the main characteristics of NCSs is the different communication delays between the plant and controller, as shown in Fig. 1. That is, in general, $\mathbf{u}(t)|_{t=kT} \neq \mathbf{v}_k$ and $\mathbf{w}_k \neq \mathbf{y}(t)|_{t=kT}$, because of existing time delays among these signals. The actual values of these communication delays depend on the network protocol adopted as well as the network traffic load. In this section, we will derive the relation between each pair of variables based on the assumptions described in Section 3.

4.1 Plant input and controller output

We first study the relation between the controller outputs \mathbf{v}_k and the plant inputs $\mathbf{u}(t)$. Since the actuators receive controller commands discontinuously, we assume that the actuator inputs are piecewise constant as shown in the actuator timing diagram in Fig. 2(c). Hence, for the m th actuator, the input signal $u_m(t)$ can be described as follows, for $kT \leq t < (k+1)T$:

$$u_m(t) = v_m(k-1)\mathbf{1}_{(0, a_m(k))}(t) + v_m(k)\mathbf{1}_{(a_m(k), T)}(t), \quad (3)$$

where

$$\mathbf{1}_{(a(k), b(k))}(t) = \begin{cases} 1 & \text{if } a(k) + kT \leq t < b(k) + kT \\ 0 & \text{otherwise,} \end{cases} \quad (4)$$

That is, $u_m(t)$ is the combination of piecewise constant functions. For example, consider the first actuator at the k th sampling instant shown in Fig. 2. The following relation holds: $u_1(t) = v_1(k-1)\mathbf{1}_{(0, a_1(k))} + v_1(k)\mathbf{1}_{(a_1(k), T)}$, for $kT \leq t < (k+1)T$. Then, $\mathbf{u}(t) = [u_1(t), \dots, u_M(t)]^T$ and $\mathbf{v}(k) = [v_1(k), \dots, v_M(k)]^T$.

4.2 Plant model in discrete-time domain

In order to analyze the closed-loop system in discrete-time, we will use the following state-space solution of a first-order matrix differential equation to discretize the continuous-time plant dynamics model [4]:

$$\mathbf{x}(t) = \exp(\mathbf{A}_p(t - t_0))\mathbf{x}(t_0) + \int_{t_0}^t \exp(\mathbf{A}_p(t - q'))\mathbf{B}_p\mathbf{u}(q')dq'. \quad (5)$$

We first discretize the plant model at the controller sampling instants by applying Eq. (5) with $t_0 = kT$ and $t = (k+1)T$. For simplicity, we use the notation $\mathbf{x}_k \triangleq \mathbf{x}(k) = \mathbf{x}(kT)$, $\mathbf{A} \triangleq \exp(\mathbf{A}_pT)$, $q = q' - kT$, and $\mathbf{\Gamma}(T, q) \triangleq \exp(\mathbf{A}_p(T - q))\mathbf{B}_p \in \mathbb{R}^{N \times M}$. Then,

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \int_0^T \exp(\mathbf{A}_p(T - q))\mathbf{B}_p\mathbf{u}(kT + q)dq, \\ &= \mathbf{A}\mathbf{x}_k + \int_0^T \mathbf{\Gamma}(T, q)\mathbf{u}(kT + q)dq, \end{aligned}$$

$$\begin{aligned}
&= \mathbf{A}\mathbf{x}_k + \int_0^T [\boldsymbol{\Gamma}_1(T, q), \dots, \boldsymbol{\Gamma}_M(T, q)] \begin{bmatrix} u_1(kT + q) \\ \vdots \\ u_M(kT + q) \end{bmatrix} dq, \\
&= \mathbf{A}\mathbf{x}_k + \sum_{m=1}^M \int_0^T \boldsymbol{\Gamma}_m(T, q) u_m(kT + q) dq,
\end{aligned} \tag{6}$$

where $\boldsymbol{\Gamma}_m \in \mathbb{R}^{N \times 1}$ and $u_m \in \mathbb{R}$, $m = 1, \dots, M$. The first and second equalities are obtained by definition. The third and fourth equalities are used to explicitly describe $\boldsymbol{\Gamma}$ and \mathbf{u} in terms of their elements, $\boldsymbol{\Gamma}_m$'s and u_m 's, respectively, $m = 1, \dots, M$. Now, because the actuator commands arrive at different times within the sample interval, u_m is not constant over $[kT, (k+1)T]$. From Eq. (3), we can compute $\int_0^T \boldsymbol{\Gamma}_m(T, q) u_m(kT + q) dq$ as follows:

$$\begin{aligned}
\int_0^T \boldsymbol{\Gamma}_m(T, q) u_m(kT + q) dq &= \int_0^{a_m(k)} \boldsymbol{\Gamma}_m(T, q) v_m(k-1) dq + \int_{a_m(k)}^T \boldsymbol{\Gamma}_m(T, q) v_m(k) dq, \\
&\triangleq \mathbf{B}_m^1(k) v_m(k-1) + \mathbf{B}_m^0(k) v_m(k),
\end{aligned} \tag{7}$$

where we also use the timing relations in Assumption 5 and the following definitions: $\mathbf{B}_m^1(k) \triangleq \int_0^{a_m(k)} \boldsymbol{\Gamma}_m(T, q) dq$ and $\mathbf{B}_m^0(k) \triangleq \int_{a_m(k)}^T \boldsymbol{\Gamma}_m(T, q) dq$. Therefore, by applying Eq. (7) to Eq. (6), we have:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \sum_{m=1}^M \sum_{j=0}^1 \mathbf{B}_m^j(k) v_m(k-j). \tag{8}$$

By further exchanging the order of summations, we have the following derivation:

$$\begin{aligned}
\mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \sum_{j=0}^1 \sum_{m=1}^M \mathbf{B}_m^j(k) v_m(k-j) \\
&= \mathbf{A}\mathbf{x}_k + \sum_{j=0}^1 \left[\mathbf{B}_1^j(k) \mathbf{B}_2^j(k) \dots \mathbf{B}_M^j(k) \right] \begin{bmatrix} v_1(k-j) \\ v_2(k-j) \\ \vdots \\ v_M(k-j) \end{bmatrix} \\
&= \mathbf{A}\mathbf{x}_k + \sum_{j=0}^1 \mathbf{B}^j(k) \mathbf{v}(k-j) \\
&= \mathbf{A}\mathbf{x}_k + \sum_{j=0}^1 \mathbf{B}_k^j \mathbf{v}_{k-j}.
\end{aligned} \tag{9}$$

In Eq. (9), \mathbf{A} is time-invariant because it is independent of the delay variables $a_m(k)$, but the \mathbf{B}_k^j 's will depend on the values of $a_m(k)$'s. Therefore, if the $a_m(k)$ depend on k , the NCS model will be time-varying.

4.3 $\mathbf{x}((k+1)T - \delta T)$: the state value between sampling instants

Because the sampling of sensors happens between the sampling instants of the controller, we need to derive the formula for the state values between sampling instants, i.e., $\mathbf{x}((k+1)T - \delta T)$, where $0 \leq \delta < 1$. By using Eq. (5) again with $t_0 = kT$, and $t = (k+1 - \delta)T$,

$$\mathbf{x}((k+1 - \delta)T) = \exp(\mathbf{A}_p(T - \delta T)) \mathbf{x}(kT)$$

$$\begin{aligned}
& + \int_{kT}^{(k+1-\delta)T} \exp(\mathbf{A}_p((k+1-\delta)T - q')) \mathbf{B}_p \mathbf{u}(q') dq' \\
& = \mathbf{A}_\delta \mathbf{x}_k + \sum_{j=0}^1 \mathbf{B}_\delta^j(k) \mathbf{v}(k-j) \\
& = \mathbf{A}_\delta \mathbf{x}_k + \sum_{j=0}^1 \mathbf{B}_{\delta k}^j \mathbf{v}_{k-j}, \tag{10}
\end{aligned}$$

where $\mathbf{A}_\delta = \exp(\mathbf{A}_p(T - \delta T))$, $\mathbf{B}_\delta^j(k) \in \mathbb{R}^{N \times M}$, and $\mathbf{v}_{k-j} \in \mathbb{R}^{M \times 1}$. Note that \mathbf{A}_δ is not constant but depends on δ , in contrast to \mathbf{A} in Eq. (6), and $\mathbf{B}_\delta^j(k)$ can be formulated similarly to \mathbf{B}_k^j . If $0 \leq \delta T < T - a_m(k)$, then the m th elements of $\mathbf{B}_\delta^0(k)$ and $\mathbf{B}_\delta^1(k)$ are $\mathbf{B}_{\delta m}^0(k) \triangleq \int_{a_m(k)}^{T-\delta} \mathbf{\Gamma}_m(T, q) dq$, and $\mathbf{B}_{\delta m}^1(k) \triangleq \int_0^{a_m(k)} \mathbf{\Gamma}_m(T, q) dq$, respectively. If $T - a_m(k) \leq \delta T < T$, then the m th elements of $\mathbf{B}_\delta^0(k)$ and $\mathbf{B}_\delta^1(k)$ are $\mathbf{B}_{\delta m}^0(k) \triangleq \mathbf{0}$, and $\mathbf{B}_{\delta m}^1(k) \triangleq \int_0^{\delta T} \mathbf{\Gamma}_m(T, q) dq$, respectively.

