CDS Technical Report 2010.002; Submitted, IEEE T. Automatic Control
http://www.cds.caltech.edu/~murray/papers/lsm10-tac.html

Delay-Based Controller Design for Continuous-Time and Hybrid Applications

Javad Lavaei, Somayeh Sojoudi and Richard M. Murray

Abstract

Motivated by the availability of different types of delays in embedded systems and biological circuits, the objective of this work is to study the benefits that delay can provide in simplifying the implementation of controllers for continuous-time systems. Given a continuous-time linear time-invariant (LTI) controller, we propose three methods to approximate this controller arbitrarily precisely by a simple controller composed of delay blocks, a few integrators and possibly a unity feedback. Different problems associated with the 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 investigated. We also study the design of an LTI continuous-time controller satisfying given control objectives whose delay-based implementation needs the least number of delay blocks. A direct application of this work is in the sampled-data control of a real-time embedded system, where the sampling frequency is relatively high and/or the output of the system is sampled irregularly. Based on our results on delay-based controller design, we propose a digital-control scheme that can implement every continuous-time stabilizing (LTI) controller. Unlike a typical sampled-data controller, the hybrid controller introduced here—consisting of an ideal sampler, a digital controller, a number of modified second-order holds and possibly a unity feedback—is robust to sampling jitter and can operate at arbitrarily high sampling frequencies without requiring expensive, high-precision computation.

I. Introduction

The field of control systems has seen remarkable progress in areas such as robust control, adaptive control, cooperative control, system identification and optimal control [15], [26], [28], [35], [46]. 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. It is, therefore, potentially useful to impose a simplicity constraint on the

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).

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 [9] 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 in the continuous-time domain due to the ubiquity of communication, computation or propagation delays in both embedded systems and biological circuits [22], [37], [45]. The book [36] describes 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 can benefit the control process. For instance, delay can be used to create a limit cycle for nonlinear systems [8], to perform deadbeat tracking for continuous-time systems [42], or to stabilize oscillatory systems [1], [39]. In the continuous-time domain, 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/communications and cavity delay lines (trombone delay lines) in optics play the role of delay lines [11], [33].

Neurons and gene regulatory networks are two sources of delays in biology [2]. Time delays appear in genetic networks due to transcription, translation, and translocation processes [39], [40]. Time delays have important roles in biological systems such as causing protein levels to oscillate in gene regulatory networks or making different rhythmic spatio-temporal patterns in neural networks [14], [38], [39]. Recently, there has been a considerable amount of interest in synthetic biology, whose goal is to build artificial biological systems for engineering applications [16], [21], [34]. This is often achieved by assembling and programming different biological components in such a way that the resulting circuit performs a computation, fabricates a molecular-scale structure or controls a system of molecular sensors and actuators [3], [17], [41]. By regarding a biological system composed of several interacting components as a distributed control system, two easy-to-manipulate parameters for design purposes are (i) the topology of the distributed system (interaction topology) and (ii) time delays in the interactions. Hence, the interaction graph together with the amount of delays in the interactions plays the role of the controller in a biological system. A primary motivation of the present work is the necessity of treating delays as the control parameters for this important class of emerging systems.

In more traditional engineered control systems, a common source of delays is the discrete delay in clocked systems. Since the invention of digital circuits and digital computers, there has been an every-growing interest in the digital control of continuous-time systems. Computer controlled systems have been widely used in a broad range of applications from robotics, autopilot and radar to anti-lock braking systems [4], [29]. A typical digital-control scheme for a continuous-time system is composed of an analog-to-digital converter (*sampler*), a digital processor and a digital-to-analog converter (*hold circuit*). This configuration is referred to as *sampled-data control system* and has been long studied in the literature [12], [27], [44]. Among many problems that have been investigated in the context of sampled-data control systems are stability, robustness, sensitivity, frequency-domain characterization, H_2 and H_∞ sampled-data controllers and best achievable tracking performance [6], [13], [18], [24], [30], [43]. Current silicon technology has enabled the design of embedded systems operating at very high frequencies [5]. However, the conventional methods for the synthesis of sampled-data control systems require high processing power to cope with numerical issues if the sampling rate is relatively fast. More precisely, increasing the sampling frequency makes the digital controller extremely sensitive to measurement noise and computational round-off errors. The situation becomes worse if the sampling is subject to jitter and irregularities.

In this paper, we first consider the continuous-time domain in which the delay operator does not appear naturally. Given a continuous-time linear time-invariant (LTI) controller, we 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. This result implies that every high-order LTI controller has a simple delay-based implementation, which uses delay blocks rather than several integrators. Several properties of the proposed delay-based controller are investigated throughly in this paper. Later on, we tackle the problem of designing a continuous-time LTI controller satisfying given control objectives whose delay-based implementation needs the least number of delay blocks.

As an application of the aforementioned results, we propose a robust digital-control scheme for continuous-time systems that can be used in two important scenarios: (i) having a high sampling frequency with limited computational power (ii) having a slow processor with jitter and irregular sampling times. Note that the second scenario occurs when the sampling frequency is relatively faster than the slow processing rate and, in addition, the sampling times are prone to delays and irregularities [32]. The main focus here will be on the first application (scenario), while the second application can be treated similarly. We show that every continuous-time stabilizing (LTI) controller can be implemented in a hybrid form consisting of a sampler, a digital processor, some so-called "modified second-order holds" and possibly a unity feedback from the hold circuit to the sampler. This hybrid controller benefits from the fact that the increase of the sampling frequency has a direct influence only on the memory size of the controller, as opposed to its parameters. This property makes the parameters of the controller robust to the sampling rate and irregularities.

The rest of the paper is organized as follows. The delay-based implementation of a given controller in the continuous-time domain is studied in Section II, and subsequently the delay-based controller design is tackled in Section III. The results are then applied to the sampled-data control problem in Section IV. Simulation results are presented in Section V to illustrate the techniques developed here. Finally, some concluding remarks are given in Section VI.

II. Continuous-Time Delay-Based Implementation

Consider a continuous-time LTI system $\mathcal{S}$ 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 $x(t) \in \Re^n$, $u(t) \in \Re^m$ and $y(t) \in \Re^r$ denote the state, input and output of the system, respectively. Let $P(s)$ denote the transfer function of $\mathcal{S}$. 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 $G(s) = L \in \Re^{r\times m}$, i.e., $u(t) = Ly(t)$. 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:

$$u(t) = \sum_{i=1}^{p} \alpha_i y(t - \tau_i), \tag{2}$$

where $\alpha_1, \alpha_2, ..., \alpha_p$ are constant gains and $\tau_1, \tau_2, ..., \tau_p$ 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 $\sum_{i=1}^{p} \alpha_i e^{-\tau_i s}$. Since the decision problem of whether there exists a stabilizing controller of the form $u(t) = Ly(t)$ is NP-hard, it is expected that a direct design of a controller of the type (2) is cumbersome. This section aims to develop an indirect method for designing a controller in the form of (2) based on a given LTI controller. To this end, consider a given LTI controller $G(s)$ satisfying prescribed design specifications. We write a state-space realization of $G(s)$ as

$$\begin{aligned} \dot{x}_c(t) &= A_c x_c(t) + B_c y(t), \\ u(t) &= C_c x_c(t) + D_c y(t), \end{aligned} \tag{3}$$

where $x_c(t) \in \Re^{n_c}$ represents the state of the controller. The first goal of this part is to approximate the given controller $G(s)$ by a simple delay-based controller $\hat{G}(s)$ of the form $\sum_{i=1}^{p} \alpha_i e^{-\tau_i s}$ 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 (2) that still have easy implementation and can approximate every stabilizing controller.

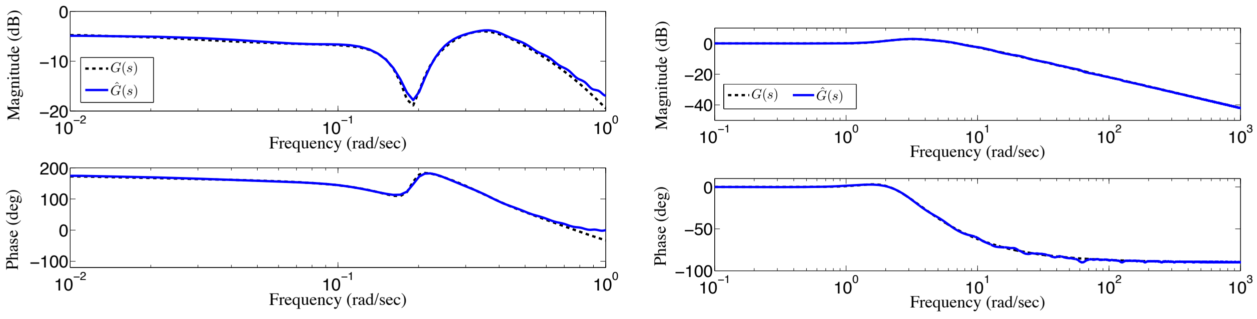
A. Illustrative Examples

Before developing the main results, we wish to illustrate with two examples how a high-order rational controller $G(s)$ can be approximated by a simple delay-based controller.

