

# Simple Delay-Based Implementation of Continuous-Time Controllers

Javad Lavaei, Somayeh Sojoudi and Richard M. Murray

**Abstract**—The objective of this work is to study the benefits that delay can provide in simplifying the control process of large-scale systems, motivated by the availability of different types of delays in man-made and biological systems. We show that a continuous-time linear time-invariant (LTI) controller can be approximated by a simple controller that mainly uses delay blocks instead of integrators. More specifically, three methods are proposed to approximate a pre-designed stabilizing LTI controller arbitrarily precisely by a simple delay-based controller composed of delay blocks, a few integrators and possibly a unity feedback. Different problems associated with the developed approximation procedures, such as finding the optimal number of delay blocks or studying the robustness of the designed controller with respect to delay values, are then addressed.

## I. INTRODUCTION

The field of control systems has seen a remarkable progress in different sub-areas such as robust control, adaptive control, cooperative control, system identification, optimal control, etc [1], [2], [3], [4], [5]. This has made it possible to engineer high performance controllers for real-world systems. However, the complex structure of such controllers is often an obstruction to their implementation in practice. It is, therefore, potentially useful to impose a simplicity constraint on the structure of the controller being designed for a large-scale system. This problem has not yet attracted much attention in the literature, and there are only a few works aiming at designing low-complex controllers. For example, Brockett [6] tackles a similar problem by optimizing a performance index that accounts for the complexity of the controller.

On the other hand, many theories have been developed for the analysis and synthesis of time-delay control systems due to the ubiquity of communication, computation or propagation delays in both man-made and biological systems [7], [8], [9]. The book [10] exemplifies the presence of delay in biology, chemistry, economics, mechanics, physics, physiology, and engineering sciences. Most of the existing controller design methods for time-delay systems regard delay as a nuisance and design a controller for the undelayed model of system in such a way that it is sufficiently robust to the underlying delay. Nevertheless, it is known that the voluntary introduction of delay in the control of an undelayed system could benefit the control process. For instance, delay can be used to create a limit cycle for nonlinear systems [11], to perform deadbeat tracking for continuous-time systems [12], or to stabilize oscillatory systems [13], [14].

The authors are with the Department of Control and Dynamical Systems, California Institute of Technology, Pasadena, USA (emails: lavaei@cds.caltech.edu; sojoudi@cds.caltech.edu; murray@cds.caltech.edu).

Given a continuous-time linear time-invariant (LTI) controller, the primary objective of this paper is to show that the controller can be approximated arbitrarily precisely by a simple delay-based controller. This controller is composed of some delay blocks, a few integrators and possibly a unity feedback (if the controller is stable and single-input single-output, the number of integrators is at most two). In other words, every high-order LTI controller has a simple delay-based implementation, which uses delay blocks rather than several integrators. Delay blocks, known also as delay lines, are intended to delay their incoming signal by a certain time period and exist in many different fields. For example, transmission lines in electronics and communications, cavity delay lines or trombone delay lines in optics, and neurons in neurobiology play the role of delay lines [15], [16]. Gene regulatory networks are another source of delay in biology, which can be defined as a set of interactions between the genes of a living organism [17]. Time delay appears in genetic networks due to transcription, translation, and translocation processes [14], [18]. The study of genetic networks has led to artificial networks such as VLSI neurochips [19], [20], [21]. For the applications where a delay line is already available or can be made much easier than an integrator, it may be preferable to design a controller based on delay blocks rather than integrators.

The rest of the paper is organized as follows. The problem is formulated in Section II. An illustrative example is then provided in Section III to demonstrate how a high-order controller can be approximated by a few delay blocks. Different delay-based controller design methods are proposed in Section IV. Some practical issues associated with the aforementioned design methods are then discussed in Section V. The results are further clarified in a numerical example in Section VI. Finally, some concluding remarks are given in Section VII.

## II. PROBLEM FORMULATION

Consider a continuous-time linear time-invariant (LTI) plant with the state-space representation

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t),\end{aligned}\tag{1}$$

where  $x(t) \in \mathbb{R}^n$ ,  $u(t) \in \mathbb{R}^m$  and  $y(t) \in \mathbb{R}^r$  denote the state, input and output of the system, respectively. Let  $P(s)$  denote the transfer function of the plant. Assume that a controller  $G(s)$  must be designed for the system in order for its behavior to satisfy certain specifications. It is preferred in practice that  $G(s)$  has the least possible complexity. The

simplest structure that one can think of for  $G(s)$  is likely a static output-feedback controller  $G(s) = L \in \mathbb{R}^{n \times m}$ , i.e.,

$$u(t) = Ly(t). \quad (2)$$

However, it is well-known that all LTI systems are not stabilizable via static output feedbacks. A more complex, but still simple, type of controller is as follows:

$$u(t) = \sum_{i=1}^k \alpha_i y(t - \tau_i), \quad (3)$$

where  $\alpha_1, \alpha_2, \dots, \alpha_k$  are constant gains and  $\tau_1, \tau_2, \dots, \tau_k$  are some nonnegative delays. The above controller is motivated by biological systems, as discussed in the introduction. Note that this controller can be expressed in the Laplace domain as

$$\sum_{i=1}^k \alpha_i e^{-\tau_i s}. \quad (4)$$

Since the decision problem of whether there exists a stabilizing controller of the form (2) is NP-hard, it is expected that a direct design of a controller of the type (3) is cumbersome. The present work aims to develop an indirect method for designing a controller in the form of (3). To this end, the first phase is to contrive a controller  $G(s)$  (not necessarily in the required form) using conventional techniques to satisfy the design objectives. We write a state-space realization of  $G(s)$  as

$$\begin{aligned} \dot{x}_c(t) &= A_c x_c(t) + B_c y(t), \\ u(t) &= C_c x_c(t) + D_c y(t), \end{aligned} \quad (5)$$

where  $x_c(t) \in \mathbb{R}^{n_c}$  represents the state of the controller. The primary goal of the paper is to approximate the given controller  $G(s)$  by a simple delay-based controller  $\hat{G}(s)$  of the form (4) so that the approximation error is less than any prescribed tolerance. Since it may turn out that a proper approximating controller  $\hat{G}(s)$  either does not exist or exploits many delays, another objective of the paper is to characterize other variants of the type (4) that still have easy implementation and can approximate every stabilizing controller.

### III. ILLUSTRATIVE EXAMPLE

The objective of this section is to illustrate how a high-order rational controller  $G(s)$  can be approximated by a simple delay-based controller. To this end, consider the admittedly artificial controller  $G(s) = 1 - \frac{(s+0.9)^{80}}{(s+1)^{80}}$ . This wideband low-pass filter has the property that it cannot be approximated by a low-order LTI controller due to its repeated poles. To illustrate this fact, consider a 32<sup>th</sup> order LTI controller  $\hat{G}(s)$  that is obtained from  $G(s)$  using the balanced model-reduction method. The Bode plots of  $G(s)$  and  $\hat{G}(s)$  are depicted in Figure 3, which substantiate that a good LTI approximation of  $G(s)$  needs more than 32 integrators.