4.4 Plant output and controller input

Similarly, for plant outputs $\mathbf{y}(t)$ and controller inputs \mathbf{w}_k , we have the following relation due to the communication delays between the sensors and the controller. By Assumption 6 and for simplicity, we have the following equation for the r th sensor-controller delay: $s_r(k) = [\bar{s}_r(k)]T = [\hat{s}_r(k) + \tilde{s}_r(k)]T$, where \hat{s}_r is an integer and $0 \leq \tilde{s}_r < 1$. Therefore, for any value of $s_r(k)$ and by Eq. (10),

$$\begin{aligned}
\mathbf{x}(kT - s_r(k)) & = \mathbf{x}(kT - \bar{s}_r(k)T) \\
& = \mathbf{x}((k - \hat{s}_r(k))T - \tilde{s}_r(k)T) \\
& = \mathbf{A}_{\tilde{s}_r(k)} \mathbf{x}_{k-1-\hat{s}_r(k)} \\
& + \sum_{j=0}^1 \mathbf{B}_{\tilde{s}_r(k)}^j(k-1-\hat{s}_r(k)) \mathbf{v}(k-1-\hat{s}_r(k)-j), \tag{11}
\end{aligned}$$

where we let $\delta = \tilde{s}_r(k)$ in Eq. (10). At the k th sampling instant, the values of the plant output received by the controller are:

$$\begin{aligned}
\mathbf{w}_k & = \mathbf{w}(kT) \\
& = \begin{bmatrix} w_1(kT) \\ \vdots \\ w_R(kT) \end{bmatrix} = \begin{bmatrix} y_1(kT - s_1(k)) \\ \vdots \\ y_R(kT - s_R(k)) \end{bmatrix} = \begin{bmatrix} \mathbf{C}_1 \mathbf{x}(kT - s_1(k)) \\ \vdots \\ \mathbf{C}_R \mathbf{x}(kT - s_R(k)) \end{bmatrix} \\
& = \begin{bmatrix} \mathbf{C}_1 \left[\mathbf{A}_{\tilde{s}_1(k)} \mathbf{x}_{k-1-\hat{s}_1(k)} \right. \\ \left. + \sum_{j=0}^1 \mathbf{B}_{\tilde{s}_1(k)}^j(k-1-\hat{s}_1(k)) \mathbf{v}(k-1-\hat{s}_1(k)-j) \right] \\ \vdots \\ \mathbf{C}_R \left[\mathbf{A}_{\tilde{s}_R(k)} \mathbf{x}_{k-1-\hat{s}_R(k)} \right. \\ \left. + \sum_{j=0}^1 \mathbf{B}_{\tilde{s}_R(k)}^j(k-1-\hat{s}_R(k)) \mathbf{v}(k-1-\hat{s}_R(k)-j) \right] \end{bmatrix} \\
& = \begin{bmatrix} \mathbf{C}_1 \mathbf{A}_{\tilde{s}_1(k)} \mathbf{x}_{k-1-\hat{s}_1(k)} \\ \vdots \\ \mathbf{C}_R \mathbf{A}_{\tilde{s}_R(k)} \mathbf{x}_{k-1-\hat{s}_R(k)} \end{bmatrix}
\end{aligned}$$

$$\begin{aligned}
& + \begin{bmatrix} \mathbf{C}_1 \sum_{j=0}^1 \mathbf{B}_{\hat{s}_1(k)}^j (k-1-\hat{s}_1(k)) \mathbf{v}(k-1-\hat{s}_1(k)-j) \\ \vdots \\ \mathbf{C}_R \sum_{j=0}^1 \mathbf{B}_{\hat{s}_R(k)}^j (k-1-\hat{s}_R(k)) \mathbf{v}(k-1-\hat{s}_R(k)-j) \end{bmatrix} \\
& = \sum_{i=1}^2 \Theta_k^i \mathbf{x}_{k-i} + \sum_{i=1}^3 \Phi_k^i \mathbf{v}_{k-i}, \tag{12}
\end{aligned}$$

$$\begin{aligned}
\text{where} \quad \Theta_k^i &= \begin{bmatrix} \Theta_k^{i1} \\ \vdots \\ \Theta_k^{iR} \end{bmatrix}, \quad \Theta_k^{ir} = \begin{cases} \mathbf{C}_r \mathbf{A}_{\hat{s}_r(k)}, & \text{if } \hat{s}_r(k) = i-1, \\ 0, & \text{otherwise,} \end{cases} \\
\text{and} \quad \Phi_k^i &= \begin{bmatrix} \Phi_k^{i1} \\ \vdots \\ \Phi_k^{iR} \end{bmatrix}, \\
\Phi_k^{ir} &= \begin{cases} \mathbf{C}_r \mathbf{B}_{\hat{s}_r(k)}^0 (k-1), & \text{if } i=1, \\ \mathbf{C}_r \left[\mathbf{B}_{\hat{s}_r(k)}^1 (k-1) + \mathbf{B}_{\hat{s}_r(k)}^0 (k-2) \right], & \text{if } i=2, \\ \mathbf{C}_r \mathbf{B}_{\hat{s}_r(k)}^1 (k-2), & \text{if } i=3. \end{cases}
\end{aligned}$$

4.5 Closed-loop model of NCSs

In order to analyze the system property and provide guidelines for controller design, we will derive the closed-loop model that combines the discrete-time plant model, Eq. (9), controller model, Eq. (2), and Eq. (12).

$$\begin{aligned}
\mathbf{x}_{k+1} &= \mathbf{A} \mathbf{x}_k + \mathbf{B}_k^0 \mathbf{v}_k + \mathbf{B}_k^1 \mathbf{v}_{k-1} \\
\mathbf{x}_{k+1} &= \mathbf{A} \mathbf{x}_k + \mathbf{B}_k^0 [\mathbf{H} \mathbf{z}_k + \mathbf{J} \mathbf{w}_k] + \mathbf{B}_k^1 \mathbf{v}_{k-1} \\
\mathbf{x}_{k+1} &= \mathbf{A} \mathbf{x}_k + \mathbf{B}_k^0 \mathbf{H} \mathbf{z}_k \\
&+ \mathbf{B}_k^0 \mathbf{J} \left[\sum_{i=1}^2 \Theta_k^i \mathbf{x}_{k-i} + \sum_{i=1}^3 \Phi_k^i \mathbf{v}_{k-i} \right] + \mathbf{B}_k^1 \mathbf{v}_{k-1}. \tag{13}
\end{aligned}$$

Also, the controller dynamics, Eq. (2), can be further expressed as follows:

$$\mathbf{z}_{k+1} = \mathbf{F} \mathbf{z}_k + \mathbf{G} \left[\sum_{i=1}^2 \Theta_k^i \mathbf{x}_{k-i} + \sum_{i=1}^3 \Phi_k^i \mathbf{v}_{k-i} \right] \tag{14}$$

$$\mathbf{v}_k = \mathbf{H} \mathbf{z}_k + \mathbf{J} \left[\sum_{i=1}^2 \Theta_k^i \mathbf{x}_{k-i} + \sum_{i=1}^3 \Phi_k^i \mathbf{v}_{k-i} \right]. \tag{15}$$

By further combining Eqs. (13)–(15), and defining:

$$\mathbf{X}_k = [\mathbf{x}_k^T \ \mathbf{x}_{k-1}^T \ \mathbf{x}_{k-2}^T \mid \mathbf{z}_k^T \mid \mathbf{v}_{k-1}^T \ \mathbf{v}_{k-2}^T \ \mathbf{v}_{k-3}^T]^T \in \mathbb{R}^{3N+Q+3R},$$

we obtain the closed-loop dynamics as follows:

$$\mathbf{X}_{k+1} = \Xi_k \mathbf{X}_k, \quad (16)$$

where

$$\Xi_k = \left[\begin{array}{c|c|c} \Xi_k(1,1) & \Xi_k(1,2) & \Xi_k(1,3) \\ \hline \Xi_k(2,1) & \Xi_k(2,2) & \Xi_k(2,3) \\ \hline \Xi_k(3,1) & \Xi_k(3,2) & \Xi_k(3,3) \end{array} \right] \in \mathbb{R}^{(3N+Q+3R) \times (3N+Q+3R)}. \quad (17)$$

The variables $\Xi(i, j)$ are defined as follows:

$$\begin{aligned} \Xi_k(1,1) &= \left[\begin{array}{cc} \mathbf{A} & \mathbf{B}_k^0 \mathbf{J} \Theta_k^1 \quad \mathbf{B}_k^0 \mathbf{J} \Theta_k^2 \\ \hline \mathbf{I}_{2N \times 2N} & \mathbf{0}_{2N \times N} \end{array} \right] \in \mathbb{R}^{3N \times 3N}, \\ \Xi_k(1,2) &= \left[\begin{array}{c} \mathbf{B}_k^0 \mathbf{H} \\ \hline \mathbf{0}_{2N \times Q} \end{array} \right] \in \mathbb{R}^{3N \times Q}, \\ \Xi_k(1,3) &= \left[\begin{array}{ccc} \mathbf{B}_k^0 \mathbf{J} \Phi_k^1 + \mathbf{B}_k^1 & \mathbf{B}_k^0 \mathbf{J} \Phi_k^2 & \mathbf{B}_k^0 \mathbf{J} \Phi_k^3 \\ \hline \mathbf{0}_{2N \times 3R} \end{array} \right] \in \mathbb{R}^{3N \times 3R}, \\ \Xi_k(2,1) &= \left[\begin{array}{ccc} \mathbf{0}_{Q \times N} & \mathbf{G} \Theta_k^1 & \mathbf{G} \Theta_k^2 \end{array} \right] \in \mathbb{R}^{Q \times 3N}, \\ \Xi_k(2,2) &= \mathbf{F} \in \mathbb{R}^{Q \times Q}, \\ \Xi_k(2,3) &= \left[\begin{array}{ccc} \mathbf{G} \Phi_k^1 & \mathbf{G} \Phi_k^2 & \mathbf{G} \Phi_k^3 \end{array} \right] \in \mathbb{R}^{Q \times 3R}, \\ \Xi_k(3,1) &= \left[\begin{array}{cc} \mathbf{0}_{R \times N} & \mathbf{J} \Theta_k^1 \quad \mathbf{J} \Theta_k^2 \\ \hline \mathbf{0}_{2R \times 3N} \end{array} \right] \in \mathbb{R}^{3R \times 3N}, \\ \Xi_k(3,2) &= \left[\begin{array}{c} \mathbf{H} \\ \hline \mathbf{0}_{2R \times Q} \end{array} \right] \in \mathbb{R}^{3R \times Q}, \\ \Xi_k(3,3) &= \left[\begin{array}{ccc} \mathbf{J} \Phi_k^1 & \mathbf{J} \Phi_k^2 & \mathbf{J} \Phi_k^3 \\ \hline \mathbf{I}_{2R \times 2R} & \mathbf{0}_{2R \times R} \end{array} \right] \in \mathbb{R}^{3R \times 3R}. \end{aligned}$$

The closed-loop system could be time-varying since Ξ_k will depend on the network delay characteristics. If the network delays are constant, then the closed-loop system will be time-invariant.