Example 1: Consider the dynamical control system $\mathcal{G}_n$

$$\begin{aligned} \dot{m}_i(t) &= \frac{1}{0.05 + p_{\sigma(i)}(t)} + 0.05 - 0.502 m_i(t) + l_i \zeta_1(t), \quad i = 1, 2, ..., 10, \\ \dot{p}_i(t) &= 0.35 m_i(t) - 0.7 p_i(t), \quad i = 1, 2, ..., 10, \\ \zeta_2(t) &= m_1(t) + p_4(t) + p_{10}(t), \end{aligned} \tag{4}$$

with the parameters $(l_1, l_2, ..., l_{20}) = (1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,1)$ and $(\sigma(1), \sigma(2), ..., \sigma(10)) = (10, 1, 2, ..., 9)$. The control system $\mathcal{G}_n$ is indeed a specific gene regulatory network consisting of 10 gene-protein pairs in a ring, where $\zeta_1(t)$ is an external injection signal, $\zeta_2(t)$ is the output of the circuit, and $m_i(t), p_i(t)$ are the concentrations of mRNA and protein for the i^{th} species [39]. We linearize the system $\mathcal{G}_n$ around its unique positive equilibrium point



(a) (b)

Fig. 1. (a): The Bode plots of the controllers $G(s)$ (dashed) and $\tilde{G}(s)$ (solid) for Example 1 in Subsection II-A; (b): The Bode plots of the controllers $G(s)$ and $\tilde{G}(s)$ for Example 2 in Subsection II-A.

and denote the linearized system as $\mathcal{G}(s)$. Since the control system $\mathcal{G}(s)$ has very slow dynamics (due to the existence of the mode with eigenvalue -0.0285), let the frequency range of interest be the interval $[0,1]$. By considering $\mathcal{G}_n$ as a part of a larger gene regulatory network $\mathcal{N}$, the circuit $\mathcal{G}_n$ can be regarded as a controller in the network $\mathcal{N}$. The goal of this example is to study if the controller $G(s)$ can be replaced by a simple delayed-based controller in the network $\mathcal{N}$. To this end, define $\hat{G}(s) := \sum_{j=0}^{20} e^{-3js} g(3j)$, where $g(t)$ denotes the impulse response of the controller $G(s)$. The Bode diagrams of the controllers $G(s)$ and $\hat{G}(s)$ are plotted over the frequency range $[0,1]$ in Figure 1(a), which show that the static controller $\hat{G}(s)$ with only 20 delay blocks performs very similarly to the controller $G(s)$ with 20 states over the desired range of frequencies.

Example 2: Consider the (admittedly artificial) controller $G(s) = 1 - \frac{(s+0.9)^{80}}{(s+1)^{80}}$, which is hard to approximate by a low-order LTI controller due to its repeated poles. 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 function $\hat{g}(t)$ is shown in Figures 2(a) and 2(b). This piecewise linear function has 10 knots given by the vector $\boldsymbol{\tau}$ as follows:

$$\boldsymbol{\tau} = \begin{bmatrix} 0 & 0.1 & 0.2 & 0.3 & 0.4 & 0.5 & 0.81 & 1.21 & 1.96 & 2.72 \end{bmatrix}. \tag{5}$$

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

$$\begin{aligned} &\hat{g}(\tau_1) = 7.901, \quad \hat{g}(\tau_2) = 4.631, \quad \hat{g}(\tau_3) = 2.505, \quad \hat{g}(\tau_4) = 1.121, \quad \hat{g}(\tau_5) = 0.312, \\ &\hat{g}(\tau_6) = -0.264, \quad \hat{g}(\tau_7) = -0.551, \quad \hat{g}(\tau_8) = -0.14, \quad \hat{g}(\tau_9) = 0.1, \quad \hat{g}(\tau_{10}) = 0.0163, \end{aligned} \tag{6}$$

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

$$\hat{G}(s) := \sum_{i=1}^{9} \left(\frac{w_i}{s^2} + \frac{\hat{g}(\tau_i)}{s} \right) e^{-\tau_i s} + \sum_{i=1}^{9} \left(-\frac{w_i}{s^2} - \frac{\hat{g}(\tau_{i+1})}{s} \right) e^{-\tau_{i+1} s}, \tag{7}$$

where

$$w_i = \frac{\hat{g}(\tau_{i+1}) - \hat{g}(\tau_i)}{\tau_{i+1} - \tau_i}, \quad i = 1, 2, ..., 9. \tag{8}$$

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 1(b) to show how closely $\hat{G}(s)$ approximates $G(s)$. Note that a purely integrator-based implementation of $G(s)$ performing as well as $\hat{G}(s)$ is expected to need more than 30 integrators (this can be verified using the balanced model-reduction method [15]).

To develop a concrete theory for the general case, assume for now that $G(s)$ is a single-input single-output (SISO) controller that is (asymptotically) stable. These assumptions will be removed in Subsections II-G and II-H. With no loss of generality, we suppose 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 $\hat{G}(s)$.

B. Method 1: Approximation by pure delays

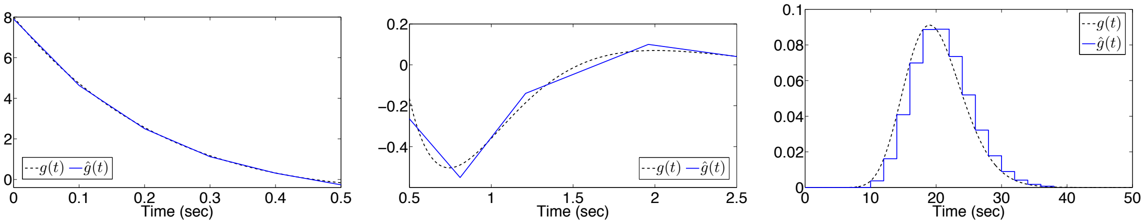
Let $\hat{G}(s)$ be a function in the Laplace domain that is analytic on the open left-half s-plane. It follows from the maximum modulus theorem and the stability of the controller $G(s)$ that

$$\max_{s:\ \text{Re}\{s\} \geq 0} |G(s) - \hat{G}(s)| = \max_{\omega \in \Re} |G(j\omega) - \hat{G}(j\omega)|, \tag{9}$$

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 computed by restricting evaluation to the $j\omega$ axis. On the other hand, the definition of the Fourier transform yields

$$G(j\omega) = \int_0^\infty g(t) e^{-j\omega t} dt. \tag{10}$$

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



(a)

(b)

(c)

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).

can take advantage of any integral approximation method, such as the midpoint method. More precisely, consider some nonnegative numbers $\tau_1 < \tau_2 < ... < \tau_p$ and define $\hat{G}(s)$ as

$$\hat{G}(s) = \sum_{i=1}^{p-1} g(\tau_i)(\tau_{i+1} - \tau_i)e^{-\tau_i s} \tag{11}$$

or

$$\hat{G}(s) = \sum_{i=1}^{p-1} g(\bar{\tau}_i)(\tau_{i+1} - \tau_i)e^{-\bar{\tau}_i s}, \tag{12}$$

where $\bar{\tau}_i = \frac{\tau_i + \tau_{i+1}}{2}$ for $i = 1, 2, ..., p-1$. The main focus of this subsection will be on the approximating controller (12) 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 $\omega \in \Re$:

$$\begin{aligned} |G(j\omega) - \hat{G}(j\omega)| \leq \sqrt{2}\int_0^{\tau_1} |g(t)|dt + \sqrt{2}\int_{\tau_p}^{\infty} |g(t)|dt \\ + \sqrt{2}\sum_{i=1}^{p-1} \frac{(\tau_{i+1} - \tau_i)^3}{24} \max_{\tau \in [\tau_i, \tau_{i+1}]} \left\{ \left| \frac{\partial^2 \big(g(\tau)\cos(\omega\tau)\big)}{\partial^2 \tau} \right|, \left| \frac{\partial^2 \big(g(\tau)\sin(\omega\tau)\big)}{\partial^2 \tau} \right| \right\}. \end{aligned} \tag{13}$$

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 ω due to the existence of the second derivative of the term $\cos(\omega\tau)$. This fact can also be justified from another standpoint: if $\tau_1, \tau_2, .., \tau_p$ 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. As a result, $\hat{G}(j\omega)$ cannot approximate $G(j\omega)$ for high frequencies.

However, in the case when the system 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 (13) that the numbers $\tau_1, \tau_2, ..., \tau_p$ (in addition to p) can be chosen in such a way that $\hat{G}(j\omega)$ approximates $G(j\omega)$ arbitrarily precisely over any desired range of frequencies.

C. Method 2: Approximation by step-like functions

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 $\tau_1, \tau_2, ..., \tau_p$, the function $g(t)$ can be approximated by a step-like function such as

$$\hat{g}(t) = \begin{cases} g(\tau_i) & t \in [\tau_i, \tau_{i+1}], \quad i = 1, 2, ..., p-1 \\ 0 & t < \tau_1 \;\text{ or }\; t > \tau_p \end{cases} \tag{14}$$

or

$$\hat{g}(t) = \begin{cases} g\left(\bar{\tau}_i\right) & t \in [\tau_i, \tau_{i+1}], \quad i = 1, 2, ..., p-1 \\ 0 & t < \tau_1 \;\text{ or }\; t > \tau_p \end{cases} \tag{15}$$

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 (15) is as follows:

$$\hat{G}(s) = \frac{1}{s} \sum_{i=1}^{p} \alpha_i e^{-\tau_i s}, \tag{16}$$

where

$$\alpha_1 := g(\bar{\tau}_1), \quad \alpha_i := g(\bar{\tau}_i) - g(\bar{\tau}_{i-1}) \quad (i = 2, 3..., p-1), \quad \alpha_p := -g(\bar{\tau}_{p-1}). \tag{17}$$

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

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

$$\begin{aligned} |G(j\omega) - \hat{G}(j\omega)| \leq \sqrt{2} \int_0^{\tau_1} |g(t)| dt + \sqrt{2} \int_{\tau_p}^{\infty} |g(t)| dt + \sum_{i=1}^{p-1} \max_{\tau \in [\tau_i, \tau_{i+1}]} |g''(\tau)| \frac{\sqrt{2}(\tau_{i+1} - \tau_i)^3}{24} \\ + \sqrt{2} \sum_{i=1}^{p-1} |g'(\bar{\tau}_i)| \max\left\{ |\text{Re}\{H(i,\omega)\}|, |\text{Im}\{H(i,\omega)\}| \right\}, \end{aligned} \tag{18}$$

where

$$H(i,\omega):=\int_{\tau_i}^{\tau_{i+1}}(t-\bar{\tau}_i)e^{-j\omega t}dt,\ i=1,2,...,p-1,\ \omega\in\Re. \tag{19}$$

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

$$g(t)=g(\bar{\tau}_i)+g'(\bar{\tau}_i)(t-\bar{\tau}_i)+\frac{g''(\gamma(t))}{2}(t-\bar{\tau}_i)^2. \tag{20}$$

Therefore

$$\begin{aligned}|\text{Re}\{G(j\omega)-\hat{G}(j\omega)\}|&=\left|\int_0^\infty(g(t)-\hat{g}(t))\cos(\omega t)dt\right|\leq\sum_{i=1}^{p-1}\int_{\tau_i}^{\tau_{i+1}}\left|\frac{1}{2}g''(\gamma(t))(t-\bar{\tau}_i)^2\cos(\omega t)dt\right|\\&+\sum_{i=1}^{p-1}\left|\int_{\tau_i}^{\tau_{i+1}}g'(\bar{\tau}_i)(t-\bar{\tau}_i)\cos(\omega t)dt\right|+\left|\int_0^{\tau_1}g(t)\cos(\omega t)dt\right|+\left|\int_{\tau_p}^{\infty}g(t)\cos(\omega t)dt\right|\\&\leq\sum_{i=1}^{p-1}\max_{\tau\in[\tau_i,\tau_{i+1}]}|g''(\tau)|\frac{(\tau_{i+1}-\tau_i)^3}{24}+\int_0^{\tau_1}|g(t)|dt+\sum_{i=1}^{p-1}|g'(\bar{\tau}_i)|\,|\text{Re}\{H(i,\omega)\}|+\int_{\tau_p}^{\infty}|g(t)|dt.\end{aligned} \tag{21}$$

An inequality similar to (21) can be written for $|\text{Im}\{G(j\omega)-\hat{G}(j\omega)\}|$ which together with (21) 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 ω goes to infinity. The inequality provided in Theorem 2 implies that one can design the delays $\tau_1,\tau_2,...,\tau_p$ (besides p) 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).