Alternatively, let  $g(t)$  be the Laplace inverse of the controller  $G(s)$  and approximate the signal  $g(t)$  by a piecewise linear function  $\hat{g}(t)$ . A candidate for the approximating

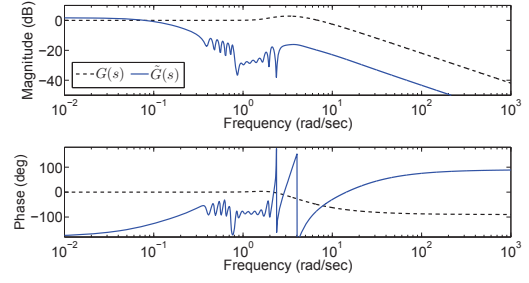


Fig. 1. The Bode plots of the controllers  $G(s) = 1 - \frac{(s+0.9)^{80}}{(s+1)^{80}}$  (dashed) and  $\hat{G}(s)$  (solid).

function  $\hat{g}(t)$  is shown in Figure 2. This piecewise linear function has 10 knots given by the vector  $\tau$  as follows:

$$\tau = \begin{bmatrix} 0 & 0.1 & 0.2 & 0.3 & 0.4 & 0.5 & 0.81 & 1.21 & 1.96 & 2.72 \end{bmatrix}. \quad (6)$$

The values of  $\hat{g}(t)$  at the breakpoints are

$$\begin{aligned} \hat{g}(\tau_1) &= 7.901, & \hat{g}(\tau_2) &= 4.631, & \hat{g}(\tau_3) &= 2.505, \\ \hat{g}(\tau_4) &= 1.121, & \hat{g}(\tau_5) &= 0.312, & \hat{g}(\tau_6) &= -0.264, \\ \hat{g}(\tau_7) &= -0.551, & \hat{g}(\tau_8) &= -0.14, & \hat{g}(\tau_9) &= 0.1, \\ \hat{g}(\tau_{10}) &= 0.0163, \end{aligned} \quad (7)$$

where  $\tau_i$  denotes the  $i^{\text{th}}$  element of  $\tau$  for every  $i \in \{1, 2, \dots, 10\}$ . Define  $\hat{G}(s)$  to be the Laplace transform of  $\hat{g}(t)$ , which can be obtained as

$$\begin{aligned} \hat{G}(s) &:= \sum_{i=1}^9 \left( \frac{w_i}{s^2} + \frac{\hat{g}(\tau_i)}{s} \right) e^{-\tau_i s} \\ &+ \sum_{i=1}^9 \left( -\frac{w_i}{s^2} - \frac{\hat{g}(\tau_{i+1})}{s} \right) e^{-\tau_{i+1} s}, \end{aligned} \quad (8)$$

where

$$w_i = \frac{\hat{g}(\tau_{i+1}) - \hat{g}(\tau_i)}{\tau_{i+1} - \tau_i}, \quad i = 1, 2, \dots, 9. \quad (9)$$

The implementation of  $\hat{G}(s)$  requires 2 integrators and 9 delay blocks. The Bode plots of  $G(s)$  and  $\hat{G}(s)$  are compared in Figure 2 to show how well  $\hat{G}(s)$  approximates  $G(s)$ . This example elucidates that a high-order controller  $G(s) = 1 - \frac{(s+0.9)^{80}}{(s+1)^{80}}$ , whose satisfactory LTI approximation needs at least 33 integrators, can be approximated very well by 2 integrators and 9 delay blocks. Note that the infinity norm of the approximation error  $G(s) - \hat{G}(s)$  is equal to 0.022. The present paper aims to develop a concrete theory for the general case.

### IV. DELAY-BASED CONTROLLER DESIGN

In this section, assume that  $G(s)$  is a single-input single-output (SISO) controller that is (asymptotically) stable. The results will be extended to the general case in the next section. Moreover, with no loss of generality we assume that  $G(s)$  is strictly proper, because the direct term  $D_c$  in the controller corresponds to a static feedback that can be added to the delay-based controller directly. Three different methods will be proposed in the sequel for designing  $\hat{G}(s)$ .

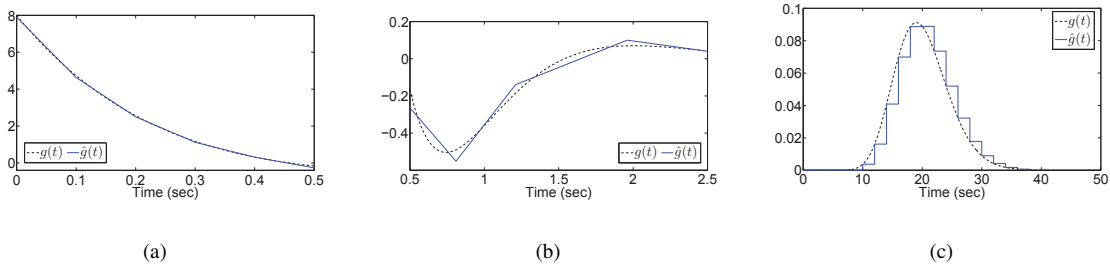


Fig. 2. (a): A piecewise linear approximation of the inverse Laplace of  $G(s) = 1 - \frac{(s+0.9)^{80}}{(s+1)^{80}}$  in the interval  $[0, 0.5]$ ; (b): a piecewise linear approximation of the inverse Laplace of  $G(s) = 1 - \frac{(s+0.9)^{80}}{(s+1)^{80}}$  in the interval  $[0.5, 2.5]$ ; (c): an approximation of the inverse Laplace of the controller  $G(s) = \frac{1}{(s+1)^{20}}$  by a step-like function (needed for Method 2).

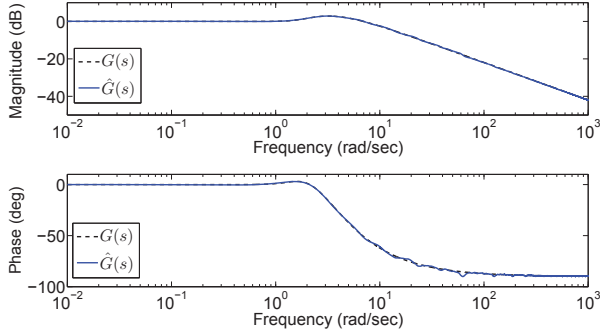


Fig. 3. The Bode plots of the controllers  $G(s) = 1 - \frac{(s+0.9)^{80}}{(s+1)^{80}}$  and  $\hat{G}(s)$ .