At the first step of analysis procedures, we assume the controller has been designed. Then the bound for different network delays can be found based on the stability criterion that all the eigenvalues of Ξ_k are less than 1. However, in the MIMO case, there are $M + R$ different time delays and it is very difficult to determine the upper bound of delay values. The models, Eqs. (13)–(15), provide the system structure under network delays. For the controller design, although the exact value of the system matrices are unknown, an estimation algorithm can be developed to identify on-line the constant system parameters when the system has constant network delays. For random time delays due to a network protocol or random processing times, a stochastic controller may be used to guarantee stability and performance [15, 27, 29].

5 Illustrative example of NCS stability analysis

In this section, we consider a two-axis example of a three-axis milling machine tool. Each axis moves on a linear slide and is driven through a ball screw by a DC motor with a tachometer which provides an angular

velocity measurement. The DC motor is driven by a PWM drive with control input between 0 (negative) and 255 (positive). Each axis also has a linear encoder that provides linear position measurement. Therefore, both position and velocity feedback are available. The two axes operate independently. The mathematical model of each axis between the PWM input (U) and the position output (P) is described by a second-order linear system:

$$G(s) = \frac{P(s)}{U(s)} = \frac{K}{s(\tau s + 1)} \quad (18)$$

The time constants τ (*sec*) for each axis are 0.055 (X) and 0.056 (Y) and the overall gains K ($((mm/sec)/PWM)$) are 28.346 (X) and 28.956 (Y), respectively.

Then, we define $x_1 = P_x, x_2 = V_x, x_3 = P_y, x_4 = V_y, u_1 = u_x$, and $u_2 = u_y$, where P_i and V_i are the position and velocity variables of the i -axis. The state space form of the two-axis system can be expressed as follows:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -18.18 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -17.86 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 515.38 & 0 \\ 0 & 0 \\ 0 & 517.07 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

or $\dot{\mathbf{x}} = \mathbf{A}_p \mathbf{x} + \mathbf{B}_p \mathbf{u}$ (19)

We further assume that $c_r^s(k) + p_r^s(k) \leq \Delta_r$, $a_m(k) < T$, and $\max(a_m) < T - \max(s_r)$, for $r = 1, \dots, 4$ and $m = 1, 2$, respectively. These assumptions can be achieved by properly selecting the sampling period T given sensing and actuation delays. Based on Assumptions 4 and 5 in Section 3, we have the following relations: $s_r = \Delta_r$, for $r = 1, \dots, 4$. Hence, from Eq. (3), the plant input signals $u_1(t)$ and $u_2(t)$ can be described as follows, for $kT \leq t < (k+1)T$:

$$\begin{aligned} u_1(t) &= v_1(k-1)\mathbf{1}_{(0,a_1)}(t) + v_1(k)\mathbf{1}_{(a_1,T)}(t) \\ u_2(t) &= v_2(k-1)\mathbf{1}_{(0,a_2)}(t) + v_2(k)\mathbf{1}_{(a_2,T)}(t) \end{aligned}$$

If these actuator delays a_m are constant and known in advance, says $a_1 = 1$ *ms* and $a_2 = 2$ *ms*, by applying Eq. (9), we can obtain the discrete-time plant model (at $T = 10$ *ms*) as follows:

$$\mathbf{x}_{k+1} = \begin{bmatrix} 1 & 0.0091 & 0 & 0 \\ 0 & 0.8338 & 0 & 0 \\ 0 & 0 & 1 & 0.0092 \\ 0 & 0 & 0 & 0.8365 \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} 0.0198 & 0 \\ 4.2788 & 0 \\ 0 & 0.0158 \\ 0 & 3.8547 \end{bmatrix} \mathbf{v}_k + \begin{bmatrix} 0.0045 & 0 \\ 0.4336 & 0 \\ 0 & 0.0086 \\ 0 & 0.8807 \end{bmatrix} \mathbf{v}_{k-1}.$$

We can further calculate the signal $x_r(kT - s_r)$ by obtaining the sensing delays of s_r , say $s_1 = 3$, $s_2 = 4$, $s_3 = 5$, and $s_4 = 6$ (*ms*). For example, by applying Eq. (11), $x_1(kT - s_1)$ can be described as follows:

$$x_1(kT - s_1) = [1 \ 0.0066 \ 0 \ 0] \mathbf{x}_{k-1} + [0.0198 \ 0] \mathbf{v}_{k-1} + [0.0045 \ 0] \mathbf{v}_{k-2}.$$

Therefore, Eq. (12) becomes:

$$\begin{aligned} \mathbf{w}_k &= \begin{bmatrix} w_1(k) \\ w_2(k) \\ w_3(k) \\ w_4(k) \end{bmatrix} = \begin{bmatrix} x_1(kT - s_1) \\ x_2(kT - s_2) \\ x_3(kT - s_3) \\ x_4(kT - s_4) \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0.0066 & 0 & 0 \\ 0 & 0.8966 & 0 & 0 \\ 0 & 0 & 1 & 0.0048 \\ 0 & 0 & 0 & 0.9311 \end{bmatrix} \mathbf{x}_{k-1} + \begin{bmatrix} 0.0089 & 0 \\ 2.4632 & 0 \\ 0 & 0.0023 \\ 0 & 1.0159 \end{bmatrix} \mathbf{v}_{k-1} + \begin{bmatrix} 0.0032 & 0 \\ 0.4663 & 0 \\ 0 & 0.0040 \\ 0 & 0.9803 \end{bmatrix} \mathbf{v}_{k-2}. \end{aligned}$$

In order to validate the stability and performance of standard controller design, we first consider a memoryless state feedback controller, i.e., $\mathbf{u}(t) = -K\mathbf{x}(t)$, where K is designed based on the pole placement in continuous time domain. In this case, \mathbf{F} , \mathbf{G} , and \mathbf{H} are zero matrices of compatible dimension, but $\mathbf{J} = -K$. Therefore, by letting $\mathbf{X}_k = [\mathbf{x}_k^T \mathbf{x}_{k-1}^T \mathbf{z}_k^T \mathbf{v}_{k-1}^T \mathbf{v}_{k-2}^T]^T$, the closed-loop dynamics can be obtained as follows:

$$\mathbf{X}_{k+1} = \Xi \mathbf{X}_k \quad (20)$$

In this example, the system dimension is 13. However, if there are no time delays, then the system dimension becomes 7, i.e., $\mathbf{X}_k = [\mathbf{x}_k^T \mathbf{z}_k^T \mathbf{v}_{k-1}^T]^T$. The 13 eigenvalues ('o') of Ξ are plotted in Fig. 3 along with the 7 eigenvalues ('x') of the closed-loop system without delays. Figs. 3(a) and 3(b) show two different feedback gains K . The values of time delays ($s_r, r = 1, \dots, 4$ and $a_m, m = 1, 2$) are identical in Figs. 3(a) and 3(b). In each plot, the dotted lines are the real and imaginary axes and the solid line is the unit circle. The 'o' symbols are the locations of the eigenvalues of the closed-loop system with delays and the 'x' symbols are those without delays. However, only the four right-most points map to the the eigenvalues of original closed-loop system, i.e., $\text{eig}(A_p - B_p K)$. From this comparison, we find that the closed-loop systems with multiple time delays will perform differently if the controller designer does not consider these time delays at the first design stage. In fact, at some combination of different time delays and sampling periods, the closed-loop systems could be unstable.

6 Optimal controller design

In this section, we formulate and solve the optimal controller design problem for the NCS model presented in Section 4. The standard control algorithm utilizes the state values at the sampling instants as controller inputs. In an NCS, the sensor signals are not all sampled at the same instants. Hence, in order to apply the Linear Quadratic Regulator (LQR) optimal controller design procedure, we develop a delay transformation which maps an NCS model into a standard control model with delayed states as state variables. For the ease of presentation, we only consider the case where sensor and actuator delays are constant and $\mathbf{y} = \mathbf{x}$, i.e., $N = R$.

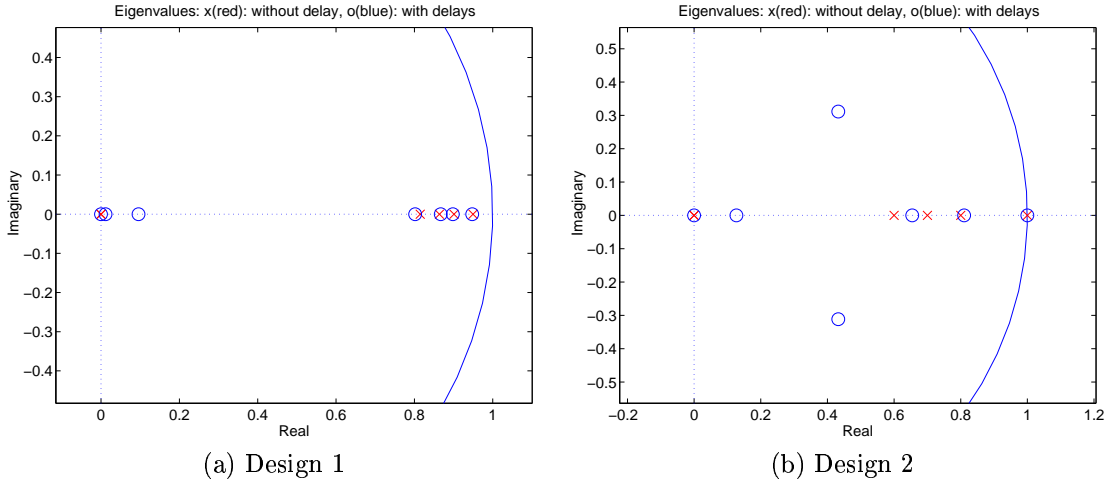


Figure 3: The location of eigenvalues of closed-loop systems.

The optimal control derivation proceeds as follows. First, we incorporate the sensing and actuation delays into the system model as described in Section 4. This results in a delayed state-variable model with the same dimension, ($\mathbf{x} \in \mathbb{R}^N$) as the original system; however, the \mathbf{A} and \mathbf{B} matrices have changed to include the delays. If the delays are constant, as is assumed here, the delayed state-variable model is time-invariant. Standard optimal control techniques are then applied to this delayed state-variable model.