D. Method 3: Piecewise linear approximation

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 $\tau_1,\tau_2,...,\tau_p$ (listed in an ascending order), namely

$$\hat{g}(t)=\begin{cases}\frac{g(\tau_{i+1})-g(\tau_i)}{\tau_{i+1}-\tau_i}(t-\tau_i)+g(\tau_i) & t\in[\tau_i,\tau_{i+1}]\\ 0 & t<\tau_1\ \text{ or }\ t>\tau_p\end{cases} \tag{22}$$

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

$$\hat{G}(s)=\sum_{i=1}^{p}\beta_i(s)e^{-\tau_i s}, \tag{23}$$

where

$$\beta_1(s) := \frac{w_1}{s^2} + \frac{g(\tau_1)}{s}, \quad \beta_i(s) := \frac{w_i}{s^2} - \frac{w_{i-1}}{s^2} \quad (i = 2, ..., p-1), \quad \beta_p(s) := -\frac{w_{p-1}}{s^2} - \frac{g(\tau_p)}{s},$$

and

$$w_i := \frac{g(\tau_{i+1}) - g(\tau_i)}{\tau_{i+1} - \tau_i}, \qquad i = 1, 2, ..., p-1. \tag{24}$$

Note that the approximating controller $\hat{G}(s)$ introduced above can be implemented using p 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:

$$\|G(s) - \hat{G}(s)\|_\infty \leq \sqrt{2}\int_0^{\tau_1} |g(t)|dt + \sqrt{2}\int_{\tau_p}^{\infty} |g(t)|dt + \sum_{i=1}^{p-1} \max_{\tau\in[\tau_i,\tau_{i+1}]} |g''(\tau)| \frac{\sqrt{2}(\tau_{i+1} - \tau_i)^3}{12}. \tag{25}$$

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

$$g(t) - \hat{g}(t) = \frac{1}{2}g''(\eta(t))(t - \tau_{i+1})(t - \tau_i), \quad t \in [\tau_i, \tau_{i+1}], \tag{26}$$

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

$$\begin{aligned}
|\text{Re}\{G(j\omega) - \hat{G}(j\omega)\}| &= \left|\int_0^{\infty} (g(t) - \hat{g}(t))\cos(\omega t)dt\right| \leq \left|\int_0^{\tau_1} g(t)\cos(\omega t)dt\right| \\
&+ \left|\int_{\tau_p}^{\infty} g(t)\cos(\omega t)dt\right| + \sum_{i=1}^{p-1}\int_{\tau_i}^{\tau_{i+1}} \left|\frac{1}{2}g''(\eta(t))(t - \tau_{i+1})(t - \tau_i)\cos(\omega t)dt\right| \\
&\leq \frac{1}{2}\sum_{i=1}^{p-1} \max_{\tau\in[\tau_i,\tau_{i+1}]} |g''(\tau)| \int_{\tau_i}^{\tau_{i+1}} (\tau_{i+1} - t)(t - \tau_i)dt + \int_0^{\tau_1} |g(t)|dt + \int_{\tau_p}^{\infty} |g(t)|dt \\
&= \sum_{i=1}^{p-1} \max_{\tau\in[\tau_i,\tau_{i+1}]} |g''(\tau)| \frac{(\tau_{i+1} - \tau_i)^3}{12} + \int_0^{\tau_1} |g(t)|dt + \int_{\tau_p}^{\infty} |g(t)|dt.
\end{aligned} \tag{27}$$

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

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

In this subsection, we approximated the time-domain signal $g(t)$ with a piecewise linear function, by assuming 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.

E. Optimal Choice of Delays

Three methods have been proposed in the preceding subsections 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 $\{\tau_1, \tau_2, ..., \tau_p\}$. 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 $\hat{g}(t)$. Given a function norm $\|\cdot\|$ (namely 1 or ∞ 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 ε. 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 ∞-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 h to obtain a discretized signal $g_h(t)$. The goal is to find a discrete piecewise linear signal $\hat{g}_h(t)$ such that $\|g_h(t) - \hat{g}_h(t)\|_\infty \leq \varepsilon$. Four problems can be defined as follows for a given positive real ε and a natural number p:

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

Note that the delays being found will be all multiples of the sampling time h. Let N denote the number of discrete points of the function $g_h(t)$. It is shown in [25] 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.

To solve P2, represent the points of the discrete signal $g_h(t)$ with $p_1, p_2, ..., p_N$. Construct a directed graph $\mathcal{G}$ with N vertices as follows. For every $i, j \in \{1, 2, ..., N\}$ 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 $p_i, p_{i+1}, ..., p_j$ is less than or equal to ε. 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.

F. Unstable Controllers

Assume that a given controller $G(s)$ is unstable. The next question would be how to implement this controller 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 system $\mathcal{S}$, the controller itself must be stabilizable. Therefore, there exists a matrix gain $L \in \Re^{1 \times n_c}$ such that $A_c - B_c L$ is Hurwitz. Define $w(t) := L x_c(t)$ and $e(t) := y(t) + w(t)$. The controller G is equivalent to the feedback configuration given in Figure 3, whose backward path is a unity feedback and whose forward path is a controller $G_e(s)$ with the control law

$$\begin{aligned} \dot{x}_c(t) &= (A_c - B_c L) x_c(t) + B_c e(t), \\ u(t) &= C_c x_c(t), \\ w(t) &= L x_c(t). \end{aligned} \tag{28}$$

It can be observed that the controller $G_e(s)$ with the single input $e(t)$ and the outputs $u(t)$ and $w(t)$ is stable. Now, each of the transfer functions from "$e(t)$ to $u(t)$" and "$e(t)$ to $w(t)$"

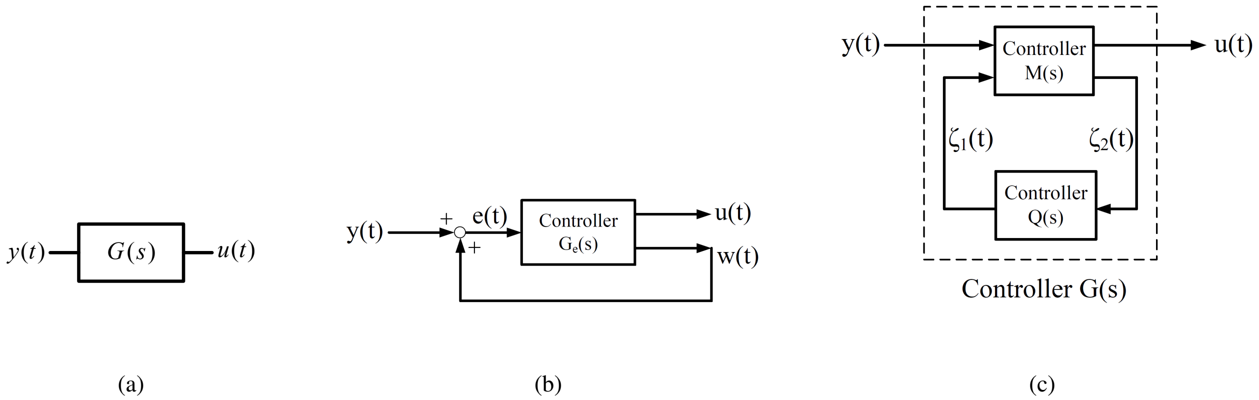


Fig. 3. Figures (a) and (b) show an unstable controller and its equivalent feedback representation, respectively, where the forward path controller $G_e(s)$ is stable. Figure (c) describes a generic model for the stabilizing controller $G(s)$.

can be approximated by a simple delay-based controller. This makes the controller $G_e(s)$ be approximated by a controller $\hat{G}_e(s)$ consisting of delay blocks and at most four integrators (due to the existence of two SISO transfer functions). As a result, every stabilizing unstable controller $G(s)$ can be approximated by a feedback controller with the unity feedback whose forward path is a delay-based controller.

G. Multi-Input Multi-Output and Distributed Control Systems

Assume for now that the controller $G(s)$ is multi-input single-output. The results developed earlier can be adopted to show that:

- If $G(s)$ is stable, each of Methods 1, 2 or 3 can be easily used to approximate $G(s)$ with a delay-based controller consisting of at most two integrators.
- If $G(s)$ is unstable, it should be first realized as the configuration given in Figure 3(b). Then, the forward path can be approximated using a delay-based controller with the main difference that the resulting controller could potentially need more than 4 integrators. The reason is that the signal $w(t)$ is no longer a scalar, and hence the approximation of the transfer function from $e(t)$ to $w(t)$ requires as many double-integrators as the number of nonzero rows of the auxiliary matrix L. Note that if, for instance, $G(s)$ is stabilizable through one of its single inputs, there exists a proper matrix L with only one nonzero row.

In the case when $G(s)$ is a multi-input multi-output (MIMO) controller, regard this as a set of multi-input single-output sub-controllers and then apply the above result to each of these sub-controllers.

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.