#### A. Method 1

Let  $\hat{G}(s)$  be a function in the Laplace domain which is analytic on the open left-half  $s$ -plane. It follows from the maximum modulus theorem and the stability of the controller  $G(s)$  that

$$\max_{s: \operatorname{Re}\{s\} \geq 0} |G(s) - \hat{G}(s)| = \max_{\omega \in \mathbb{R}} |G(j\omega) - \hat{G}(j\omega)|, \quad (10)$$

where the operator  $|\cdot|$  returns the absolute value of a complex number. Therefore, the maximum difference between the controllers  $G(s)$  and  $\hat{G}(s)$  can be evaluated by only the restriction to the  $j\omega$  axis. On the other hand, the definition of the Fourier transform yields

$$G(j\omega) = \int_0^{\infty} g(t)e^{-j\omega t} dt. \quad (11)$$

Since each term  $e^{-j\omega t}$  has the form of a delay component, the above integral implies that  $G(s)$  can be regarded as a controller with static distributed delays. In contrast, the controller  $\hat{G}(s)$  to be designed should be in the form of static lumped delays. Hence, the question of interest would be how to approximate the distributed delays with lumped delays. To answer this question, one can take advantage of any integral approximation method, such as the midpoint method. More precisely, consider some nonnegative numbers  $\tau_1 < \tau_2 < \dots < \tau_k$  and define  $\hat{G}(s)$  as

$$\hat{G}(s) = \sum_{i=1}^{k-1} g(\tau_i)(\tau_{i+1} - \tau_i)e^{-\tau_i s} \quad (12)$$

or

$$\hat{G}(s) = \sum_{i=1}^{k-1} g(\bar{\tau}_i)(\tau_{i+1} - \tau_i)e^{-\bar{\tau}_i s}, \quad (13)$$

where

$$\bar{\tau}_i = \frac{\tau_i + \tau_{i+1}}{2}, \quad i = 1, 2, \dots, k-1. \quad (14)$$

The main focus of this subsection will be on the approximating controller (13) as the other one can be analyzed similarly.

*Theorem 1:* The approximation error  $G(j\omega) - \hat{G}(j\omega)$  satisfies the following inequality for every  $\omega \in \mathbb{R}$ :

$$\begin{aligned} |G(j\omega) - \hat{G}(j\omega)| &\leq \sqrt{2} \int_0^{\tau_1} |g(t)| dt + \sqrt{2} \int_{\tau_k}^{\infty} |g(t)| dt \\ &+ \sqrt{2} \sum_{i=1}^{k-1} \frac{(\tau_{i+1} - \tau_i)^3}{24} \\ &\times \max_{\tau \in [\tau_i, \tau_{i+1}]} \left\{ \left| \frac{\partial^2 (g(\tau) \cos(\omega\tau))}{\partial^2 \tau} \right|, \left| \frac{\partial^2 (g(\tau) \sin(\omega\tau))}{\partial^2 \tau} \right| \right\}. \end{aligned} \quad (15)$$

*Proof:* The proof is a direct consequence of the midpoint error formula. The details are omitted for brevity (see the proof of Theorem 2 for a similar argument). ■

Notice that the right side of the inequality given in Theorem 1 can become large for sufficiently large values of  $\omega$  due to the existence of the second derivative of the term  $\cos(\omega\tau)$ . This fact can also be justified from another point of view: if  $\tau_1, \tau_2, \dots, \tau_k$  are integer multiples of some real number, then  $\hat{G}(j\omega)$  will be a periodic number, otherwise it would be almost periodic with a potentially large period. In other words,  $\hat{G}(j\omega)$  cannot approximate  $G(j\omega)$  for high frequencies. However, in the case when the plant for which  $G(s)$  is designed acts as a low-pass filter with an appropriate stop frequency, it is not critical that  $G(j\omega)$  and  $\hat{G}(j\omega)$  are quite different for high frequencies. On the other hand, it can be inferred from the inequality (15) that the numbers  $\tau_1, \tau_2, \dots, \tau_k$  (in addition to  $k$ ) can be chosen in such a way that  $\hat{G}(j\omega)$  approximates  $G(j\omega)$  arbitrarily precisely over any desired range of frequencies. This point will be further discussed in the next section

## B. Method 2

Since  $\hat{G}(s)$  proposed by Method 1 has an undesirable behavior in high frequencies, a more sophisticated approach can be used to resolve this issue. The basic idea behind the new method is to approximate the impulse response of the controller  $G(s)$  by a step-like function. Figure 2c illustrates this idea for the particular controller  $G(s) = \frac{1}{(s+1)^{20}}$ . Given a monotonically increasing sequence of nonnegative numbers  $\tau_1, \tau_2, \dots, \tau_k$ , the function  $g(t)$  can be approximated by a step-like function such as

$$\hat{g}(t) = \begin{cases} g(\tau_i) & t \in [\tau_i, \tau_{i+1}], \quad i = 1, 2, \dots, k-1 \\ 0 & t < \tau_1 \quad \text{or} \quad t > \tau_k \end{cases} \quad (16)$$

or

$$\hat{g}(t) = \begin{cases} g(\bar{\tau}_i) & t \in [\tau_i, \tau_{i+1}], \quad i = 1, 2, \dots, k-1 \\ 0 & t < \tau_1 \quad \text{or} \quad t > \tau_k \end{cases} \quad (17)$$

where  $\bar{\tau}_i = \frac{\tau_i + \tau_{i+1}}{2}$ . This subsection will focus on the later  $\hat{g}(t)$  as the former one can be analyzed similarly. The transfer function corresponding to the function  $\hat{g}(t)$  given in (17) is as follows:

$$\hat{G}(s) = \frac{1}{s} \sum_{i=1}^k \alpha_i e^{-\tau_i s}, \quad (18)$$

where

$$\begin{aligned} \alpha_1 &:= g(\bar{\tau}_1), \\ \alpha_i &:= g(\bar{\tau}_i) - g(\bar{\tau}_{i-1}), \quad i = 2, 3, \dots, k-1, \\ \alpha_k &:= -g(\bar{\tau}_{k-1}). \end{aligned} \quad (19)$$

Note that  $\hat{G}(s)$  can be implemented using  $k$  static delay terms and an integrator.