Recall that $\mathbf{x}(kT)$ represents the actual values of the system states at the controller sample times; this value is not measured. $x_r(kT - s_r), r = 1, \dots, R$ are the values of the r th state received at the controller. Suppose that the sensor delays are all less than one sampling period, i.e., $s_r < T, r = 1, \dots, R$. As stated in Eq. (11), we use $\mathbf{x}(k - \bar{s}_r)$, where $\bar{s}_r = s_r/T$, to denote the true value of the continuous state at time $(k - \bar{s}_r)T$. Using Eq. (9), we can express the value of the state at the sensor sampling instants, $\mathbf{x}(k + 1 - \bar{s}_r)$ as follows:

$$\mathbf{x}(k + 1 - \bar{s}_r) = \mathbf{A}\mathbf{x}(k - \bar{s}_r) + \sum_{j=0}^2 \mathbf{B}_r^j \mathbf{v}(k - j), \quad (21)$$

where $\mathbf{B}_r^j = [\mathbf{B}_{1r}^j, \dots, \mathbf{B}_{mr}^j, \dots, \mathbf{B}_{Mr}^j]$, and $\mathbf{B}_{mr}^j(k)$'s are defined as follows:

If $T - s_r \geq a_m$, then

$$\begin{aligned} \mathbf{B}_{mr}^0(k) &\triangleq \int_{a_m + s_r}^T \mathbf{\Gamma}_m(T, q) dq, \\ \mathbf{B}_{mr}^1(k) &\triangleq \int_0^{a_m + s_r} \mathbf{\Gamma}_m(T, q) dq, \\ \mathbf{B}_{mr}^2(k) &\triangleq \mathbf{0}. \end{aligned}$$

If $T - s_r < a_m$, then

$$\mathbf{B}_{mr}^0(k) \triangleq \mathbf{0},$$

$$\begin{aligned}\mathbf{B}_{mr}^1(k) &\triangleq \int_{a_m+s_r-T}^T \mathbf{\Gamma}_m(T, q) dq, \\ \mathbf{B}_{mr}^2(k) &\triangleq \int_0^{a_m+s_r-T} \mathbf{\Gamma}_m(T, q) dq.\end{aligned}$$

Note that the actuator delays are taken care of in the \mathbf{B} matrix terms. The dependence of \mathbf{B}_r^j on the time k is dropped because all delays are assumed constant in this section. Since the sensor delays are different for each measurement, we need to carefully extract the correct dynamics for each state. The r th element of the vector $\mathbf{x}(\cdot)$, i.e., the individual state $x_r(\cdot)$, can be described as follows:

$$\begin{aligned}x_r(k+1-\tilde{s}_r) &= \mathbf{A}^{<r,r>} x_r(k-\tilde{s}_r) + \mathbf{A}^{<r,-r>} \mathbf{x}_{<-r>}(k-\tilde{s}_r) \\ &+ \sum_{j=0}^2 \mathbf{B}_r^{j<r,*>} \mathbf{v}(k-j),\end{aligned}\quad (22)$$

where $\mathbf{A}^{<r,r>} \in \mathbb{R}$ is the (r,r) th element of matrix \mathbf{A} , $\mathbf{A}^{<r,-r>} \in \mathbb{R}^{1 \times (R-1)}$ is the r th row with the r th element deleted, $\mathbf{B}^{<r,*>} \in \mathbb{R}^{1 \times M}$ is the r th row of matrix \mathbf{B} , and $\mathbf{x}_{<-r>}$ is the \mathbf{x} vector with the r th element deleted. $x_r(k-\tilde{s}_r)$ are the sensed values of the r th state that are sampled and sent over the network.

In the meantime, by using Eq. (10) and letting $\delta = \tilde{s}_r$, we can also express the sensed state $\mathbf{x}(k+1-\tilde{s}_r)$, $r = 1, \dots, R$, in terms of the state of the system at the controller sample instant, $\mathbf{x}(k)$, as follows:

$$\mathbf{x}(k+1-\tilde{s}_r) = \mathbf{A}_{\tilde{s}_r} \mathbf{x}(k) + \sum_{j=0}^1 \mathbf{B}_{\tilde{s}_r}^j \mathbf{v}(k-j).\quad (23)$$

Then, we can separate the r th element from the rest of the vector $\mathbf{x}(\cdot)$ as follows:

$$x_r(k+1-\tilde{s}_r) = \mathbf{A}_{\tilde{s}_r}^{<r,*>} \mathbf{x}(k) + \sum_{j=0}^1 \mathbf{B}_{\tilde{s}_r}^{j<r,*>} \mathbf{v}(k-j),\quad (24)$$

$$\text{and } \mathbf{x}_{<-r>}(k+1-\tilde{s}_r) = \mathbf{A}_{\tilde{s}_r}^{<-r,*>} \mathbf{x}(k) + \sum_{j=0}^1 \mathbf{B}_{\tilde{s}_r}^{j<-r,*>} \mathbf{v}(k-j).\quad (25)$$

In order to formulate the system dynamics in terms of measured states, we further define the following new variables:

$$\mathbf{x}_s(k) = \begin{bmatrix} x_1(k-s_1) \\ x_2(k-s_2) \\ \vdots \\ x_R(k-s_R) \end{bmatrix}, \quad \mathbf{A}_{s*} = \begin{bmatrix} \mathbf{A}_{\tilde{s}_1}^{<1,*>} \\ \mathbf{A}_{\tilde{s}_2}^{<2,*>} \\ \vdots \\ \mathbf{A}_{\tilde{s}_R}^{<R,*>} \end{bmatrix}, \quad \text{and } \mathbf{B}_{s*}^j = \begin{bmatrix} \mathbf{B}_{\tilde{s}_1}^{j<1,*>} \\ \mathbf{B}_{\tilde{s}_2}^{j<2,*>} \\ \vdots \\ \mathbf{B}_{\tilde{s}_R}^{j<R,*>} \end{bmatrix}.$$

$\mathbf{x}_s(k)$ will be the state variables of the new delayed state-variable model. By applying Eq. (24) for $r = 1, \dots, R$, we have the following equation:

$$\mathbf{x}_s(k+1) = \mathbf{A}_{s*} \mathbf{x}_s(k) + \sum_{j=0}^1 \mathbf{B}_{s*}^j \mathbf{v}(k-j).\quad (26)$$

Since \mathbf{A}_{s*} is the combination of $\mathbf{A}_{\tilde{s}_r}$, i.e., $\exp(A_p \tilde{s}_r T)$, which is nonsingular, we can multiply both sides of Eq. (26) by \mathbf{A}_{s*}^{-1} , and obtain the actual value of the state at the controller sample instant as a function of

the sensed (delayed) values of the states and the inputs.

$$\mathbf{x}(k) = \mathbf{A}_{s^*}^{-1} \mathbf{x}_s(k+1) - \mathbf{A}_{s^*}^{-1} \sum_{j=0}^1 \mathbf{B}_{s^*}^j \mathbf{v}(k-j). \quad (27)$$

We further apply Eq. (27) to Eq. (25) at time $k-1$ to find the values of the other system states at the instant when the r th state is sampled.

$$\begin{aligned} \mathbf{x}_{<-r>}(k - \tilde{s}_r) &= \mathbf{A}_{\tilde{s}_r}^{<-r,*>} \mathbf{x}(k-1) + \sum_{j=0}^1 \mathbf{B}_{\tilde{s}_r}^{j<-r,*>} \mathbf{v}(k-1-j) \\ &= \mathbf{A}_{\tilde{s}_r}^{<-r,*>} \left[\mathbf{A}_{s^*}^{-1} \mathbf{x}_s(k) - \mathbf{A}_{s^*}^{-1} \sum_{j=0}^1 \mathbf{B}_{s^*}^j \mathbf{v}(k-1-j) \right] \\ &\quad + \sum_{j=0}^1 \mathbf{B}_{\tilde{s}_r}^{j<-r,*>} \mathbf{v}(k-1-j) \\ &= [\mathbf{A}_{\tilde{s}_r}^{<-r,*>} \mathbf{A}_{s^*}^{-1}] \mathbf{x}_s(k) + \sum_{j=0}^1 [\mathbf{B}_{\tilde{s}_r}^{j<-r,*>} - \mathbf{A}_{\tilde{s}_r}^{<-r,*>} \mathbf{A}_{s^*}^{-1} \mathbf{B}_{s^*}^j] \mathbf{v}(k-1-j) \\ &= \mathbf{A}_{\tilde{s}_r}^{x_r} \mathbf{x}_s(k) + \sum_{j=0}^1 \mathbf{B}_{\tilde{s}_r}^{x_r j} \mathbf{v}(k-1-j), \end{aligned} \quad (28)$$

where the second equality is from Eq. (27) at time $k-1$, and, in the fourth equality, we use the following definitions: $\mathbf{A}_{\tilde{s}_r}^{x_r} \triangleq \mathbf{A}_{\tilde{s}_r}^{<-r,*>} \mathbf{A}_{s^*}^{-1}$, and $\mathbf{B}_{\tilde{s}_r}^{x_r j} \triangleq \mathbf{B}_{\tilde{s}_r}^{j<-r,*>} - \mathbf{A}_{\tilde{s}_r}^{<-r,*>} \mathbf{A}_{s^*}^{-1} \mathbf{B}_{s^*}^j$. Hence, by further plugging Eq. (28) into Eq. (22), we have:

$$\begin{aligned} x_r(k+1 - \tilde{s}_r) &= \mathbf{A}^{<r,r>} x_r(k - \tilde{s}_r) + \mathbf{A}^{<r,-r>} \left\{ [\mathbf{A}_{\tilde{s}_r}^{<-r,*>} \mathbf{A}_{s^*}^{-1}] \mathbf{x}_s(k) \right. \\ &\quad \left. + \sum_{j=0}^1 [\mathbf{B}_{\tilde{s}_r}^{j<-r,*>} - \mathbf{A}_{\tilde{s}_r}^{<-r,*>} \mathbf{A}_{s^*}^{-1} \mathbf{B}_{s^*}^j] \mathbf{v}(k-1-j) \right\} \\ &\quad + \sum_{j=0}^2 \mathbf{B}_r^{j<r,*>} \mathbf{v}(k-j), \end{aligned} \quad (29)$$