H. Stability Issue

Recall that the approximating controller $\hat{G}(s)$ obtained using Method 2 or Method 3 includes one or two integrators. A potential concern is that $\hat{G}(s)$ could have 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

$$\hat{g}(t) = \begin{cases} g(\tau_i) & t \in [\tau_i, \tau_{i+1}], \quad i = 1, 2, ..., p-1 \\ 0 & t < \tau_1 \text{ or } t > \tau_p \end{cases} \tag{29}$$

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

$$\hat{g}(t) = \begin{cases} g(\tau_i)e^{-\alpha(t-\tau_i)} & t \in [\tau_i, \tau_{i+1}], \; i = 1, 2, ..., p-1 \\ 0 & t < \tau_1 \text{ or } t > \tau_p \end{cases} \tag{30}$$

where α 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 p delay terms along with the stable low-pass filter $\frac{1}{s+\alpha}$ as opposed to an integrator.

I. Stability and Robustness

It was shown earlier 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 system $\mathcal{S}$ 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) that is approximated by a unity feedback, as depicted in Figure 3, with a delay-based sub-controller $\hat{G}_e(s)$ in the forward path. Note that the case of a stable controller $G(s)$ is a special case of this setting by letting L be zero. This subsection develops some results for the SISO case, which can be easily generalized to the MIMO case. Notice that $\hat{G}_e(s)$ is an approximation of the sub-controller $G_e(s)$, 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 uncertainty functions such that

$$\hat{G}_e(j\omega) = G_e(j\omega)(1 + \Delta_1(j\omega)) + \Delta_2(j\omega), \quad \forall \omega \in \Re \tag{31}$$

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

$$\left\| \frac{\bar{P}G_e}{1 + \bar{P}G_e} \right\|_\infty |\Delta_1(j\omega)| + \left\| \frac{\bar{P}}{1 + \bar{P}G_e} \right\|_\infty |\Delta_2(j\omega)| < 1, \quad \forall \omega \in \Re, \tag{32}$$

where $\bar{P}(s) = \left[\begin{array}{cc} P(s) & 1 \end{array} \right]$. 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 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 $\tau_1 + \delta\tau_1, \tau_2 + \delta\tau_2, ..., \tau_p + \delta\tau_p$ be used in the delay blocks instead of the nominal values $\tau_1, \tau_2, ..., \tau_p$. This means that the approximating controller

$$\hat{G}(s) = \sum_{i=1}^{p-1} g(\tau_i)(\tau_{i+1} - \tau_i)e^{-\tau_i s} \tag{33}$$

will be perturbed as follows:

$$\hat{G}(s) + \Delta\hat{G}(s) = \sum_{i=1}^{p-1} g(\tau_i)(\tau_{i+1} - \tau_i)e^{-(\tau_i + \delta\tau_i)s}. \tag{34}$$

It is easy to observe that $\Delta\hat{G}(j\omega)$ is negligible for small values of ω; in particular, $\Delta\hat{G}(0) = 0$. However, $\Delta\hat{G}(j\omega)$ may become large for a high frequency ω. 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 the filtering property of the system).

Remark 1: The two above-mentioned analyses are based on the assumption that the delay blocks operate either at the nominal values $\tau_1, ..., \tau_p$ or at some fixed perturbed values $\tau_1 + \delta\tau_1, \tau_2 + \delta\tau_2, ..., \tau_p + \delta\tau_p$. Nevertheless, the delay terms are always subject to jitter in practice, which make them time-varying as $\tau_1 + \delta\tau_1(t), \tau_2 + \delta\tau_2(t), ..., \tau_p + \delta\tau_p(t)$. The foregoing stability analyses can be easily adopted to derive sufficient conditions guaranteeing the bounded-input bounded-output stability of the closed-loop system under time-varying delays. For instance, the condition (32) should be modified as

$$\left\| \frac{\bar{P}G_e}{1+\bar{P}G_e} \right\|_\infty \|\Delta_1\|_\infty + \left\| \frac{\bar{P}}{1+\bar{P}G_e} \right\|_\infty \|\Delta_2\|_\infty < 1, \tag{35}$$

where Δ_1 and Δ_2 are time-varying uncertainties that account for both the impulse response approximation error and jitter in the delay terms. Other sufficient conditions can be obtained using the techniques discussed in [22], which turn out to be explicit in terms of the variation rates $\dot{\delta\tau}_1(t), ..., \dot{\delta\tau}_p(t)$.

III. Near-Optimal Delay-Based Controller Design

The problem of implementing a given continuous-time controller $G(s)$ in a delay-based form was investigated in the preceding section. Now, assume that some design specifications are provided instead of a controller $G(s)$ directly. Since there may exist an infinite number of controllers $G(s)$ satisfying the underlying design objectives, we wish to find the one whose delay-based implementation needs the least number of delay blocks. To this end, for simplicity and with no loss of generality, assume that the system $\mathcal{S}$ is strongly stabilizable, single-input multi-output and its direct term D is equal to zero. Two methods will be proposed in the sequel for designing a stable controller $G(s)$ whose delay-based implementation is near-optimal, where:

- In method 1, the order of the unknown controller $G(s)$ is set *a priori* and the design specifications are rather general.
- In method 2, the order of the unknown controller $G(s)$ is arbitrary (not fixed), and the stability of the closed-loop system is the only design objective.

For the first method, denote the order of the controller $G(s)$ being designed as n_c and the given design specifications as $\mathcal{D}$. Assume that the control specifications $\mathcal{D}$ can be translated into a matrix inequality as

$$\mathcal{L}(A_c, B_c, C_c, R) \prec 0, \tag{36}$$

for some slack (matrix) variable R and matrix operator $\mathcal{L}$ that is bilinear (quadratic) in its argument, where $\prec$ represents the matrix inequality in the negative-definite sense. It is noteworthy that many specifications such as guaranteed H_2 performance, guaranteed H_∞ performance, robust pole-placement or any combinations of these specifications can be expressed in the above form (even the ones involving rank constraints) [7], [15]. The simplicity of the best piecewise linear approximation of $g(t)$ is directly related to how smooth (up to the second order) this function is. Hence, the performance index

$$J := \int_0^\infty \|g''(t)\|_2^2 \ dt, \tag{37}$$

where $\|\cdot\|_2$ denotes the 2-norm operator, is a measure of the difficulty of approximating $g(t)$ by a piecewise linear function. In particular, when J is equal to 0, the impulse response $g(t)$ must be a line. Thus, the goal is to minimize the performance index J in order to find a controller $G(s)$ whose digital implementation is near-optimal. The stabilizable controller $G(s)$ being found can be assumed to be both controllable and observable (because an infinitesimal perturbation of a stabilizable controller always makes it controllable and observable). The state-space representation (A_c, B_c, C_c) of $G(s)$ can be considered to be in the observable form, implying that C_c is equal to $\begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}$. Therefore, the only unknown parameters are A_c and B_c. We introduce the following optimization problem.

Optimization 1: Minimize the scalar α subject to

$$\mathcal{L}(A_c, B_c, C_c, R) \prec 0, \quad \begin{bmatrix} A_cP + PA_c^T & A_cB_c \\ B_c^TA_c^T & -I \end{bmatrix} \prec 0, \quad \begin{bmatrix} -\alpha & C_cA_cP \\ PA_c^TC_c^T & -P \end{bmatrix} \prec 0, \tag{38}$$

for matrix variables $A_c \in \Re^{n_c \times n_c}$ and $B_c \in \Re^{n_c \times r}$, a symmetric matrix variable $P \in \Re^{n_c \times n_c}$ and a slack variable R of appropriate dimension, where A_c is in the (observable) canonical form.

Denote the optimal values of the matrices A_c and B_c solving Optimization 1 with A_c^* and B_c^*, respectively. The objective is to show that Optimization 1 indeed minimizes the performance index J and, more precisely, the optimal value of α is equal to the minimum of J.

Theorem 4: The controller $G(s)$ with the state-space matrices (A_c^*, B_c^*, C_c) is stable, satisfies the design specifications $\mathcal{D}$ and minimizes the performance index J.

Proof: Given a controllable, observable, and stable controller $G(s)$ with the matrices (A_c, B_c, C_c), one can write

$$g(t) = C_c e^{A_c t} B_c, \quad \forall t \geq 0. \tag{39}$$

Hence,

$$g''(t) = C_c A_c e^{A_c t} A_c B_c, \quad \forall t \geq 0. \tag{40}$$

As a result, the performance index J can be obtained as

$$J = \int_0^\infty C_c A_c e^{A_c t} A_c B_c B_c^T A_c^T e^{A_c^T t} A_c^T C_c^T \, dt, \tag{41}$$

or equivalently $J = C_c A_c P A_c^T C_c^T$, where P is the unique solution of the Lyapunov equation

$$A_c P + P A_c^T + A_c B_c B_c^T A_c^T = 0. \tag{42}$$

Now, it can be proved that J is equal to the minimum (infimum) of a scalar α subject to the constraints

$$A_c P + P A_c^T + A_c B_c B_c^T A_c^T \prec 0, \quad C_c A_c P A_c^T C_c^T \prec \alpha, \quad -P \prec 0. \tag{43}$$

The Schur complement formula can be used twice to deduce that the above constraints are identical to the ones given in Optimization 1. ■

Theorem 4 states that Optimization 1 yields a controller $G(s)$ whose digital implementation is near-optimal. Nevertheless, regardless of the first constraint in this optimization corresponding to the design specifications, the other two constraints are nonlinear in the variables A_c, B_c and P. This is a common issue in many control problems for designing a fixed-order controller [15]. However, it can be observed that if either A_c or B_c, P are fixed, the second and third constraints in Optimization 1 turn into linear matrix inequalities. Hence, one can start from a stable controller and solve this optimization problem iteratively by fixing A_c and B_c, P alternatively until a local solution is found (see [31] and the references therein for similar algorithms).