*Theorem 2:* The approximation error  $G(j\omega) - \hat{G}(j\omega)$  satisfies the following inequality for every  $\omega \in \Re$ :

$$\begin{aligned} |G(j\omega) - \hat{G}(j\omega)| &\leq \sum_{i=1}^{k-1} \max_{\tau \in [\tau_i, \tau_{i+1}]} |g''(\tau)| \frac{\sqrt{2}(\tau_{i+1} - \tau_i)^3}{24} \\ &+ \sqrt{2} \sum_{i=1}^{k-1} |g'(\bar{\tau}_i)| \max \{ |\operatorname{Re}\{H(i, \omega)\}|, |\operatorname{Im}\{H(i, \omega)\}| \} \\ &+ \sqrt{2} \int_0^{\tau_1} |g(t)| dt + \sqrt{2} \int_{\tau_k}^{\infty} |g(t)| dt, \end{aligned} \quad (20)$$

where

$$H(i, \omega) := \int_{\tau_i}^{\tau_{i+1}} (t - \bar{\tau}_i) e^{-j\omega t} dt, \quad i = 1, 2, \dots, k-1, \quad \omega \in \Re. \quad (21)$$

*Proof:* One can use the Taylor series with the Lagrange form of the remainder to obtain that for every  $i \in \{1, 2, \dots, k-1\}$  and  $t \in [\tau_i, \tau_{i+1}]$ , there exists a function  $\gamma(t) \in [\tau_i, \tau_{i+1}]$  such that

$$g(t) = g(\bar{\tau}_i) + g'(\bar{\tau}_i)(t - \bar{\tau}_i) + \frac{g''(\gamma(t))}{2}(t - \bar{\tau}_i)^2. \quad (22)$$

Therefore

$$\begin{aligned} |\operatorname{Re}\{G(j\omega) - \hat{G}(j\omega)\}| &= \left| \int_0^{\infty} (g(t) - \hat{g}(t)) \cos(\omega t) dt \right| \\ &\leq \sum_{i=1}^{k-1} \int_{\tau_i}^{\tau_{i+1}} \left| \frac{1}{2} g''(\gamma(t)) (t - \bar{\tau}_i)^2 \cos(\omega t) dt \right| \\ &\quad + \sum_{i=1}^{k-1} \left| \int_{\tau_i}^{\tau_{i+1}} g'(\bar{\tau}_i) (t - \bar{\tau}_i) \cos(\omega t) dt \right| \\ &\quad + \left| \int_0^{\tau_1} g(t) \cos(\omega t) dt \right| + \left| \int_{\tau_k}^{\infty} g(t) \cos(\omega t) dt \right| \\ &\leq \sum_{i=1}^{k-1} \max_{\tau \in [\tau_i, \tau_{i+1}]} |g''(\tau)| \frac{(\tau_{i+1} - \tau_i)^3}{24} + \int_0^{\tau_1} |g(t)| dt \\ &\quad + \sum_{i=1}^{k-1} |g'(\bar{\tau}_i)| |\operatorname{Re}\{H(i, \omega)\}| + \int_{\tau_k}^{\infty} |g(t)| dt. \end{aligned} \quad (23)$$

An inequality similar to (23) can be written for  $|\operatorname{Im}\{G(j\omega) - \hat{G}(j\omega)\}|$  which together with (23) proves this theorem. ■

It is noteworthy that  $H(i, \omega)$  introduced in the above theorem has the property that it is equal to zero at  $\omega = 0$  and also tends to zero as  $\omega$  goes to infinity. The inequality provided in Theorem 2 implies that one can design the delays  $\tau_1, \tau_2, \dots, \tau_k$  (besides  $k$ ) so that the approximation error is less than any given number at every frequency (note that since  $G(s)$  is strictly proper,  $g(t)$  attenuates to zero as  $t$  increases).

## C. Method 3

Although Method 2 eliminates the fluctuation effect created by Method 1 at high frequencies, we propose a third method that normally needs fewer delays than Method 2 at the cost of deploying one more integrator. Let the function  $g(t)$  be approximated by a piecewise linear function  $\hat{g}(t)$  with the breakpoints  $\tau_1, \tau_2, \dots, \tau_k$  (listed in an ascending order), i.e.,

$$\hat{g}(t) = \begin{cases} \frac{g(\tau_{i+1}) - g(\tau_i)}{\tau_{i+1} - \tau_i} (t - \tau_i) + g(\tau_i) & t \in [\tau_i, \tau_{i+1}] \\ 0 & t < \tau_1 \quad \text{or} \quad t > \tau_k \end{cases} \quad (24)$$

for all  $i \in \{1, 2, \dots, k-1\}$ . As before, the function  $\hat{G}(s)$  can be obtained as follows:

$$\hat{G}(s) = \sum_{i=1}^k \beta_i(s) e^{-\tau_i s}, \quad (25)$$

where

$$\begin{aligned} \beta_1(s) &:= \frac{w_1}{s^2} + \frac{g(\tau_1)}{s}, \\ \beta_i(s) &:= \frac{w_i}{s^2} - \frac{w_{i-1}}{s^2}, \quad i = 2, \dots, k-1, \\ \beta_k(s) &:= -\frac{w_{k-1}}{s^2} - \frac{g(\tau_k)}{s}, \end{aligned} \quad (26)$$

and

$$w_i := \frac{g(\tau_{i+1}) - g(\tau_i)}{\tau_{i+1} - \tau_i}, \quad i = 1, 2, \dots, k-1. \quad (27)$$

Note that the approximating controller  $\hat{G}(s)$  introduced above can be implemented using  $k$  static delay terms and two integrators. It is desired to measure the estimation error  $\|G(j\omega) - \hat{G}(j\omega)\|_\infty$ , where  $\|\cdot\|_\infty$  denotes the infinity norm.

*Theorem 3:* The approximation error  $\|G(j\omega) - \hat{G}(j\omega)\|_\infty$  satisfies the following inequality:

$$\begin{aligned} \|G(s) - \hat{G}(s)\|_\infty &\leq \sqrt{2} \int_0^{\tau_1} |g(t)| dt + \sqrt{2} \int_{\tau_k}^\infty |g(t)| dt \\ &\quad + \sum_{i=1}^{k-1} \max_{\tau \in [\tau_i, \tau_{i+1}]} |g''(\tau)| \frac{\sqrt{2}(\tau_{i+1} - \tau_i)^3}{12}. \end{aligned} \quad (28)$$