Therefore, by using the definition of $\mathbf{x}_s(k)$, we can obtain the new delayed state-variable model as follows:

$$\begin{aligned} \mathbf{x}_s(k+1) &= \begin{bmatrix} x_1(k+1 - \tilde{s}_1) \\ x_2(k+1 - \tilde{s}_2) \\ \vdots \\ x_R(k+1 - \tilde{s}_R) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{A}^{<1,1>} & 0 & 0 & 0 \\ 0 & \mathbf{A}^{<2,2>} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \mathbf{A}^{<R,R>} \end{bmatrix} \mathbf{x}_s(k) \\ &\quad + \begin{bmatrix} \mathbf{A}^{<1,-1>} \mathbf{A}_{\tilde{s}_1}^{<-1,*>} \mathbf{A}_{s^*}^{-1} \\ \mathbf{A}^{<2,-2>} \mathbf{A}_{\tilde{s}_2}^{<-2,*>} \mathbf{A}_{s^*}^{-1} \\ \vdots \\ \mathbf{A}^{<R,-R>} \mathbf{A}_{\tilde{s}_R}^{<-R,*>} \mathbf{A}_{s^*}^{-1} \end{bmatrix} \mathbf{x}_s(k) + \begin{bmatrix} \mathbf{B}_1^{0<1,*>} \\ \mathbf{B}_2^{0<2,*>} \\ \vdots \\ \mathbf{B}_R^{0<R,*>} \end{bmatrix} \mathbf{v}(k) \end{aligned}$$

$$+ \sum_{j=1}^2 \begin{bmatrix} \mathbf{B}_1^{j<1,*>} + \mathbf{A}^{<1,-1>} \left[\mathbf{B}_{\tilde{s}_1}^{j-1<-1,*>} - \mathbf{A}_{\tilde{s}_1}^{<-1,*>} \mathbf{A}_{s^*}^{-1} \mathbf{B}_{s^*}^{j-1} \right] \\ \mathbf{B}_2^{j<2,*>} + \mathbf{A}^{<2,-2>} \left[\mathbf{B}_{\tilde{s}_2}^{j-1<-2,*>} - \mathbf{A}_{\tilde{s}_2}^{<-2,*>} \mathbf{A}_{s^*}^{-1} \mathbf{B}_{s^*}^{j-1} \right] \\ \vdots \\ \mathbf{B}_R^{j<R,*>} + \mathbf{A}^{<R,-R>} \left[\mathbf{B}_{\tilde{s}_R}^{j-1<-R,*>} - \mathbf{A}_{\tilde{s}_R}^{<-R,*>} \mathbf{A}_{s^*}^{-1} \mathbf{B}_{s^*}^{j-1} \right] \end{bmatrix} \mathbf{v}(k-j)$$

$$\text{or } \mathbf{x}_s(k+1) = \mathbf{A}_{x_s} \mathbf{x}_s(k) + \sum_{j=0}^2 \mathbf{B}_{x_s}^j \mathbf{v}(k-j) \quad (30)$$

where

$$\mathbf{A}_{x_s} = \begin{bmatrix} \mathbf{A}^{(1,1)} & 0 & 0 & 0 \\ 0 & \mathbf{A}^{(2,2)} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \mathbf{A}^{(R,R)} \end{bmatrix} + \begin{bmatrix} \mathbf{A}^{<1,-1>} \mathbf{A}_{\tilde{s}_1}^{<-1,*>} \mathbf{A}_{s^*}^{-1} \\ \mathbf{A}^{<2,-2>} \mathbf{A}_{\tilde{s}_2}^{<-2,*>} \mathbf{A}_{s^*}^{-1} \\ \vdots \\ \mathbf{A}^{<R,-R>} \mathbf{A}_{\tilde{s}_R}^{<-R,*>} \mathbf{A}_{s^*}^{-1} \end{bmatrix}, \text{ and}$$

$$\mathbf{B}_{x_s}^j = \begin{cases} \begin{bmatrix} \mathbf{B}_1^{0<1,*>} \\ \mathbf{B}_2^{0<2,*>} \\ \vdots \\ \mathbf{B}_R^{0<R,*>} \end{bmatrix}, & j=0 \\ \begin{bmatrix} \mathbf{B}_1^{j<1,*>} + \mathbf{A}^{<1,-1>} \left[\mathbf{B}_{\tilde{s}_1}^{j-1<-1,*>} - \mathbf{A}_{\tilde{s}_1}^{<-1,*>} \mathbf{A}_{s^*}^{-1} \mathbf{B}_{s^*}^{j-1} \right] \\ \mathbf{B}_2^{j<2,*>} + \mathbf{A}^{<2,-2>} \left[\mathbf{B}_{\tilde{s}_2}^{j-1<-2,*>} - \mathbf{A}_{\tilde{s}_2}^{<-2,*>} \mathbf{A}_{s^*}^{-1} \mathbf{B}_{s^*}^{j-1} \right] \\ \vdots \\ \mathbf{B}_R^{j<R,*>} + \mathbf{A}^{<R,-R>} \left[\mathbf{B}_{\tilde{s}_R}^{j-1<-R,*>} - \mathbf{A}_{\tilde{s}_R}^{<-R,*>} \mathbf{A}_{s^*}^{-1} \mathbf{B}_{s^*}^{j-1} \right] \end{bmatrix}, & j=1, 2. \end{cases}$$

We further define $\mathbf{z}(k) = [\mathbf{x}_s(k)^T, \mathbf{v}(k-1)^T, \mathbf{v}(k-2)^T]^T \in \mathbb{R}^{N+2R}$. $\mathbf{z}(k)$ will be the new state variables in the standard optimal control design technique. Then, we have the following state-space model for controller design:

$$\begin{aligned} \mathbf{z}(k+1) &= \begin{bmatrix} \mathbf{A}_{x_s} & \mathbf{B}_{x_s}^1 & \mathbf{B}_{x_s}^2 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \end{bmatrix} \mathbf{z}(k) + \begin{bmatrix} \mathbf{B}_{x_s}^0 \\ \mathbf{I} \\ \mathbf{0} \end{bmatrix} \mathbf{v}(k) \\ &= \mathbf{A}_z \mathbf{z}(k) + \mathbf{B}_z \mathbf{v}(k) \end{aligned} \quad (31)$$

where

$$\mathbf{A}_z = \begin{bmatrix} \mathbf{A}_{x_s} & \mathbf{B}_{x_s}^1 & \mathbf{B}_{x_s}^2 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{(N+2R) \times (N+2R)}, \quad \text{and} \quad \mathbf{B}_z = \begin{bmatrix} \mathbf{B}_{x_s}^0 \\ \mathbf{I} \\ \mathbf{0} \end{bmatrix} \in \mathbb{R}^{(N+2R) \times R}.$$

For the controller design of tracking problem, we first let the desired trajectory $\mathbf{d}(t), \mathbf{d}(k)$ be described by the following equation:

$$\begin{aligned} \dot{\mathbf{d}}(t) &= \mathbf{A}_d^c \mathbf{d}(t), \quad \text{in continuous time,} \\ \text{or } \mathbf{d}(k+1) &= \mathbf{A}_d^d \mathbf{d}(k), \quad \text{in discrete time,} \end{aligned} \quad (32)$$

where $\mathbf{d}(\cdot) = [d_1(\cdot), \dots, d_R(\cdot)]^T$. Furthermore, since the new state variables are measured at time $k - \tilde{s}_r$, we define the desired trajectory at each individual time instant $k - \tilde{s}_r$ as $\mathbf{d}_s(k) \triangleq [d_1(k - \tilde{s}_1), \dots, d_R(k - \tilde{s}_R)]^T$.

For the optimal controller design, we then choose the following cost function V :

$$\begin{aligned} V &= \sum_{k=0}^N \left\{ [\mathbf{x}_s(k) - \mathbf{d}_s(k)]^T \mathbf{Q}_s [\mathbf{x}_s(k) - \mathbf{d}_s(k)] + \sum_{i=0}^2 \mathbf{v}(k-i)^T \mathbf{R}_i \mathbf{v}(k-i) \right\} \\ &= \sum_{k=0}^N [\mathbf{z}(k) - \mathbf{z}_d(k)]^T \mathbf{Q} [\mathbf{z}(k) - \mathbf{z}_d(k)] + \mathbf{v}(k)^T \mathbf{R} \mathbf{v}(k), \end{aligned} \quad (33)$$

where $\mathbf{z}_d(k) \triangleq [\mathbf{d}_s(k)^T, \mathbf{0}^T]^T$, i.e., the combined vector of desired trajectory and zero input, and \mathbf{Q}_s and \mathbf{R}_i are the weighting matrices for the original system states and inputs, respectively. \mathbf{Q} and \mathbf{R} are further defined as follows:

$$\mathbf{R} = \mathbf{R}_0, \quad \text{and} \quad \mathbf{Q} = \begin{bmatrix} \mathbf{Q}_s & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R}_2 \end{bmatrix}.$$

Based on the standard LQR algorithm [16], the optimal controller law $\mathbf{v}^*(k)$ can be obtained as follows:

$$\begin{aligned} \mathbf{v}^*(k) &= -\mathbf{K}(k)[\mathbf{z}(k) - \mathbf{z}_d(k)] \\ &= -\mathbf{K}_s(k)[\mathbf{x}_s(k) - \mathbf{d}_s(k)] - \sum_{i=1}^2 \mathbf{K}_v^i(k) \mathbf{v}^*(k-i), \end{aligned} \quad (34)$$

where $\mathbf{K} = [\mathbf{K}_s, \mathbf{K}_v^1, \mathbf{K}_v^2]$. The new optimal controller utilizes the available measured states as well as past inputs as the controller inputs. The number of past inputs depends on the sensor and actuator delays as described in Eqs. (14) and (15). An NCS-LQR design example for the illustrative example presented in Section 5 will be presented in the next section.

7 Numerical examples of NCS-LQR

In this section, we revisit the two-axis example in Section 5. Three cases of the optimal controller design will be compared. The first case applies the standard LQR controller for the original delay-free system. In the second case, the LQR controller that is designed based on delay-free system is used to control the delayed system. The final case considers the LQR design based on the delayed state-variable model discussed in Section 6.