Due to the design specifications being rather general, the complexity of Optimization 1 is not clear. As a second method, let the design specification $\mathcal{D}$ be only the stability of the closed-loop system, the order of the controller $G(s)$ being found be unknown, and the controller be biproper if necessary ($G(s)$ was strictly proper in the previous method). Consider a single-input single-output, stable, low-pass filter $\bar{\mathcal{F}}(s)$ whose relative degree is greater than 2. Denote the impulse response of $\bar{\mathcal{F}}(s)G(s)$ as $\bar{g}(t)$. Define a new performance index $\bar{J}$ as

$$\bar{J} := \int_0^\infty \|\bar{g}''(t)\|_2^2 \ dt. \tag{44}$$

Unlike the performance index J, the new index $\bar{J}$ operates on the filtered impulse response to remove any possible jitter that makes the second derivative of $g(t)$ unnecessarily high but does not affect the piecewise linear approximation of $g(t)$ noticeably. It will be shown in the sequel that although finding a stable, stabilizing controller $G(s)$ minimizing $\bar{J}$ may not be a convex problem, it can be cast as a well-known problem for which there exist different sufficient conditions in the convex form. Note that the main reason why $\bar{J}$ is considered here instead of J is that the introduction of the filter $\bar{\mathcal{F}}(s)$ simplifies the corresponding optimization problem and converts it to a well-studied problem. Since $\bar{\mathcal{F}}(s)$ is stable with a relative degree greater than 2, the transfer function $s^2\bar{\mathcal{F}}(s)$ has a state-space realization as $(A_f, B_f, C_f, 0)$, where A_f is a Hurwitz matrix. Design two matrix gains $L_1 \in \Re^{m\times n}$ and $L_2 \in \Re^{n\times r}$ such that the matrices $A + BL_1$ and $A + L_2C$ become both Hurwitz. Consider the system

$$\begin{aligned}
\dot{\mathbf{x}}_f(t) &= \begin{bmatrix} A + BL_1 + L_2C & 0 \\ B_fL_1 & A_f \end{bmatrix} \mathbf{x}_f(t) + \begin{bmatrix} -L_2 \\ 0 \end{bmatrix} y(t) + \begin{bmatrix} B \\ B_f \end{bmatrix} \zeta_1(t), \\
u_f(t) &= \begin{bmatrix} 0 & C_f \end{bmatrix} \mathbf{x}_f(t), \\
\zeta_2(t) &= \begin{bmatrix} -C_2 & 0 \end{bmatrix} \mathbf{x}_f(t) + y(t),
\end{aligned}$$

with the inputs $y(t), \zeta_1(t)$ and the outputs $u_f(t), \zeta_2(t)$. Find a finite-dimensional, stable, LTI controller from $\zeta_2(t)$ to $\zeta_1(t)$ to minimize the 2-norm of the transfer function from $y(t)$ to $u_f(t)$ in above control system, and denote it with $Q^*(s)$. It can be observed that finding $Q^*(s)$ amounts to a standard H_2 *strong stabilization* problem. Note that the closely related problems of H_2 *strong stabilization* and H_∞ *strong stabilization* have been thoroughly investigated in several works [19], [20], [10], [23].

Theorem 5: Let $G(s)$ be taken as the controller given in Figure 3(c) with $Q(s)$ equal to $Q^*(s)$ and $M(s)$ with the control law

$$\begin{aligned}
\dot{\mathbf{x}}(t) &= (A + BL_1 + L_2C)\mathbf{x}(t) - L_2y(t) + B\zeta_1(t), \\
u(t) &= L_1\mathbf{x}(t) + \zeta_1(t), \\
\zeta_2(t) &= -C\mathbf{x}(t) + y(t).
\end{aligned} \tag{45}$$

This choice of the controller $G(s)$ is stable, stabilizes the system $\mathcal{S}$ and minimizes the performance index $\bar{J}$.

Sketch of Proof: It follows from the linear fractional transformation that every stabilizing, finite-dimensional, LTI controller $G(s)$ can be decomposed into the form given in Figure 3(c) for some stable controller $Q(s)$ [12]. Augment the controller $G(s)$ with the modified filter $s^2\bar{\mathcal{F}}(s)$ by connecting the filter to the output of the controller $G(s)$, and denote the output of the augmented system with $u_f(t)$. It can be observed that the impulse response of the augmented system is equal to $\bar{g}''(t)$. This result is due to the facts that $s^2\bar{\mathcal{F}}(s)$ is strictly proper and that the term s^2 acts as a double differentiator. Now, it follows from Parseval's theorem that $\int_0^\infty \|\bar{g}''(t)\|_2^2 \ dt$ is equal to the 2-norm of the transfer function of the augmented system from $y(t)$ to $u_f(t)$. The proof is completed by noting that the model (45) under the controller $Q(s)$ from $\zeta_2(t)$ to $\zeta_1(t)$ is a state-space representation of this augmented system. ■

Theorem 5 states that finding a stable, stabilizing controller $G(s)$ with a near-optimal digital implementation amounts to the well-studied problem of *stable* H_2 *optimal control* (or H_2 *strong stabilization*). As an alternative to the index $\bar{J}$, one can argue that the minimization of the simple index $\int_0^\infty \|g(t)\|_2^2 \ dt$ (with no differentiation involved) also leads to a near-optimal $g(t)$. This minimization can be converted to finding a stable H_2 optimal controller $Q(s)$ for the configuration given in Figure 3(c). To summarize, two indices J and $\bar{J}$ were introduced in this section to design a controller $G(s)$ with a smooth impulse response. Once the function $g(t)$ is obtained using either of the above-mentioned methods, the technique spelled out in Subsection II-E can be used to find a minimal set of delays $\{\tau_1, \tau_2, ..., \tau_p\}$.

IV. Sampled-data Controller Design

Consider the LTI system $\mathcal{S}$ given in (1) and, with no loss of generality, assume that $D = 0$. There are numerous applications for which it is desirable to control this system using a digital controller, e.g. a micro-controller. A conventional digital-control scheme, referred to as *sampled-data control system*, is depicted in Figure 4, which consists of the following components:

- *Sampler:* This part is intended to sample the output of the system $\mathcal{S}$ at a pre-specified frequency f_0.
- *Digital Controller:* This controller processes the digital signal provided by the sampler.

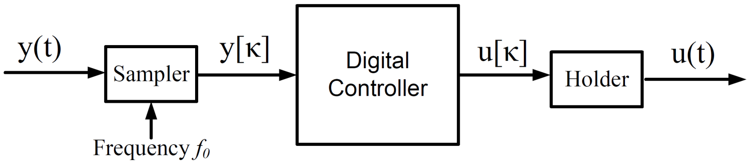


Fig. 4. This figure illustrates a conventional sampled-data control system.

- *Hold circuit:* This part generates the input of the system $\mathcal{S}$ by converting the discrete-time output of the digital controller to a continuous-time signal.

Unlike an ideal sampler, there exist different types of ideal hold circuits such as a zero-order hold or a first-order hold. After choosing a sampling frequency and a proper type of hold circuit, the main challenge is to design a digital controller, denoted by G_d, for the sampled-data control system in such a way that the closed-loop system satisfies certain design specifications. Three methods have been long studied in the literature for this purpose:

i) Design a controller G_d for the discrete-time equivalent model of the system $\mathcal{S}$.

ii) Design G_d by first finding a continuous-time (finite-dimensional) controller G for the system $\mathcal{S}$ and then discretizing it.

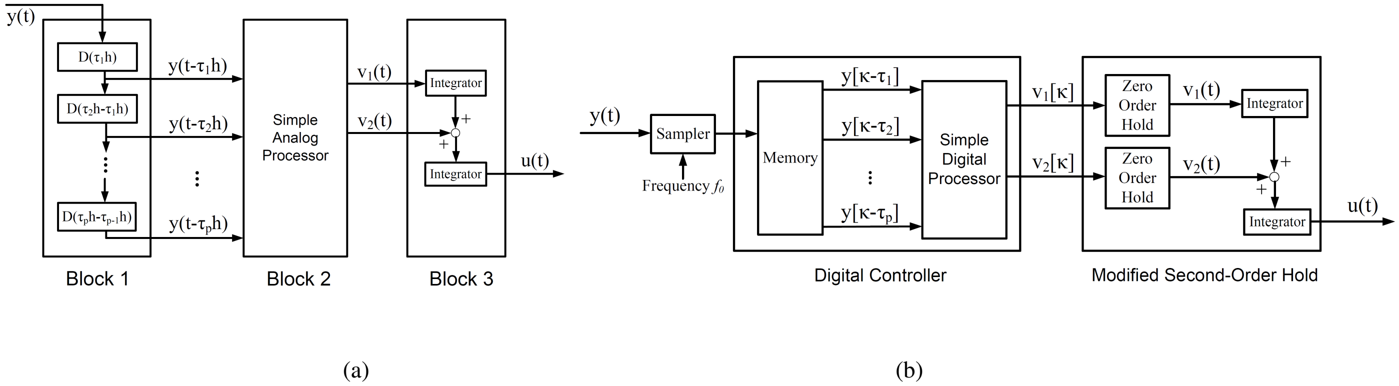
iii) Design G_d directly for the time-varying closed-loop system.

With the ongoing technological advances, it is now possible to sample the outputs of many real-world systems at a very high rate f_0, on the order of several kilohertz. Although a high sampling rate is desirable for collecting more information from the continuous-time output $y(t)$, a sampled-data controller designed using the aforementioned techniques may suffer from some robustness issues for a relatively large f_0. To illustrate this fact, consider method (ii) and assume that the hold circuit of the sampled-data control system is a zero-order hold. Let G be a given finite-dimensional, continuous-time controller designed for the system $\mathcal{S}$, with the state-space representation

$$\begin{aligned}\dot{x}_c(t) &= A_c x_c(t) + B_c y(t),\\ u(t) &= C_c x_c(t) + D_c y(t).\end{aligned} \tag{46}$$

The digital controller G_d can be taken as the discrete-time equivalent model of G obtained using the step-invariant method, which turns out to be

$$\begin{aligned}x_d[\kappa+1] &= A_d x_d[\kappa] + B_d y[\kappa],\\ u[\kappa] &= C_d x_d[\kappa] + D_d y[\kappa], \quad \kappa = 0, 1, 2, ...,\end{aligned} \tag{47}$$



(a)

(b)

Fig. 5. (a) An analog implementation of the controller $\hat{G}(s)$; (b) The hybrid controller $\hat{G}_d$ associated with the continuous-time controller $\hat{G}$.

where

$$A_d = e^{hA_c}, \quad B_d = \int_0^h e^{tA_c} dt \ B_c, \quad C_d = C_c, \quad D_d = D_c, \quad h = \frac{1}{f_0}. \tag{48}$$

(Instead of the step-invariant method, one can alternatively use other existing methods such as the Tustin approximation.) Observe that as the sampling period h goes to 0, A_d and B_d converge to I and 0, respectively. This implies that the convergence is independent of the values of the matrices A_c and B_c, which makes the digital controller G_d extremely fragile and sensitive to measurement and numerical round-off errors. By denoting the order of the controller G with n_c, it can be argued that this undesirable sensitivity is a consequence of generating the input $u[\kappa]$ in terms of the last $n_c + 1$ samples of the output, i.e. $y[\kappa], y[\kappa - 1], ..., y[\kappa - n_c]$. More precisely, as h goes to zero, all these samples become indistinguishable and, therefore, performing numerical computations on them leads to a poor implementation. This observation is valid for the aforementioned methods (i) and (iii) as well. A question arises as to whether it is possible to generate $u[\tau]$ in terms of some sufficiently distant samples, namely $y[\kappa{-}\tau_1], y[\kappa{-}\tau_2], ..., y[\kappa{-}\tau_p]$ for some disparate numbers $\tau_1, \tau_2, ..., \tau_p$, and deploy a new type of (fast) hold circuit so that the resulting digital controller becomes satisfactorily robust and easily implementable (note that the idea of using distant output samples is not equivalent to slow sampling). The results developed in the preceding section will be exploited here to address this problem under the assumption that both sampler and hold circuit operate at the same high frequency. The generalization to the case when the hold circuit (or actuator) operates at a slower frequency (or even aperiodically) is straightforward.