*Proof:* Given an index  $i \in \{1, 2, \dots, k-1\}$ , it follows from the polynomial interpolation formula that

$$g(t) - \hat{g}(t) = \frac{1}{2} g''(\eta(t))(t - \tau_{i+1})(t - \tau_i), \quad t \in [\tau_i, \tau_{i+1}], \quad (29)$$

where  $\eta(t)$  is some time instant in the interval  $[\tau_i, \tau_{i+1}]$ . Therefore, one can write:

$$\begin{aligned} |\operatorname{Re}\{G(j\omega) - \hat{G}(j\omega)\}| &= \left| \int_0^\infty (g(t) - \hat{g}(t)) \cos(\omega t) dt \right| \\ &\leq \sum_{i=1}^{k-1} \int_{\tau_i}^{\tau_{i+1}} \left| \frac{1}{2} g''(\eta(t))(t - \tau_{i+1})(t - \tau_i) \cos(\omega t) dt \right| \\ &\quad + \left| \int_0^{\tau_1} g(t) \cos(\omega t) dt \right| + \left| \int_{\tau_k}^\infty g(t) \cos(\omega t) dt \right| \\ &\leq \frac{1}{2} \sum_{i=1}^{k-1} \max_{\tau \in [\tau_i, \tau_{i+1}]} |g''(\tau)| \int_{\tau_i}^{\tau_{i+1}} (\tau_{i+1} - t)(t - \tau_i) dt \\ &\quad + \int_0^{\tau_1} |g(t)| dt + \int_{\tau_k}^\infty |g(t)| dt \\ &= \sum_{i=1}^{k-1} \max_{\tau \in [\tau_i, \tau_{i+1}]} |g''(\tau)| \frac{(\tau_{i+1} - \tau_i)^3}{12} \\ &\quad + \int_0^{\tau_1} |g(t)| dt + \int_{\tau_k}^\infty |g(t)| dt. \end{aligned} \quad (30)$$

A similar inequality can be obtained for  $|\operatorname{Im}\{G(j\omega) - \hat{G}(j\omega)\}|$  whose combination with the above relation completes the proof. ■

It follows from the inequality provided in Theorem 3 that the delays  $\tau_1, \tau_2, \dots, \tau_k$  (together with  $k$  itself) can be contrived in such a way that the approximation error in infinity norm does not exceed a prescribed tolerance.

*Remark:* This subsection approximates the time-domain signal  $g(t)$  by a piecewise linear function, but assumes that the knots of the approximating signal lie on the curve of the function  $g(t)$ . This assumption has been made for simplicity and it is not required in general to choose the corners of the approximating function  $\hat{g}(t)$  on the signal  $g(t)$ . This idea is illustrated in Figure 2b. The theory developed above can be easily extended to the general case.

## V. PRACTICAL ISSUES

The goal of this section is to address some practical issues associated with the aforementioned delay-based controller designs.

### A. Optimal Choice of Delays

Three methods have been proposed in the preceding section for approximating a given high-order controller by a simple delay-based controller. In terms of the given delays, upper bounds on the infinity norm of the error were proposed for each method. However, a fundamental question in the first place would be how to find an optimal set of delays  $\{\tau_1, \tau_2, \dots, \tau_k\}$ . The provided upper bounds can definitely help pick appropriate delays. Alternatively, one can take advantage of the existing methods in the literature for this purpose. More specifically, notice that Methods 2 and 3 rely on the approximation of a function  $g(t)$  by a step-like or a piecewise linear function  $\hat{g}(t)$ . In terms of a given function norm  $\|\cdot\|$  (namely 1 or  $\infty$  norm), there are systematic methods in the literature for finding a function  $\hat{g}(t)$  with the minimum number of breakpoints such that the error  $\|g(t) - \hat{g}(t)\|$  is less than a prescribed positive tolerance  $\varepsilon$ . The most straightforward way for this purpose is to discretize the signal  $g(t)$  in order to make the underlying problem finite dimensional. One of these methods will be outlined in the sequel for piecewise linear approximation with respect to the  $\infty$ -norm. Let  $T$  denote a positive time such that  $|g(t)| \leq \varepsilon$  for all  $t \geq T$ . Discretize the signal  $g(t)$  over the interval  $[0, T]$  with a sampling period  $d$  to obtain a discretized signal  $g_d(t)$ . The goal is to find a discrete piecewise linear signal  $\hat{g}_d(t)$  such that  $\|g_d(t) - \hat{g}_d(t)\|_\infty \leq \varepsilon$ . Four problems can be defined as follows for a given positive real  $\varepsilon$  and a natural number  $k$ :

- *P1:* Find a piecewise linear function  $\hat{g}_d(t)$  with the minimum number of breakpoints (corners) such that  $\|g_d(t) - \hat{g}_d(t)\|_\infty \leq \varepsilon$ .
- *P2:* Find a piecewise linear function  $\hat{g}_d(t)$  with the minimum number of breakpoints such that  $\hat{g}_d(t)$  overlaps on  $g_d(t)$  at its corners (when regarded as a graph) and that  $\|g_d(t) - \hat{g}_d(t)\|_\infty \leq \varepsilon$ .
- *P3:* Find a piecewise linear function  $\hat{g}(t)$  with at most  $k$  breakpoints such that  $\|g_d(t) - \hat{g}_d(t)\|_\infty$  is minimum.
- *P4:* Find a piecewise linear function  $\hat{g}(t)$  with at most  $k$  breakpoints such that  $\hat{g}_d(t)$  overlaps on  $g_d(t)$  at its corners and that  $\|g_d(t) - \hat{g}_d(t)\|_\infty$  is minimum.

Note that the delays being found will all be multiples of the sampling time  $d$ . Let  $N$  denote the number of discrete points of the function  $g_d(t)$ . It is shown in [22] that there are deterministic algorithms for solving P1, P2, P3 and P4 whose complexities are  $O(N)$ ,  $O(N^2)$ ,  $O(N^2 \log N)$  and  $O(N^2 \log N)$ , respectively. This implies that P1 seems to be the easiest problem to solve, which is indeed the most desirable one for the purpose of the present paper. However, since the algorithm for solving P1 is somewhat involved, the algorithm for P2 will be briefly explained next which provides insight into how the algorithms for the other problems P1, P3 and P4 work. Represent the points of the discrete signal  $g_d(t)$  with  $p_1, p_2, \dots, p_N$ . Construct a directed graph  $\mathcal{G}$  with  $N$  vertices as follows. For every  $i, j \in \{1, 2, \dots, N\}$  and  $i < j$ , connect vertex  $i$  to vertex  $j$  via a directed edge if the infinity norm between the line connecting  $p_i$  to  $p_j$  and all

points  $p_i, p_{i+1}, \dots, p_j$  is less than or equal to  $\varepsilon$ . This graph can be built in  $O(N^2)$ . Now, every path in this graph from vertex 1 to vertex  $N$  is a candidate for  $\hat{g}(t)$ . An optimal  $\hat{g}(t)$  corresponds to the shortest path from vertex 1 to vertex  $N$ , which can be found in  $O(N^2)$  due to the graph being acyclic.

### B. Strong Stabilization

The current paper assumes that a stable controller  $G(s)$  is available for a plant  $P(s)$ , which needs to be approximated by a simple delay-based controller. Nonetheless, such a stable controller may not exist in general. The problem of designing a stable controller for a given system is referred to as *strong stabilization* and has a long history in control. The main motivation for the strong stabilization problem is that a stable controller for a system provides more robustness with respect to specific faults in the control system. It is well-known that  $P(s)$  is strongly stabilizable if and only if the system has an even number of unstable real poles between every two consecutive unstable real blocking zeros of  $P(s)$  (including the zeros at infinity). Note that a complex number  $z$  is said to be a blocking zero of the system if  $P(z)$  is equal to zero. There are many methods in the literature for designing a stable controller for a strongly stabilizable system. These methods normally design a high-order stable controller, say with an order twice the order of the system [23], [24]. Hence, the current work can be used to simplify the implementation of such a high-order stable controller.