The actuation and sensing delays are assumed as follows: $a_1 = 1, a_2 = 2, s_1 = 3, s_2 = 4, s_3 = 5$, and $s_4 = 6$, (ms) and the sampling time is $10 ms$. The choice of this delay characteristics is based on the performance analysis of one type of priority-based control networks such as DeviceNet [18]. Also, based on the assumptions used in Section 5, Eq. (30) has the following form:

$$\mathbf{x}_s(k+1) = \begin{bmatrix} 1 & 0.0090 & 0 & 0 \\ 0 & 0.8338 & 0 & 0 \\ 0 & 0 & 1 & 0.0090 \\ 0 & 0 & 0 & 0.8365 \end{bmatrix} \mathbf{x}_s(k)$$

$$+ \begin{bmatrix} 0.0323 & 0 \\ 3.9787 & 0 \\ 0 & 0.0342 \\ 0 & 3.4630 \end{bmatrix} \mathbf{v}(k) + \begin{bmatrix} 0.0104 & 0 \\ 0.4032 & 0 \\ 0 & 0.0175 \\ -0.0088 & 0.7912 \end{bmatrix} \mathbf{v}(k-1) + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \mathbf{v}(k-2).$$

Furthermore, the new delayed state-variable model, Eq. (31), becomes:

$$\mathbf{z}(k+1) = \begin{bmatrix} 1 & 0.0090 & 0 & 0 & 0.0104 & 0 & 0 & 0 \\ 0 & 0.8338 & 0 & 0 & 0.4032 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.0090 & 0 & 0.0175 & 0 & 0 \\ 0 & 0 & 0 & 0.8365 & -0.0088 & 0.7912 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \mathbf{z}(k) + \begin{bmatrix} 0.0323 & 0 \\ 3.9787 & 0 \\ 0 & 0.0342 \\ 0 & 3.4630 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \mathbf{v}(k).$$

For the LQR optimal controller design, we first choose the following weighting matrices:

$$\mathbf{Q}_s = \text{diag}(20, 0.05, 20, 0.05),$$

$$\mathbf{R}_0 = \text{diag}(0.1, 0.1),$$

$$\mathbf{R}_1 = \text{diag}(0.1, 0.1),$$

$$\text{and } \mathbf{R}_2 = \text{diag}(0.001, 0.001).$$

Most of the cost function is weighted upon position tracking. Since $\mathbf{v}(k)$, $\mathbf{v}(k-1)$, and $\mathbf{v}(k-2)$ are not independent, we discount the weights on previous inputs. By using the discrete-time system matrix \mathbf{A} and input matrix \mathbf{B} , and the weighting matrices \mathbf{Q}_s and \mathbf{R}_0 , the LQR state feedback gain, called \mathbf{K}_{lqr} , is computed as follows:

$$\mathbf{K}_{lqr} = \begin{bmatrix} 3.6038 & 0.1829 & 0 & 0 \\ 0 & 0 & 3.5879 & 0.1827 \end{bmatrix}. \quad (35)$$

The gain is the same as the standard LQR design for the delay-free systems. On the other hand, by using the NCS system matrix \mathbf{A}_z , input matrix \mathbf{B}_z , and the weighting matrices \mathbf{Q} and \mathbf{R} , the NCS-LQR state feedback gain, called \mathbf{K}_{lqr}^{ncs} , is obtained as follows:

$$\mathbf{K}_{lqr}^{ncs} = \begin{bmatrix} 3.7917 & 0.1802 & 0 & 0 & 0.1102 & 0 & 0 & 0 \\ 0.0002 & 0 & 4.0722 & 0.1881 & -0.0016 & 0.2145 & 0 & 0 \end{bmatrix}.$$

\mathbf{K}_{lqr}^{ncs} is computed based on the proposed NCS delayed state-variable model discussed in Section 6.

Simulation results of the first LQR design are shown in Figs. 4–6. Fig. 4 shows the location of eigenvalues of three different closed-loop systems: (1) the discrete-time LQR (DT-LQR) controller for the delay-free system, (2) the DT-LQR controller for the delayed system, and (3) the NCS-LQR controller for the delayed system. The eigenvalue locations for the NCS-LQR case are closer to those of the DT-LQR for the delay-free system than those of the DT-LQR for the delayed system.

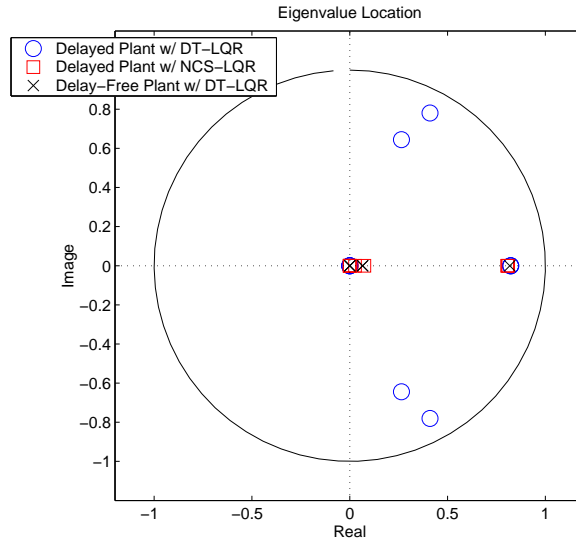


Figure 4: Eigenvalue location of three different closed-loop systems: \mathbf{K}_{lqr} for delay-free system, \mathbf{K}_{lqr} for delayed system, and \mathbf{K}_{lqr}^{ncs} for delayed system.

Fig. 5 shows the simulation results of the tracking performance of the two-axis system, i.e., X and Y axes, with the DT-LQR controller for the delay-free system. Here, two constant feed-rates for forward and backward motions are considered as the desired trajectory for both X and Y axes. For this standard LQR case where time delays are not considered, $\mathbf{v}(k) = -\mathbf{K}_{lqr}[\mathbf{x}(k) - \mathbf{d}(k)]$ is used as controller input. The plot on the left-hand side is the top-view trajectory, and the four plots on the right-hand side are the positions and velocities of X and Y axes. This case can be used as a basis for comparison. Fig. 6 shows the summary of the tracking errors with different indexes. The left-hand plot is the root-mean-square tracking errors of (1) the entire simulation time (0-7 sec), (2) the first second (0-1 sec), and (3) the last two seconds (5-7 sec). The right-hand plot is the result of the integral of the time multiplied by the absolute values of the error (ITAE). The NCS-LQR controller improves the control performance over the DT-LQR controller when considering the time delays.

The simulation results of two other desired trajectories, a circle with a radius of 100 mm, and constant trajectories for both position and velocity, are also shown in Figs. 7–10. The circular trajectory can be viewed as a time-varying trajectory; the constant trajectory is a regulation case for both position and velocity. For both types of trajectories, the proposed NCS-LQR guarantees better control performance than the DT-LQR. Especially, the 2-D top-view plots in Figs. 7 and 9 show the dramatic improvement of the proposed NCS-LQR controller design.

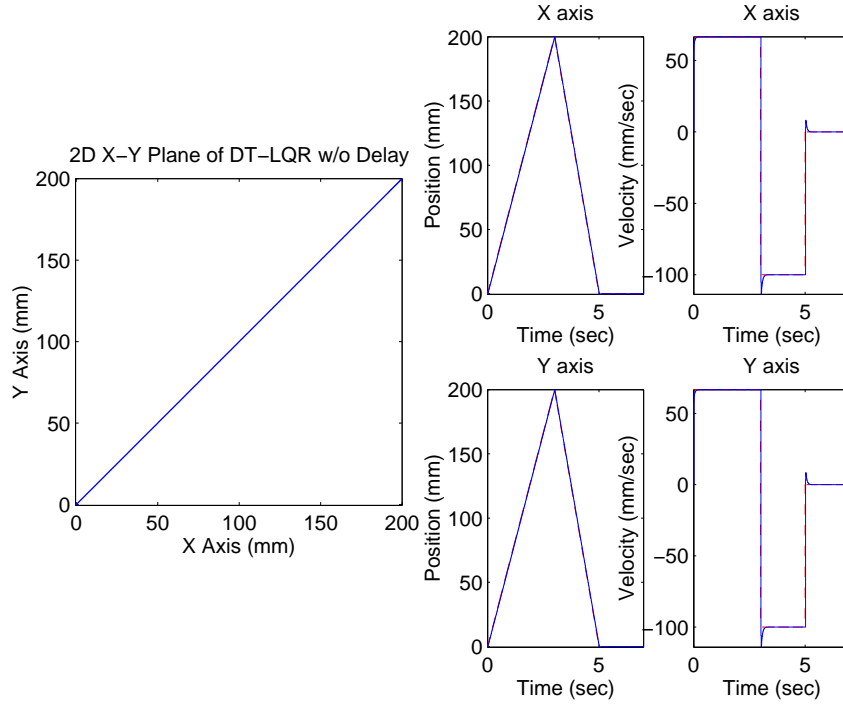


Figure 5: Standard LQR controller design: \mathbf{K}_{lqr} for delay-free system.

8 Control performance analysis of the proposed optimal control

A NCS design chart proposed by Lian et al. could be used to choose network and control parameters [21]. Based on the design chart, for example, sampling time, node number and device priority could be re-designed to meet the required control and network performance. The NCS design chart in [21] focused on the design considerations; the controllers are assumed to be pre-designed and may or may not take into account the network-induced delays. In Section 6 we proposed a delayed state-variable model and an optimal control design technique for NCSs. We present a simulation study showing how the control performance is improved with this NCS optimal control design technique.

In order to demonstrate the performance improvement, we consider the X-axis subsystem presented in Sections 5 and 7. This X-axis subsystem has one actuator and two sensors (position and velocity). A priority-based DeviceNet network is used to interconnect these three nodes [18]. A data rate of 100 *Kbps* is used and the node numbering for these three nodes are as follows: 1. actuator, 2. position, and 3. velocity; i.e., the priority is “actuator” > “position” > “velocity.” The message transmission times for these three nodes are all equal to 1 *ms*. If the network is not saturated, the time delays of these three nodes are 1, 2, and 3 *ms*, respectively. On the other hand, if the message period is smaller than 3 *ms*, then the network is saturated and the time delays become time-varying.