Given an LTI continuous-time controller $G(s)$ satisfying some prescribed design specifications, our goal is to implement this controller in the form of the configuration given in Figure 4 with a high sampling rate f_0. Assume for now that $G(s)$ is stable and single-input single-output. These assumptions will be removed later in this subsection. In addition, with no loss of generality, suppose that D_c is equal to 0 (because this term corresponds to a direct static feedback from $y(t)$ to $u(t)$ that can be easily implemented). The method developed in the previous section (in particular, the discretization technique discussed in Subsection II-E) can be used to approximate the time-domain signal $g(t)$ by a piecewise linear signal $\hat{g}(t)$ with a finite number of breakpoints all belonging to the set $\{0, h, 2h, ...\}$. Recall that the Laplace transform of $\hat{g}(t)$ can be written as

$$\hat{G}(s) = \sum_{i=1}^{p} \left(\frac{\alpha_i}{s^2} + \frac{\beta_i}{s} \right) e^{-\tau_i h s}, \tag{49}$$

for some scalars $\alpha_1, ..., \alpha_p, \beta_1, ..., \beta_p$, where $\tau_1 h, ..., \tau_p h$ denote the breakpoints of the signal $\hat{g}(t)$ in an ascending order. An analog implementation of the controller $\hat{G}(s)$ is visualized in Figure 5(a), which consists of three blocks:

- Block 1 delays the incoming signal $y(t)$ by $\tau_1 h, ..., \tau_p h$ seconds.
- Block 2 performs basic math operations to generate the signals $v_1(t) := \sum_{i=1}^{p} \alpha_i y(t - \tau_i h)$ and $v_2(t) := \sum_{i=1}^{p} \beta_i y(t - \tau_i h)$.
- Block 3 employs two integrators to generate $u(t)$ from $v_1(t)$ and $v_2(t)$.

Define $\hat{G}_d$ as a hybrid controller with the configuration depicted in Figure 5(b) corresponding to the continuous-time controller $\hat{G}$. Notice that $\hat{G}_d$ is obtained from the particular configuration of $\hat{G}$ given in Figure 5(a) using the following steps:

- Block 1 is replaced by an ideal sampler with the sampling frequency f_0.
- Block 2 is substituted by a memory capable of storing the last $\tau_p + 1$ samples of $y(t)$ and a digital processor for computing $v_1[\kappa] := \sum_{i=1}^{p} \alpha_i y[\kappa - \tau_i]$ and $v_2[\kappa] := \sum_{i=1}^{p} \beta_i y[\kappa - \tau_i]$.
- Block 3 is replaced by two zero-order holds and two integrators. This resulting block can be regarded as a "modified second-order hold" because of its analogy to a standard second-order hold that consists of a conventional digital-to-analog converter and two integrators (analog circuits).

The hybrid controller $\hat{G}_d$ introduced above is indeed a sampled-data controller with an ideal sampler and a modified second-order hold. Recall that the parameters A_d and B_d of a controller G_d obtained using a conventional discretization method converge to I and 0 as h tends to zero, which makes the controller sensitive to measurement and computational errors. In contrast, the correlation between the parameters $\alpha_1, ..., \alpha_p, \beta_1, ..., \beta_p$ of the controller $\hat{G}_d$ and the sampling period h is minimal in the sense that the precision of these parameters need not be increased as h goes to zero. Indeed, reducing h mainly affects the memory size, rather than the foregoing coefficients. This key property makes the hybrid controller $\hat{G}_d$ suitable for fast-sampling applications.

We wish to study the error resulting from implementing the continuous-time controller $\hat{G}$ as the hybrid controller $\hat{G}_d$. To this end, note that although $\hat{G}$ is time-invariant, its counterpart $\hat{G}_d$ is time-varying. In order to bypass the time-varying nature of this hybrid controller, since the system $\mathcal{S}$ acts as a low-pass filter (due to being strictly proper) and the sampling frequency f_0 is relatively high, it is reasonable to assume that high-frequency harmonics of the output signal $y(t)$ in the system $\mathcal{S}$ under $\hat{G}$ or $\hat{G}_d$ are negligible. Hence, assume that the output of the system $\mathcal{S}$ goes through an ideal low-pass filter $\mathcal{F}$ with the cut-off frequency $\omega_0 := \frac{2\pi}{h}$ before being processed by the controller. Let $\mathcal{F} \circ \hat{G}$ and $\mathcal{F} \circ \hat{G}_d$ denote the cascades of the filter $\mathcal{F}$ with the controllers $\hat{G}$ and $\hat{G}_d$, respectively.

Theorem 6: The hybrid controller $\mathcal{F} \circ \hat{G}_d$ is a linear time-invariant system with the transfer function

$$\mathcal{F} \circ \hat{G}_d(j\omega) = \begin{cases} \mathcal{F} \circ \hat{G}(j\omega) \cdot \left(e^{-j\omega\frac{h}{2}} \frac{\sin\left(\omega\frac{h}{2}\right)}{\omega\frac{h}{2}} \right) & \omega \in [-\omega_0, \omega_0] \\ \mathcal{F} \circ \hat{G}(j\omega) & \text{otherwise.} \end{cases} \tag{50}$$

Proof: In the cascade controller $\mathcal{F} \circ \hat{G}_d$, let $y_f(t)$, $y(t)$ and $u(t)$ denote the incoming signal of $\mathcal{F}$, the incoming signal of $\hat{G}_d$ and the output of $\hat{G}_d$, respectively. Due to the presence of the filter $\mathcal{F}$, the relation $\mathcal{F} \circ \hat{G}_d(j\omega) = \mathcal{F} \circ \hat{G}(j\omega) = 0$ holds if $\omega \notin [-\omega_0, \omega_0]$. Now, consider a frequency $\omega \in [-\omega_0, \omega_0]$. It can be verified that (see [12], Chapter 3)

$$\begin{aligned} Y\left[e^{-j\omega h}\right] &= \frac{1}{h}\left(\sum_{k=-\infty}^{\infty} Y(j\omega + jk\omega_0)\right) = \frac{1}{h} Y_f(j\omega), \\ V_1[z] &= \left(\sum_{i=1}^{p} \alpha_i z^{-\tau_i}\right) Y[z], \quad V_1(j\omega) = h\left(e^{-j\omega\frac{h}{2}} \frac{\sin\left(\omega\frac{h}{2}\right)}{\omega\frac{h}{2}}\right) V_1\left[e^{-j\omega h}\right]. \end{aligned} \tag{51}$$

Thus,

$$V_1(j\omega) = \left(e^{-j\omega\frac{h}{2}} \frac{\sin\left(\omega\frac{h}{2}\right)}{\omega\frac{h}{2}}\right) \left(\sum_{i=1}^{p} \alpha_i e^{-j\omega h\tau_i}\right) Y_f(j\omega). \tag{52}$$

Similarly,

$$V_2(j\omega) = \left(e^{-j\omega\frac{h}{2}} \frac{\sin\left(\omega\frac{h}{2}\right)}{\omega\frac{h}{2}} \right) \left(\sum_{i=1}^{p} \beta_i e^{-j\omega h\tau_i} \right) Y_f(j\omega). \tag{53}$$

The proof follows immediately from the equations (52) and (53). ■

As pointed out earlier, the approximating controller $\hat{G}$ can be arbitrarily close to the original controller G. On the other hand, Theorem 6 states that the hybrid controller $\hat{G}_d$ behaves differently from its continuous-time counterpart $\hat{G}$ by a factor $e^{-j\omega\frac{h}{2}}\frac{\sin\omega\frac{h}{2}}{\omega\frac{h}{2}}$ in the Fourier domain if its incoming signal has no harmonics at frequencies greater than ω_0. Notice that as h goes to 0, the real-valued factor $\frac{\sin\omega\frac{h}{2}}{\omega\frac{h}{2}}$ tends to 1 and so does the complex-valued factor $e^{-j\omega\frac{h}{2}}$. As a result, $\hat{G}_d$ is a digital implementation of the original controller G. In order to mitigate the effect of the discretization error $e^{-j\omega\frac{h}{2}}\frac{\sin\omega\frac{h}{2}}{\omega\frac{h}{2}}$, let the controller $\hat{G}_d$ be manipulated so that its discrepancy with the original controller $\hat{G}$ becomes only a multiplicative real-valued factor $\frac{\sin\omega\frac{h}{2}}{\omega\frac{h}{2}}$ (this reduces the delay between the outputs of $\hat{G}_d$ and $\hat{G}$). To this end, the following procedure can be taken.