### C. Unstable Controllers

Assume that a given controller  $G(s)$  is unstable. The next question would be how to implement it in practice using delay terms with the aim of simplifying the control structure. The easiest approach is to decompose  $G(s)$  as the cascade of stable and unstable sub-controllers and then simplify only the stable part. This technique is inefficient in the case when most of the poles of the controller  $G(s)$  are unstable. Thus, a more advanced technique will be introduced here. Since  $G(s)$  stabilizes the plant  $P(s)$ , the controller itself must be stabilizable. As a result, the pair  $(A_c, B_c)$  is stabilizable and, therefore, there exists a matrix gain  $F \in \mathbb{R}^{1 \times n_c}$  for which  $A_c - B_c F$  is Hurwitz. Define  $\tilde{y}(t) := Fx_c(t)$  and  $v(t) := y(t) + \tilde{y}(t)$ . The controller  $G(s)$  is equivalent to the feedback controller given in Figure 4 whose backward path is a unity feedback and whose forward path is a sub-controller  $\tilde{G}(s)$  with the control law

$$\begin{aligned} \dot{x}_c(t) &= (A_c - B_c F)x_c(t) + B_c v(t), \\ u(t) &= C_c x_c(t), \\ \tilde{y}(t) &= Fx_c(t). \end{aligned} \quad (31)$$

Notice that  $\tilde{G}(s)$  is stable with a single input  $v(t)$  and two outputs  $u(t)$  and  $\tilde{y}(t)$ . Now, each of the transfer functions from  $v(t)$  to  $u(t)$  and  $\tilde{y}(t)$  can be approximated by a simple delay-based controller. This makes the controller  $\tilde{G}(s)$  be approximated by a controller  $\hat{G}(s)$  consisting of delay blocks and four integrators (due to the existence of two SISO transfer functions). As a result, every stabilizing unstable

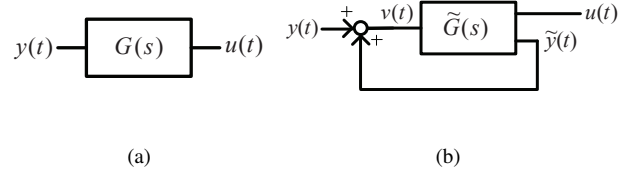


Fig. 4. Figures (a) and (b) show an unstable controller and its equivalent feedback representation, respectively, where the forward path controller  $\tilde{G}(s)$  is stable.

controller  $G(s)$  can be approximated by a feedback controller with the unity feedback whose forward path is a delay-based controller.

### D. Distributed Control Systems

Assume that  $G(s)$  is a matrix corresponding to a multi-input multi-output (MIMO) system  $P(s)$ . To approximate  $G(s)$  by a simple delay-based controller, three approaches can be taken as follows:

- Each SISO element of  $G(s)$  is approximated by a SISO delay-based controller separately. The overall set of delays will be the union of the sets of delays obtained for different elements of  $G(s)$ . The drawback of this method is that this union may lead to a large set and the number of integrators could be unnecessarily high.
- Each of Methods 1, 2 and 3 stated earlier is valid for the MIMO case and the only difference is that  $g(t)$  should be regarded as a matrix, as opposed to a scalar. The disadvantage of this method is that all SISO entries (transfer functions) in  $G(s)$  will be approximated by the same set of delays, which may give rise to requiring many delays for a good approximation in case of a large matrix  $G(s)$ .
- The MIMO controller  $G(s)$  is regarded as a union of some multi-input single-output (MISO) sub-controllers. The methods proposed earlier can be adopted to approximate each of these MISO sub-controllers individually (the advantage of this method is to use fewer integrators).

It can be deduced from the above discussion that in the case when  $G(s)$  is a large matrix, a delay-based controller may need several delays and more than 4 integrators. Nevertheless, an important application of this work is in the distributed/decentralized control of an interconnected system. For such an application,  $G(s)$  is naturally partitioned into a number of blocks where each block represents the local controller of a control channel/agent. Then, disparate blocks of  $G(s)$  can have their own delay sets and integrators as they correspond to separate control agents. As a result, each agent need not use many (unnecessary) delays.

### E. Stability Issue

Recall that the approximating controller  $\hat{G}(s)$  obtained using Method 2 or Method 3 includes one or two integrators. Hence, it may be speculated that  $\hat{G}(s)$  has a pole at the origin, whereas  $G(s)$  has no pole in the closed right-half complex plane. However, it can be shown that  $\hat{G}(0)$  is finite

in both cases, as a pole-zero cancellation occurs. Since this cancellation cannot take place perfectly in practice, an extra pole at zero will be introduced using Methods 2 and 3. Although this new pole may not affect the stability of the closed-loop system, in the case when a stable approximating controller is sought, one can resolve the issue easily. To present the main idea, consider Method 2 which approximates  $g(t)$  by a step-like function, namely

$$\hat{g}(t) = \begin{cases} g(\tau_i) & t \in [\tau_i, \tau_{i+1}], \quad i = 1, 2, \dots, k-1 \\ 0 & t < \tau_1 \text{ or } t > \tau_k \end{cases} \quad (32)$$

Let  $\hat{g}(t)$  be modified as below:

$$\hat{g}(t) = \begin{cases} g(\tau_i)e^{-\alpha(t-\tau_i)} & t \in [\tau_i, \tau_{i+1}], \quad i = 1, 2, \dots, k-1 \\ 0 & t < \tau_1 \text{ or } t > \tau_k \end{cases} \quad (33)$$

where  $\alpha$  is a (small) positive number. As before, define  $\hat{G}(s)$  to be the Laplace transform of  $\hat{g}(t)$ . It is easy to show that  $\hat{G}(s)$  can be implemented using  $k$  delay terms along with the stable low-pass filter  $\frac{1}{s+\alpha}$  as opposed to an integrator.