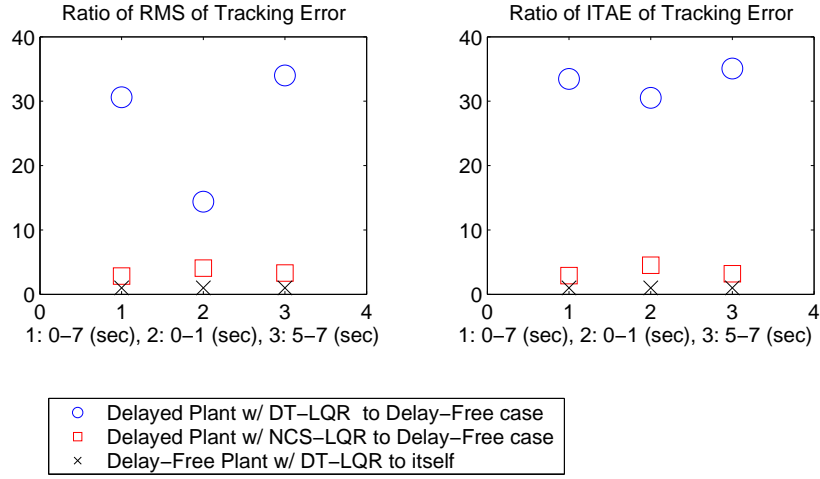


Figure 6: The summary of tracking errors: \mathbf{K}_{lqr} for delay-free system, \mathbf{K}_{lqr} for delayed system, \mathbf{K}_{lqr}^{ncs} for delayed system. Left: RMS, right: ITAE.

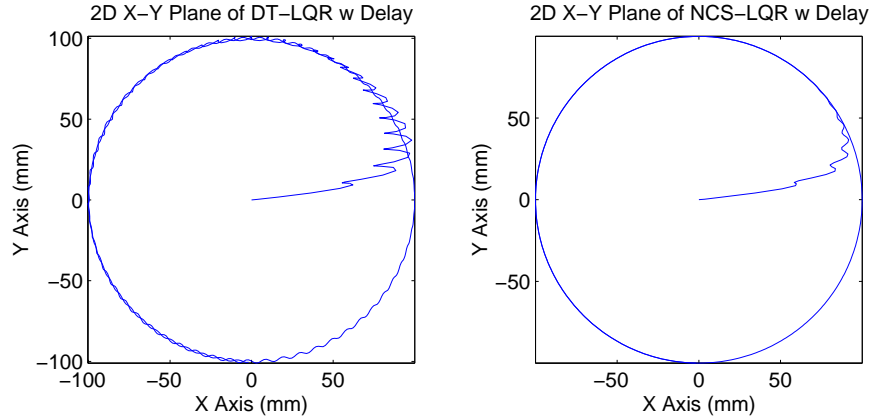


Figure 7: The 2-D top view of “circular” X-Y axis system: Left: \mathbf{K}_{lqr} for delayed system, Right: \mathbf{K}_{lqr}^{ncs} for delayed system.

The simulation results are shown in Fig. 11, which considers sampling periods from 100 to 2 ms and three cases of controller design. The first case is a discrete-time LQR controller design for a delay-free system. The simulation result is shown in Fig. 11, marked by ‘x’ symbols. This case can be viewed as the standard digital controller case. Digital controller only has the impact from sampling mechanism; hence, a smaller sampling period guarantees a better controller performance. The second case considers the same LQR controller in the first case, but the controller is used to control a delayed system. The simulation result is shown in Fig. 11, marked by ‘o’ symbols. Since this controller is designed without considering the time delay as discussed in Section 7, its performance degrades when time delays are present. Also, as the network becomes saturated, the overall control performance degrades. Based on the NCS-LQR controller design technique presented in previous sections, the control performance can be further improved. The simulation result is also shown in

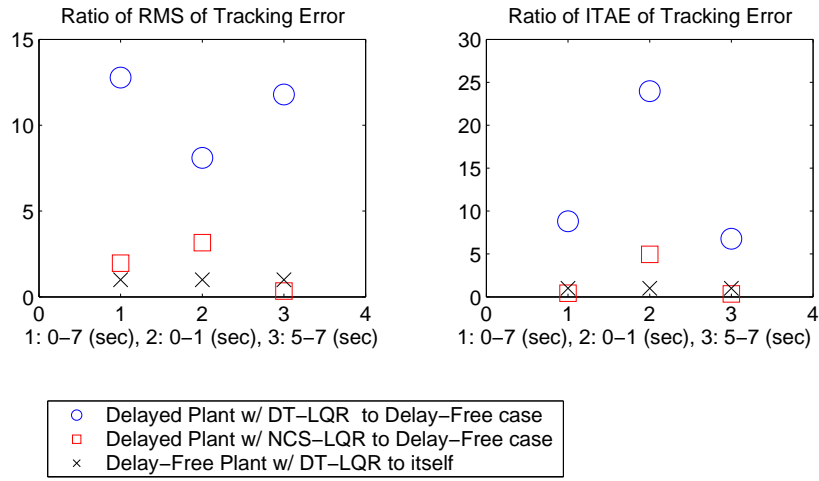


Figure 8: The summary of “circular” tracking errors: \mathbf{K}_{lqr} for delay-free system, \mathbf{K}_{lqr} for delayed system, \mathbf{K}_{lqr}^{ncs} for delayed system. Left: RMS, right: ITAE.

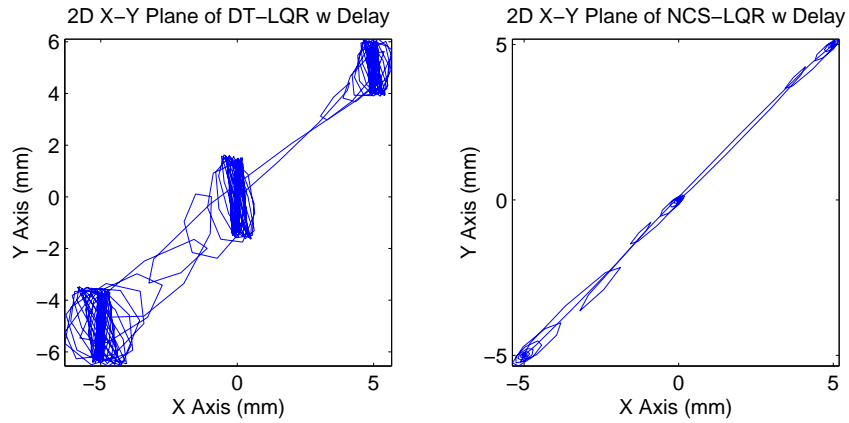


Figure 9: The 2-D top view of “constant” X-Y axis system: Left: \mathbf{K}_{lqr} for delayed system, Right: \mathbf{K}_{lqr}^{ncs} for delayed system.

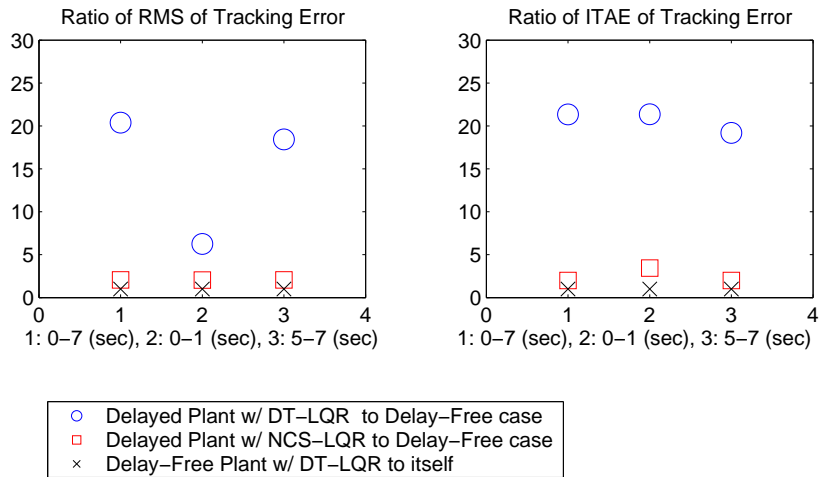


Figure 10: The summary of “constant” tracking errors: \mathbf{K}_{lqr} for delay-free system, \mathbf{K}_{lqr} for delayed system, \mathbf{K}_{lqr}^{ncs} for delayed system. Left: RMS, right: ITAE.

Fig. 11, marked by ‘□’ symbols. It can be easily seen that the NCS-LQR controller performs better than the DT-LQR controller when network-induced delays are present. When the sampling period is small and the network is not saturated, the performance of the NCS-LQR controller is similar to the standard LQR controller for a delay-free system. When the network is saturated, i.e., the sampling period is smaller than 3 ms in this case, the NCS-LQR controller still performs better than the DT-LQR controller. Therefore, in addition to properly selecting network and control parameters, an NCS-LQR controller can further improve the control performance of an NCS.

In order to verify the robustness of the proposed NCS-LQR controller, two cases of delay variance are added into the simulation model. The first case considers a constant delay variance among actuator and sensor delays; that is, $a_m + t_{dv}, m = 1, 2$ and $s_r + t_{dv}, r = 1, \dots, 4$, where t_{dv} is a constant number chosen from one of $\{0.1, 0.2, \dots, 0.9, 1, 2, 3\}$ (ms). The second case considers a random delay variance, that is, t_{dv} is a uniformly distributed random number which is chosen between 0 and one of $\{0.1, 0.2, \dots, 0.9, 1, 2, 3\}$ (ms). The simulation results of the constant delay variance (solid lines) and two cases of the random delay variance (two sets of dotted lines) at a sampling time of 10 ms are shown in Fig. 12. The tracking errors, i.e., ITAE, of all cases are normalized by that of the NCS-LQR controller without delay variance. The upper set of lines with ‘o’ and ‘*’ are the cases of the DT-LQR controller and the lower set of lines with ‘□’ and ‘×’ are the cases of the NCS-LQR controller. For small time delay variances (< 1 ms), the NCS-LQR controller maintains consistent performance and performs better than the DT-LQR controller. Also, the cases with random delay variance have better performance than the cases with constant delay variance since the former have smaller magnitude of time delays during the simulation time. In order to guarantee better control performance with time delay variance, a stochastic, robust design scheme should be further considered to compensate for distributed time delays and delay uncertainty.

9 Summary

Although the introduction of control networks to carry sensor and actuator data in control systems provides improved flexibility and reliability from a design point of view, it also induces asynchronous time delays among different devices. These time delays can degrade the system performance and may introduce instability in high bandwidth or complex systems.