Procedure 1:

- Approximate $g(t)$ with a piecewise linear function $\tilde{g}_d(t)$ in such a way that its breakpoints lie in the set $\{\frac{h}{2}, \frac{3h}{2}, \frac{5h}{2}, ...\}$, as opposed to $\{0, h, 2h, ...\}$.
- Find the Laplace transform of $\tilde{g}_d(t)$ and write it in the form of

$$\sum_{i=1}^{p} \left(\frac{\alpha_i}{s^2} + \frac{\beta_i}{s} \right) e^{-\left(\tau_i h + \frac{h}{2}\right)s}. \tag{54}$$

- Define $\tilde{G}_d$ to be the hybrid controller depicted in Figure 5(b), where

$$v_1[\kappa] := \sum_{i=1}^{p} \alpha_i y[\kappa - \tau_i], \quad v_2[\kappa] := \sum_{i=1}^{p} \beta_i y[\kappa - \tau_i]. \tag{55}$$

- The system $\mathcal{F} \circ \tilde{G}_d$ is LTI with the transfer function

$$\mathcal{F} \circ \tilde{G}_d(j\omega) = \begin{cases} \mathcal{F} \circ \tilde{G}(j\omega) \cdot \left(\frac{\sin\left(\omega\frac{h}{2}\right)}{\omega\frac{h}{2}} \right) & \omega \in [-\omega_0, \omega_0] \\ \mathcal{F} \circ \tilde{G}(j\omega) & \text{otherwise} \end{cases} \tag{56}$$

The hybrid controller $\tilde{G}_d$ introduced in Procedure 1 is another digital implementation of G which, in comparison to the hybrid controller $\hat{G}_d$, is expected to have less discrepancy with respect to the target controller $\hat{G}$.

Th results developed so far are based on the assumption that the initial controller $G(s)$ is stable. If this stabilizing controller is not stable itself, the idea spelled out in Subsection II-G

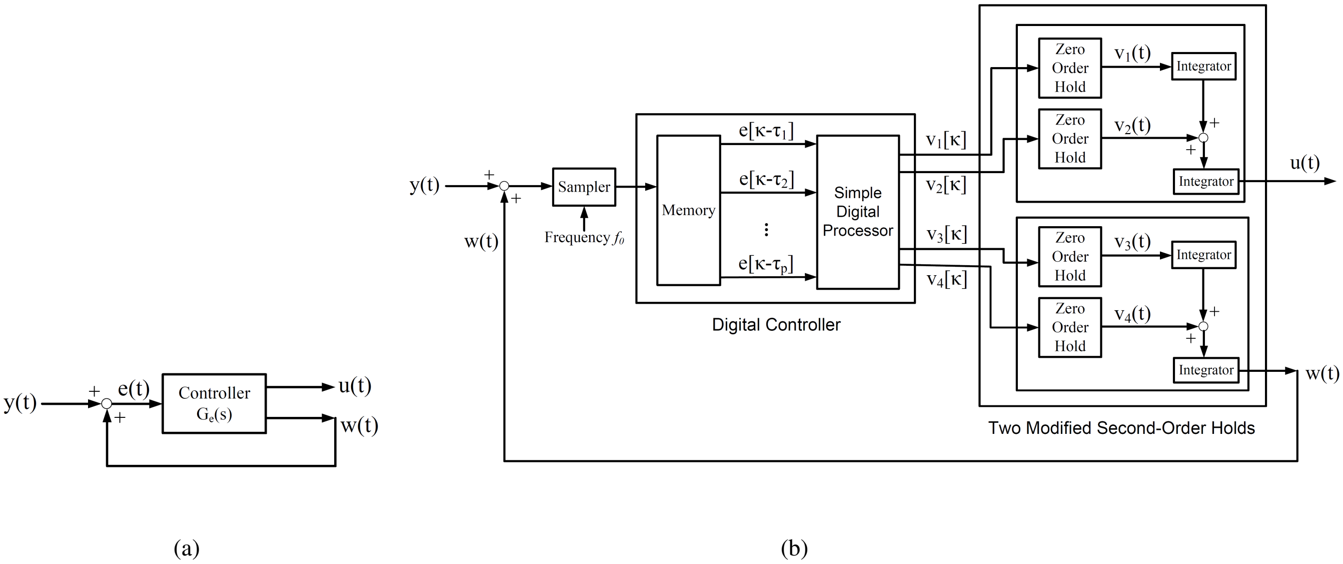
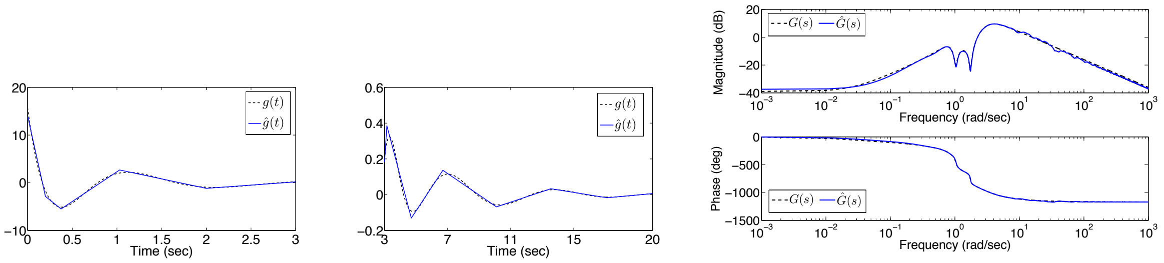


Fig. 6. (a): An equivalent implementation of an unstable controller $G(s)$; (b): the hybrid controller $\hat{G}_d$ associated with an unstable continuous-time controller G.

can be used to reconfigure the controller as the feedback form given in Figure 6(a) whose forward path is stable. Since the transfer functions from "$e(t)$ to $u(t)$" and "$e(t)$ to $w(t)$" are stable in the new configuration, they can be implemented via their hybrid counterparts explained earlier. Hence, the unstable controller $G(s)$ can be implemented in the sampled-data control scheme depicted in Figure 6(b), which consists of an ideal sampler, a digital controller, two modified second-order holds and a unity feedback. Note that if it turns out to be impossible in practice to add the signals $y(t)$ and $w(t)$ before sampling, as suggested in Figure 6(b), the output of the system, i.e. $y(t)$, can be first sampled and then added to the samples of the signal $w(t)$.

To generalize the results of this section to multi-input multi-output controllers, it suffices to follow the line of arguments discussed in Subsection II-H. The details are omitted here for brevity.

Remark 2: We proposed a new hybrid implementation of a given continuous-time controller in this section, whose real-time complexity (i.e. the processing time required by its processor) is contingent upon the number of delays $\tau_1, \tau_2, ..., \tau_p$. Given some design specifications, a question arises as what continuous-time controller $G(s)$ satisfies the design objectives and, in addition, its hybrid counterpart needs the least number of delays. To answer this question, recall that an impulse response $g(t)$ must be found whose satisfactory piecewise linear approximation with breakpoints belonging to the set $\{0, h, 2h, ...\}$ requires the least number of corners. Notice that



(a) (b) (c)

Fig. 7. (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 $[3, 20]$; (c): the Bode plots of the controllers $G(s)$ and $\hat{G}(s)$ for the example given in Section VI.

as long as the sampling period is sufficiently small, the set $\{0, h, 2h, ...\}$ can be estimated by the real set $\Re^+$, which converts the underlying problem to finding the smoothest impulse response $g(t)$ satisfying the design specifications. This problem has been already tackled in Section III.

Remark 3: The hybrid configuration proposed here can be modified slightly to handle sampling irregularities, e.g. non-uniform sampling (deterministic jitter) and missing samples. Indeed, it suffices to pre-process the samples by an interpolation algorithm to estimate the true values of the samples at the desired times $\tau_1 h, ..., \tau_p h$. The effectiveness of this approach follows from two facts: (i) there are a large number of non-uniform samples for interpolation due to the sampling period h being relatively small, (ii) since the delay terms $\tau_1 h, ..., \tau_p h$ are designed to be sufficiently distant, a sampling irregularity caused by a small perturbation in these values does not alter the output of the digital controller noticeably.

V. Numerical Example

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

$$P_1(s) := 0.0064s^5 + 0.0024s^4 + 0.071s^3 + s^2 + 0.1045s + 1,$$

$$P_2(s) := s^8 + 0.161s^7 + 6s^6 + 0.582s^5 + 9.984s^4 + 0.407s^3 + 3.9822s^2 + 0.08s + 0.08.$$

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 [10], [23] 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

$$G_1(s) := 15.76s^7 - 3.896s^6 + 60.68s^5 - 9.68s^4 + 34.99s^3 - 2.064s^2 - 12.39s + 0.2986,$$

$$G_2(s) := s^8 + 8.684s^7 + 41.18s^6 + 115.3s^5 + 208.8s^4 + 250.9s^3 + 197.9s^2 + 111.1s + 26.64.$$

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 Figures 7(a) and (b), 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 $\boldsymbol{\tau}$ was obtained as

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_1 & \tau_2 & \cdots & \tau_{12} \end{bmatrix} = \begin{bmatrix} 0 & 0.2 & 0.37 & 1.03 & 2 & 3.15 & 4.7 & 6.7 & 10.1 & 13.55 & 17.11 & 20 \end{bmatrix}.$$

The method proposed in Subsection II-E can be used to find the best piecewise linear approximation of $g(t)$ with its knots given by the vector $\boldsymbol{\tau}$. 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 Figures 7(a) and 7(b). The controller $\hat{G}(s)$ turns out to be

$$\hat{G}(s) = \sum_{i=1}^{p} \left(\frac{\alpha_i}{s^2} + \frac{\beta_i}{s} \right) e^{-\tau_i s},$$

where

$$\begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_{12} \end{bmatrix} = \Big[\, -84.1100 \quad 68.3058 \quad 0.4660 \quad -0.1936 \quad 28.2217 \quad -16.3791 \quad 5.3132$$
$$-1.6841 \quad 0.0895 \quad -0.0436 \quad 0.0223 \quad -0.0081 \,\Big],$$

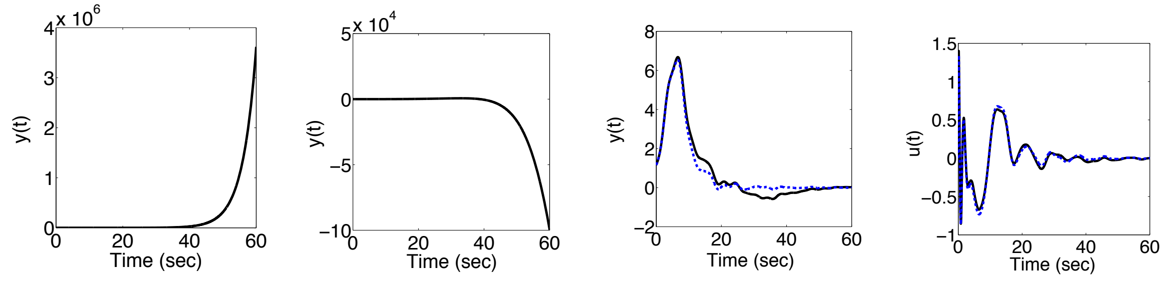
$$\begin{bmatrix} \beta_1 & \beta_2 & \cdots & \beta_{12} \end{bmatrix} = \begin{bmatrix} 13.9861 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.0061 \end{bmatrix}.$$