#### F. Stability and Robustness

It was shown in the preceding section how to approximate a nominal controller  $G(s)$  by a delay-based controller with possibly a unity feedback (in the case of an unstable  $G(s)$ ). The resultant controller may not stabilize the plant  $P(s)$  due to the approximation error not being sufficiently small. Thus, a stability analysis is required to guarantee the closed-loop stability of the system. To this end, consider a general controller  $G(s)$  (which could be stable or unstable) which is approximated by a unity feedback, as depicted in Figure 4, with a delay-based sub-controller  $\hat{G}(s)$  in the forward path. Note that the case of a stable controller  $G(s)$  is a special case of this setting by letting  $F$  be zero. This subsection develops some results for the SISO case, which can be easily generalized to the MIMO case. Notice that  $\hat{G}(s)$  is an approximation of the sub-controller  $\tilde{G}(s)$ , and that the error between these two controllers can be best modeled by both additive and multiplicative terms. Therefore, let  $\Delta_1(j\omega) \in \mathbf{C}$  and  $\Delta_2(j\omega) \in \mathbf{C}^{2 \times 1}$  be matrix functions such that

$$\hat{G}(j\omega) = \tilde{G}(j\omega)(1 + \Delta_1(j\omega)) + \Delta_2(j\omega), \quad \forall \omega \in \mathfrak{R} \quad (34)$$

(where  $\mathbf{C}$  denotes the set of complex numbers). It can be shown that the closed-loop control system (with the approximating controller designed) is stable if

$$\left\| \frac{\tilde{P}\tilde{G}}{1 + \tilde{P}\tilde{G}} \right\|_{\infty} |\Delta_1(j\omega)| + \left\| \frac{\tilde{P}}{1 + \tilde{P}\tilde{G}} \right\|_{\infty} |\Delta_2(j\omega)| < 1 \quad (35)$$

for all  $\omega \in \mathfrak{R}$ , where  $\tilde{P}(s) = [P(s) \ 1]$ . The above inequality provides a means to check the stability of the closed-loop system for a designed  $\hat{G}(s)$ , or even to design  $\hat{G}(s)$  by first finding the permissible uncertainties  $\Delta_1(j\omega), \Delta_2(j\omega)$  and then obtaining delays so that the above inequality is satisfied.

A question arises: how sensitive is the designed controller to the delay values? This question is of a great importance

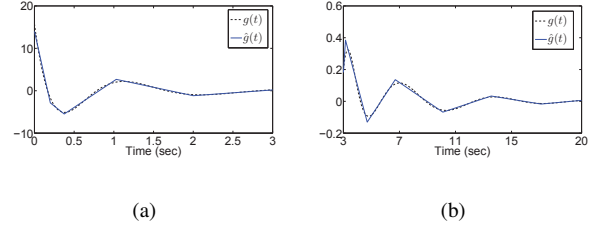


Fig. 5. (a) The time-domain signals  $g(t)$  and  $\hat{g}(t)$  in the interval  $[0, 3]$ ; (b) the time-domain signals  $g(t)$  and  $\hat{g}(t)$  in the interval  $[3, 20]$ .

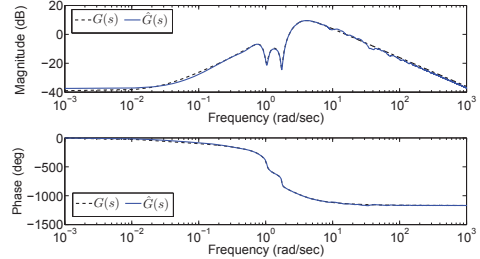


Fig. 6. The Bode plots of the controllers  $G(s)$  and  $\hat{G}(s)$  for the example given in Section VI.

due to the fact that it may not be possible to have a perfect delay block in practice. To investigate this issue, consider Method 1. Let the delay values  $\tau_1 + \delta\tau_1, \tau_2 + \delta\tau_2, \dots, \tau_k + \delta\tau_k$  be used instead of the nominal values  $\tau_1, \tau_2, \dots, \tau_k$ . This means that the approximating controller

$$\hat{G}(s) = \sum_{i=1}^{k-1} g(\tau_i)(\tau_{i+1} - \tau_i)e^{-\tau_i s} \quad (36)$$

will be perturbed as follows:

$$\hat{G}(s) + \Delta\hat{G}(s) = \sum_{i=1}^{k-1} g(\tau_i)(\tau_{i+1} - \tau_i)e^{-(\tau_i + \delta\tau_i)s}. \quad (37)$$

It is easy to observe that  $\Delta\hat{G}(j\omega)$  is negligible for small values of  $\omega$ ; in particular,  $\Delta\hat{G}(0) = 0$ . However,  $\Delta\hat{G}(j\omega)$  may become large for a high frequency  $\omega$ . In other words, a perturbation in the delays would affect the transfer function of the controller only at high frequencies, which is not a big issue if the system for which the controller is designed is strictly proper (due to its filtering property).

#### VI. NUMERICAL EXAMPLE

Consider the 8<sup>th</sup> order unstable plant  $P(s) = \frac{P_1(s)}{P_2(s)}$ , where

$$\begin{aligned} P_1(s) &:= 0.0064s^5 + 0.0024s^4 + 0.071s^3 \\ &\quad + s^2 + 0.1045s + 1, \\ P_2(s) &:= s^8 + 0.161s^7 + 6s^6 + 0.582s^5 + 9.984s^4 \\ &\quad + 0.407s^3 + 3.9822s^2 + 0.08s + 0.08. \end{aligned} \quad (38)$$

This system has been obtained from a benchmark example for the strong stabilization problem by adding the term  $0.08s + 0.08$  to the denominator (see [23], [24] and the references therein). One can design an LQG controller for this system with the weighting matrices  $Q = I$  and  $R = 1$

(the noise covariance is assumed to be  $I$ ) to obtain a stable controller  $G(s) = \frac{G_1(s)}{G_2(s)}$ , where

$$\begin{aligned} G_1(s) &:= 15.76s^7 - 3.896s^6 + 60.68s^5 - 9.68s^4 \\ &\quad + 34.99s^3 - 2.064s^2 - 12.39s + 0.2986, \\ G_2(s) &:= s^8 + 8.684s^7 + 41.18s^6 + 115.3s^5 + 208.8s^4 \\ &\quad + 250.9s^3 + 197.9s^2 + 111.1s + 26.64. \end{aligned} \quad (39)$$

We use a variant of Method 3 to approximate  $G(s)$  by a simple delay-based controller consisting of a number of delay blocks and at most two integrators (see the remark given after Theorem 3). The impulse response of the controller  $G(s)$ , plotted in Figure 5, is an oscillatory signal. This makes it impossible to find a good piecewise linear approximation of this function with only a few breakpoints, because there are several dominant peaks in the signal  $g(t)$  that should be all chosen as breakpoints. Based on the peaks of the signal  $g(t)$ , a vector of breakpoints  $\tau$  was obtained as

$$\tau = \begin{bmatrix} 0 & 0.2 & 0.375 & 1.03 & 2 & 3.15 & 4.7 & 6.7 \\ 10.1 & 13.55 & 17.11 & 20 & & & & \end{bmatrix}. \quad (40)$$