In this paper we analyzed and modeled a MIMO networked control system with multiple time delays. The time delays between sensor-controller and controller-actuator and the time skews at different devices’ sampling instants were considered and included in the derivation of a discrete-time MIMO model. By

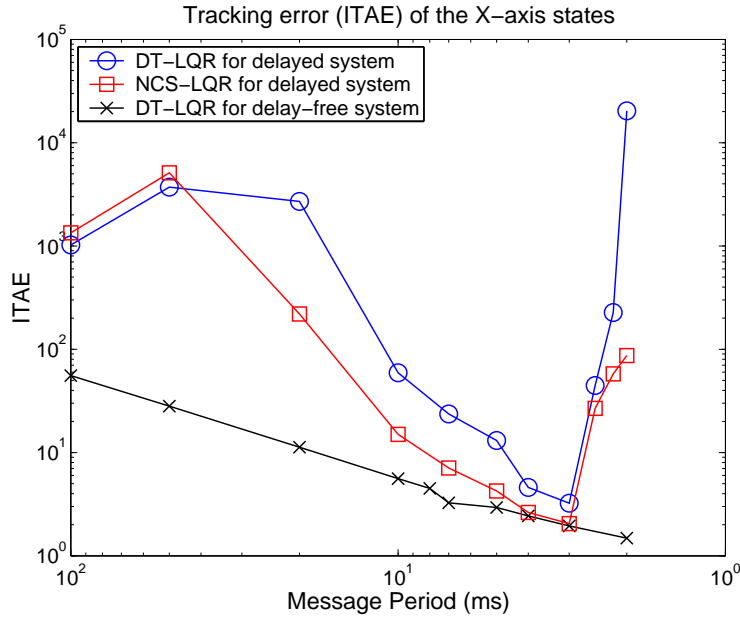


Figure 11: Comparison of digital and networked controllers: Tracking error (ITAE) v.s. message period.

including these time delay parameters, both the control system and network system designers can utilize this model to design networked control systems and optimize their overall performance.

The closed-loop NCS model discussed in Section 4 only included a standard controller designed without considering the time delay effect a priori. Therefore, this closed-loop stability analysis was used as a verification tool for the stability and performance of an NCS. Based on the time-delay modeling algorithm, a delayed state-variable model was formulated for the standard LQR controller design. This NCS-LQR controller improved the control performance over the DT-LQR controller when considering the time delays. Simulation studies provided in Sections 5, 7, and 8, demonstrated the utility of the proposed design algorithms for stability analysis and performance improvement.

The advantage of having an NCS architecture is that it provides the flexibility to quickly reconfigure the system architecture, and to easily share information with other subsystems. The change of system configuration may also change the time delay of a networked device even in a deterministic NCS. Hence, the advanced networked controller should be able to maintain a proper level of network traffic load and adaptively modify the controller algorithm. This slowly changing situation can be modeled as an adaptive control problem which can adjust on-line to the changing system parameters. Also, before a truly deterministic network is available for control applications, the time-varying delays should be included in the modeling algorithm, and a stochastic controller design could be adopted. For the uncertainties existing in an NCS such as the variance of time delays and system uncertainty, a robust controller design could then be applied.

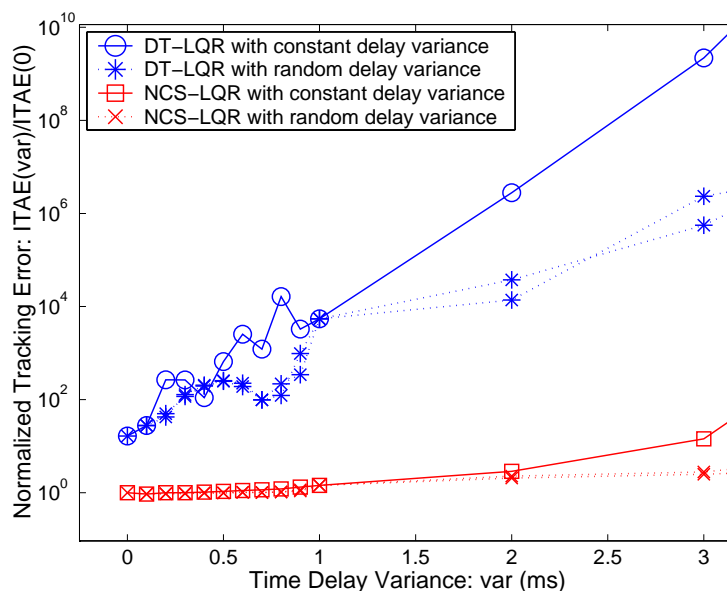


Figure 12: Robustness analysis of NCS-LQR control at a sampling time of 10 ms.

However, the stochastic or robust controller design might be conservative if the uncertainty cannot be exactly characterized. In addition, an intelligent controller design that incorporates both network and control parameters to further improve both network and control performance should be further studied. This intelligent networked controller can identify/estimate network performance and operate among multiple modes of control action, depending on network traffic or control performance.

References

- [1] M. S. Branicky, S. M. Phillips, and W. Zhang. Stability of networked control systems: explicit analysis of delay. In *Proceedings of 2000 American Control Conference*, pages 2352–2357, June 2000.
- [2] L. Dugard and E. I. Verrest, editors. *Stability and control of time-delay systems*. Springer, 1998.
- [3] G. F. Franklin, J. D. Powell, and A. Emami-Naeini. *Feedback control of dynamic systems*. Addison-Wesley, third edition, 1994.
- [4] G. F. Franklin, J. D. Powell, and M. L. Workman. *Digital control of dynamic systems*. Addison-Wesley, third edition, 1998.
- [5] F. Göktas, J. M. Smith, and R. Bajcsy. μ -synthesis for distributed control systems with network-induced delays. In *Proceedings of the 35th Conference on Decision and Control*, pages 813–814, Dec. 1996.
- [6] F. Göktas, J. M. Smith, and R. Bajcsy. Telerobotics over communication networks. In *IEEE Proceedings of the 36th Conference on Decision and Control*, pages 2399–2404, Dec. 1997.
- [7] H. Górecki, S. Fuksa, P. Grabowski, and A. Korytowski. *Analysis and synthesis of time delay systems*. John Wiley & Sons, 1989.
- [8] A. Goubet-Bartholoméüs, M. Dambrine, and J. P. Richard. Stability of perturbed systems with time-varying delays. *Systems & Control Letters*, 31:155–163, 1997.
- [9] K. Gu. Discretized Lyapunov functional for uncertain systems with multiple time-delay. *International Journal of Control*, 72(16):1436–1445, 1999.

- [10] Y. Gu, S. Wang, Q. Li, Z. Cheng, and J. Qian. On delay-dependent stability and decay estimate for uncertain systems with time-varying delay. *Automatica*, 34(8):1035–1039, Aug. 1998.
- [11] Y. Halevi and A. Ray. Integrated communication and control systems: Part I — analysis. *ASME Journal of Dynamic Systems, Measurement, and Control*, 110(4):367–373, Dec. 1988.
- [12] F.-H. Hsiao and J.-D. Hwang. Stability analysis of uncertain feedback systems with multiple time delays and series nonlinearities. *Journal of Franklin Institute*, 334B(3):491–505, 1997.
- [13] E. T. Jeung, D. C. Oh, J. H. Kim, and H. B. Park. Robust controller design for uncertain systems with time delays: LMI approach. *Automatica*, 32(8):1229–1331, Aug. 1996.
- [14] Y. H. Kim, W. H. Kwon, and H. S. Park. Stability and a scheduling method for network-based control systems. In *IECON Proceedings*, volume 2, pages 934–939, Aug. 1996.
- [15] R. Krtolica, Ü. Özgüner, H. Chan, H. Göktas, J. Winkelmann, and M. Liubakka. Stability of linear feedback systems with random communication delays. *International Journal of Control*, 59(4):925–953, Oct. 1994.
- [16] F. L. Lewis. *Optimal Control*. John Wiley & Sons, 1986.
- [17] X. Li and C. E. de Souza. Delay-dependent robust stability and stabilization of uncertain linear delay systems: a linear matrix inequality approach. *IEEE Transactions on Automatic Control*, 42(8):1144–1148, Aug. 1997.
- [18] F.-L. Lian, J. R. Moyne, and D. M. Tilbury. Performance evaluation of control networks: Ethernet, ControlNet, and DeviceNet. *IEEE Control Systems Magazine*, 21(1):66–83, Feb. 2001.
- [19] F.-L. Lian, J. R. Moyne, and D. M. Tilbury. Network design consideration for distributed control systems. *Submitted to IEEE Transactions on Control Systems Technology*, Nov. 2000.
- [20] F.-L. Lian, J. R. Moyne, and D. M. Tilbury. Analysis and modeling of networked control systems: MIMO case with multiple time delays. In *Proceedings of 2001 American Control Conference*, June 2001.
- [21] F.-L. Lian, J. R. Moyne, and D. M. Tilbury. Time delay modeling and sample time selection for networked control systems. In *Proceedings of the ASME Dynamic Systems and Control Division*, Nov. 2001.
- [22] J. S. Luo, P. P. J. van den Bosch, S. Weiland, and A. Goldenberge. Design of performance robustness for uncertain linear systems with state and control delays. *IEEE Transactions on Automatic Control*, 43(11):1593–1596, Nov. 1998.
- [23] M. Malek-Zavarei and M. Jamshidi. *Time-Delay Systems: Analysis, Optimization and Applications*. North-Holland, 1987.
- [24] J. E. Marshall. *Control of Time-Delay Systems*. Petter Peregrinus Ltd., 1979.
- [25] J. Nilsson. *Real-time control systems with delays*. PhD thesis, Lund Institute of Technology, 1998.
- [26] J. Nilsson and B. Bernhardsson. LQG control over a Markov communication network. In *Proceedings of 36th Conference on Decision and Control*, pages 4586–4591, Dec. 1997.
- [27] J. Nilsson, B. Bernhardsson, and B. Wittenmark. Stochastic analysis and control of real-time systems with random time delays. *Automatica*, 34(1):57–64, Jan. 1998.
- [28] M. N. Ovguztörel. *Time-lag control systems*. Academic Press, 1966.
- [29] N.-C. Tsai and A. Ray. Stochastic optimal control under randomly varying distributed delays. *International Journal of Control*, 68(5):1179–1202, Nov. 1997.
- [30] G. C. Walsh, O. Beldiman, and L. Bushnell. Asymptotic behavior of networked control systems. In *Proceedings of the Conference on Control Applications*, pages 1448–1453, Aug. 1999.

- [31] G. C. Walsh, H. Ye, and L. Bushnell. Stability analysis of networked control systems. In *Proceedings of the American Control Conference*, pages 2876–2889, June 1999.
- [32] B. Wittenmark, J. Nilsson, and M. Torngren. Timing problems in real-time control systems. In *Proceedings of American Control Conference*, pages 2000–2004, 1995.