The Bode plots of the controllers $G(s)$ and $\hat{G}(s)$ are compared in Figure 7(c), which illustrate that $\hat{G}(s)$ is a good approximation of $G(s)$. Let $\tilde{G}(s)$ denote a 6th order reduced model of $G(s)$ obtained using the balanced model-reduction technique. To compare $\hat{G}(s)$ with $\tilde{G}(s)$, notice that:

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

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

Now, assume that the objective is to implement the optimal controller $G(s)$ in a sampled-data control configuration with the sampling frequency $f_0 = 100$Hz under the assumption that the precision of the parameters of the digital controller is confined to four fractional digits. This assumption is made to ensure that the digital processor performs a reasonable truncation before



(a)

(b)

(c)

(d)

Fig. 8. (a): The output of the system $\mathcal{S}$ under a conventional sampled-data controller with $f_0 = 100$Hz; (b): The output of the system $\mathcal{S}$ under a conventional sampled-data controller with $f_0 = 10$Hz; (c): The output of the system $\mathcal{S}$ under $G(s)$ (dotted curve) and $\hat{G}_d$ with $f_0 = 100$Hz (solid curve); (d): The input of the system $\mathcal{S}$ under $G(s)$ (dotted curve) and $\hat{G}_d$ with $f_0 = 100$Hz (solid curve).

any computation. For this purpose, let the initial state of the system be the vector $[1\ 1\ \cdots\ 1]$. As the first approach, we convert the controller $G(s)$ to a conventional sampled-data controller using the step-invariant method and then truncate the parameters of the digital controller to 4 significant fractional digits. The output of the system is plotted in Figure 8(a) to demonstrate that the closed-loop system is unstable. Note that this instability is only the result of reducing the infinite precision to four digits. If the sampling frequency is reduced to 10Hz, the closed-loop system will be still unstable, as illustrated in Figure 8(b). In contrast, the controller $G(s)$ can be implemented in the hybrid configuration $\hat{G}_d$ using the continuous-time delay-based controller $\hat{G}(s)$. The output and input of the system are plotted under both the continuous-time controller $G(s)$ and its hybrid implementation $\hat{G}_d$ in Figures 8(c) and 8(d). These figures clearly demonstrate that the proposed hybrid controller performs similarly to the original LQG controller and that the high sampling frequency $f_0 = 100$Hz does not cause a robustness issue.

VI. Conclusions

Motivated in part 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 continuous-time systems whose control using conventional techniques needs many integrators. First, we showed that every stabilizing continuous-time linear time-invariant (LTI) 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 two. Finding the optimal number of delay blocks, finding the optimal values of the delays, and studying the robustness of the designed controller can also be treated within the framework presented. Finally, we considered the problem of designing a continuous-time LTI controller which not only satisfies prescribed design specifications, but also has the least complex delay-based implementation.

An application of our delay-based controller design is in the sampled-data control of continuous-time systems in presence of sampling jitter and/or a high sampling frequency. Indeed, since a conventional sampled-data controller with a relatively high sampling frequency needs high-precision computation to cope with a robustness issue and sampling jitter also worsens the situation, we proposed a new type of digital-control scheme that does not suffer from these issues. We showed that every continuous-time stabilizing (LTI) controller can be implemented in a hybrid configuration composed of an ideal sampler, a digital controller, a number of modified second-order holds and possibly a unity feedback. An advantage of this hybrid controller is that increasing the sampling frequency mainly affects the memory size of the controller, as opposed to its parameters. This property makes the controller robust to measurement and computational errors at high frequencies, and hence obviates the necessity of increasing the processing precision.

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] 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.

[2] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, *Molecular biology of the cell*, Fourth edition, Garland Science, 2002.

[3] E. Andrianantoandro, S. Basu, D. K. Karig, and R. Weiss, "Synthetic biology: new engineering rules for an emerging discipline," *Molecular Systems Biology*, vol. 2, 2006.

[4] K. J. Åström and B. Wittenmark, "Computer controlled systems: theory and design," *Prentice-Hall*, Englewood Cliffs, 1997.

[5] A. Babakhani, X. Guan, A. Komijani, A. Natarajan, and A. Hajimiri, “A77-GHz phased array transceiver with on-chip antennas in silicon: receiver and antennas,” *IEEE Journal of Solid-State Circuits*, vol. 41, no. 12, pp. 2795-806, 2006.

[6] B. Bamieh, J. Pearson, B. Francis, and A. Tannenbaum, “A lifting technique for linear periodic systems,” *Systems & Control Letters*, vol. 17, no. 2, pp. 79-88, 1991.

[7] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan, “Linear matrix inequalities in system and control theory,” *SIAM*, 1994.

[8] 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.

[9] R. W. Brockett, “Reduced complexity control systems, in *Plenary Papers, Milestone Reports, & Slected Survey Papers*, Myung Jin Chung and Pradeep Misra, Eds., 17th IFAC World Congress, 2008.

[10] D. U. Campos-Delgado and K. Zhou, “H_∞ strong stabilization,” *IEEE Transactions on Automatic Control*, vol. 46, no. 12, pp. 1968-1972, 2001.

[11] 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.

[12] T. Chen and B. A. Francis, “Optimal sampled-data control systems,” *Springer*, 1995.

[13] J. Chen, S. Hara, L. Qiu, and R. Middleton, “Best achievable tracking performance in sampled-data systems via LTI controllers,” *IEEE Transactions on Automatic Control*, vol. 53, no. 11, pp. 2467-2479, 2008.

[14] S. Coombes and C. Laing, “Delays in activity-based neural networks,” *Philosophical Transactions of The Royal Society A*, vol. 367, no, 1891, pp. 1117-1129, 2009.

[15] G. E. Dullerud and F. Paganini, *A course in robust control theory: a convex approach, Texts in Applied Mathematics, Springer*, 2005.

[16] M. B. Elowitz and S. Leibler, “A synthetic oscillatory network of transcriptional regulators,” *Nature*, vol. 403, no. 20, pp. 335-338, 2000.

[17] E. Franco and R. M. Murray, “Design and performance of in vitro transcription rate regulatory circuit,” in *Proceedings of the 47th IEEE Conference on Decision and Control*, 2008.

[18] H. Fujioka, “Stability analysis for a class of networked/embedded control systems: A discrete-time approach,” in *Proceedings of 2008 American Control Conference*, pp. 4997-5002, 2008.

[19] C. Ganesh and J. B. Pearson, “H_2-optimization with stable controllers,” *Automatica*, vol. 25, no. 4, pp. 629-634, 1989.

[20] C. Ganesh and J. B. Pearson, “A parametric optimization approach to H_∞ and H_2 strong stabilization,” *Automatica*, vol. 39, no. 7, pp. 1205-1211, 2003.

[21] 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.

[22] K. Gu, V. Kharitonov, and J. Chen, *Stability of time-delay systems*, Birkhäuser, 2003.

[23] S. Gumussoy and H. Ozbay, “Remarks on strong stabilization and stable H_∞ controller design,” *IEEE Transactions on Automatic Control*, vol. 50, no. 12, pp. 2083-2087, 2005.

[24] T. Hagiwara and M. Araki, “FR-operator approach to the H_2 analysis and synthesis of sampled-data systems,” *IEEE Transactions on Automatic Control*, vol. 40, no. 8, pp. 1411-1421, 1995.

[25] 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.

[26] P. Ioannou, B. Fidan, *Adaptive control tutorial*, SIAM, Advances in Design and control, 2006.

[27] E. I. Jury, “Sampled-data control systems,” *Krieger Publishing Co.*, 1977.

[28] T. Katayama, *Subspace methods for system identification*, Springer, 2005.

[29] H. Kopetz, “Real-time systems: design principles for distributed embedded applications,” *Norwell*, MA: Kluwer, 1997.

[30] S. Lall and G. Dullerud, “An LMI solution to the robust synthesis problem for multi-rate sampled-data systems,” *Automatica*, vol. 37, no. 12, pp. 1909-1922, 2001.

[31] J. Lavaei and A. G. Aghdam, “Simultaneous LQ control of a set of LTI systems using constrained generalized sampled-data hold functions,” *Automatica*, vol. 43, no. 2, pp. 274-280, 2007.

[32] M. Lluesma, A. Cervin, P. Balbastre, I. Ripoll, and A. Crespo, “Jitter evaluation of real-time control systems,” in *12th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, Sydney, 2006.

[33] 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.

[34] 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.

[35] R. Murphey and P. M. Pardalos, *Cooperative control and optimization*, Springer, 2002.

[36] S. I. Niculescu, “Delay effects on stability,” In *Lecture notes in control and information sciences*, Berlin: Springer, Vol. 269, 2001.

[37] S. I. Niculescu and K. Gu, *Advances in time-delay systems*, Springer, 2004.

[38] B. Novák and J. J. Tyson, “Design principles of biochemical oscillators,” *Nature Reviews Molecular Cell Biology*, vol. 9, no. 12, pp. 981-991, 2008.

[39] 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.

[40] 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.

[41] S. Mukherji and A. V. Oudenaarden, “Synthetic biology: understanding biological design from synthetic circuits,” *Nature Reviews Genetics*, vol. 10, pp. 859-871, 2009.

[42] 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*, 1996.

[43] N. van de Wouw, P. Naghshtabrizi, M. Cloosterman, and J. Hespanha. “Tracking control for sampled-data systems with uncertain time-varying sampling intervals and delays,” *International Journal of Robust and Nonlinear Control*, vol. 20, no. 4, pp. 387-411, 2009.

[44] V. Zakian, “Control systems design: a new framework,” Springer, 2005.

[45] Q. C. Zhong, *Robust control of time-delay systems*, Springer, 2006.

[46] K. Zhou, J. Doyle, and K. Glover, *Robust and optimal control*, Prentice-Hall, 1996.