The method proposed in Subsection V.A can be used to find the best piecewise linear approximation of  $g(t)$  with its knots given by the vector  $\tau$ . Note that the corners of the obtained approximating function  $\hat{g}(t)$  do not necessarily lie on the function  $g(t)$ . The corresponding signal  $\hat{g}(t)$  is plotted in Figure 5. The Bode plots of the controllers  $G(s)$  and  $\hat{G}(s)$  are compared in Figure 6, which illustrate that  $\hat{G}(s)$  is a good approximation of  $G(s)$ . Let  $\tilde{G}(s)$  denote a 6<sup>th</sup> order reduced model of  $G(s)$  obtained using the balanced model-reduction technique. To compare  $\hat{G}(s)$  with  $\tilde{G}(s)$ , notice that:

$$\begin{aligned} \max_{\omega \in [0,1]} |\hat{G}(j\omega) - G(j\omega)| &\simeq 0.03, \\ \max_{\omega \in [0,1]} |\tilde{G}(j\omega) - G(j\omega)| &\simeq 0.33. \end{aligned} \quad (41)$$

This implies that an LTI approximation of  $G(s)$  that performs as well as  $\hat{G}(s)$  requires at least 7 integrators, whereas  $\tilde{G}(s)$  can be implemented using 2 integrators and 11 delay blocks.

## VII. CONCLUSIONS

Motivated by biological systems, this paper studies the possibility of synthesizing controllers whose implementation mainly requires delay blocks, as opposed to integrators. This problem is particularly important for large-scale systems whose control using conventional techniques needs many integrators. It is shown that every stabilizing continuous-time linear time-invariant controller can be approximated arbitrarily precisely by a simple delay-based controller comprising delay blocks and a few integrators. In particular, if the controller is both stable and single-input single-output, the number of integrators is at most equal to two. Finding the optimal number of delay blocks, finding the optimal values of the delays, and studying the robustness of the designed controller are also discussed in the present work. Investigating what class of LTI controllers can be approximated by delay-based controllers with a small number of delays remains a subject of future research.

## ACKNOWLEDGMENT

This research was supported by ONR MURI N00014-08-1-0747 ‘‘Scalable, Data-driven, and Provably-correct Analysis of Networks,’’ ARO MURI W911NF-08-1-0233 ‘‘Tools for the Analysis and Design of Complex Multi-Scale Networks,’’ and the Army’s W911NF-09-D-0001 Institute for Collaborative Biotechnology.

## REFERENCES

- [1] K. Zhou, J. Doyle, and K. Glover, *Robust and optimal control*, Prentice-Hall, 1996.
- [2] G. E. Dullerud and F. G. Paganini, *A course in robust control theory: A convex Approach*, Springer, 2000.
- [3] P. Ioannou, B. Fidan, *Adaptive control tutorial*, SIAM, Advances in Design and control, 2006.
- [4] R. Murphey and P. M. Pardalos, *Cooperative control and optimization*, Springer, 2002.
- [5] T. Katayama, *Subspace methods for system identification*, Springer, 2005.
- [6] R. W. Brockett, ‘‘Reduced complexity control systems, in *Plenary Papers, Milestone Reports, & Selected Survey Papers*, Myung Jin Chung and Pradeep Misra, Eds., 17th IFAC World Congress, 2008.
- [7] K. Gu, V. Kharitonov, and J. Chen, *Stability of time-delay systems*, Birkhuser, 2003.
- [8] S. I. Niculescu and K. Gu, *Advances in time-delay systems*, Springer, 2004.
- [9] Q. C. Zhong, *Robust control of time-delay systems*, Springer, 2006.
- [10] S. I. Niculescu, ‘‘Delay effects on stability,’’ in *Lecture notes in control and information sciences*, Berlin: Springer, Vol. 269, 2001.
- [11] D. Bratsun, D. Volfson, L. S. Tsimring, and J. Hasty, ‘‘Delay-induced stochastic oscillations in gene regulation,’’ *Proceedings of the National Academy of Sciences*, vol. 102, no. 41, pp. 14593-14598, 2005.
- [12] K. Watanabe, E. Nobuyama, and A. Kojima, ‘‘Recent advances in control of time delay systems: a tutorial review,’’ in *Proceedings of the 35th IEEE Conference on Decision and Control*, pp. 2083-2089, 1996.
- [13] C. Abdallah, P. Dorato, J. Benites-Read, and R. Byrne, ‘‘Delayed positive feedback can stabilize oscillatory systems,’’ in *Proceedings of 1993 American Control Conference*, pp. 3106-3107, 1993.
- [14] G. Orosz, J. Moehlis, and R. M. Murray, ‘‘Controlling biological networks by time-delayed signals,’’ *Philosophical Transaction of the Royal Society A*, vol. 368, no. 1911, pp. 439-454, 2010.
- [15] R. A. Mao, K. R. Keller, and R. W. Ahrons, ‘‘Integrated MOS analog delay line,’’ *IEEE Journal of Solid-State Circuits*, vol. 4, no. 4, pp. 196-201, 1969.
- [16] A. Chamarti and K. Varahramyan, ‘‘Transmission delay line based ID generation circuit for RFID applications,’’ *IEEE Microwave and Wireless Components Letters*, vol. 16, no. 11, pp. 588-590, 2006.
- [17] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, *Molecular biology of the cell*, Fourth edition, Garland Science, 2002.
- [18] F. Rena and J. Cao, ‘‘Asymptotic and robust stability of genetic regulatory networks with time-varying delays,’’ *Neurocomputing*, vol. 71, no. 4-6, pp. 834-842, 2008.
- [19] M. B. Elowitz and S. Leibler, ‘‘A synthetic oscillatory network of transcriptional regulators,’’ *Nature*, vol. 403, no. 20, pp. 335-338, 2000.
- [20] T. S. Gardner, C. R. Cantor, and J. J. Collins, ‘‘Construction of a genetic toggle switch in escherichia coli,’’ *Nature*, vol. 403, no. 20, pp. 339-342, 2000.
- [21] A. Moini, K. Eshraghian, and A. Bouzerdoum, ‘‘The impact of VLSI technologies on neural networks,’’ in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 158-163, 1995.
- [22] S. L. Hakimi and E. F. Schmeichel, ‘‘Fitting polygonal functions to a set of points in the plane,’’ *Graphical Models and Image Processing*, vol. 53, no. 2, pp. 132-136, 1991.
- [23] D. U. Campos-Delgado and K. Zhou, ‘‘ $H_\infty$  strong stabilization,’’ *IEEE Transactions on Automatic Control*, vol. 46, no. 12, pp. 1968-1972, 2001.
- [24] S. Gumussoy and H. Ozbay, ‘‘Remarks on strong stabilization and stable  $H_\infty$  controller design,’’ *IEEE Transactions on Automatic Control*, vol. 50, no. 12, pp. 2083-2087, 2005.