

Creating two disjoint stability intervals along the delay axis via controller design: a class of LTI SISO systems

D. Tian¹ · R. Sipahi¹

Received: 26 April 2016 / Revised: 27 July 2016 / Accepted: 6 August 2016 / Published online: 24 August 2016
© Springer-Verlag Berlin Heidelberg 2016

Abstract Frequency domain technique is adapted here together with an optimization scheme to design controllers for a class of linear time-invariant single-input single-output systems (SISO) with a loop delay, such that the closed loop system can be made stable in two distinct pre-selected delay intervals $[0, h_1)$ and (h_2, h_3) , where $h_1 < h_2 < h_3$. This stability-certifying approach is demonstrated over case studies, including an experiment.

Keywords Time-delay systems · Stability · Control design · Optimization

Mathematics Subject Classification 34C25 · 34K20 · 65K10 · 93C23

1 Introduction

In many real-world applications involving sensing, actuation, and decision making, time delays arise naturally. This is mainly due to the time needed to sense and transmit information, and to prepare and execute actuator commands. The presence of delays may cause poor performance, import instability, and pose major limitations on the ability to control [1–4]. In the context of linear time-invariant (LTI) systems, effects of delays can be interpreted based on the behavior of system eigenvalues with respect to the delay parameter. Since delays manipulate the eigenvalues, if the controller is

not properly designed, then these eigenvalues with the influence of delays may bring low damping or instability to the dynamics. To remedy this, one needs to carefully design the controller, incorporating explicitly the delay parameter into the design process.

Control design in the presence of delays is challenging. Some solutions to this problem were proposed based on the Lyapunov stability theory [2] and even beyond LTI systems [5,6], where *delay-range-dependent* stability in the presence of time-varying delays was studied, both in continuous and discrete time. Lyapunov-based approaches are indeed versatile as they can address a large class of control problems; and are attractive as the corresponding problems can be solved efficiently via optimization tools under convexity assumptions, and even for multiple controller gains. On the other hand, conservatism posed on stability conditions and difficulties in designing structured controllers lead one to pursue alternatives especially in the case of LTI systems. One option is to formulate the control problem in frequency domain and take an eigenvalue-based approach. Know-how of control design in the frequency domain however is much limited, mainly characterized by the arising control problem being infinite dimensional [7] and the inherent difficulties in handling multiple design parameters using the system characteristic equation. Many studies along these lines therefore remained to focus on designing few parameters for LTI SISO systems. Some key studies can be found in [8–14].

Frequency domain techniques are however still attractive as they can be used to analyze the stability of LTI systems, without any conservatism. Moreover, these techniques can be used to formulate structured controllers. In this direction, most advancements were made by focusing on the rightmost root of the closed-loop system characteristic equation, and tuning the controller gains accordingly [14–17]. In this effort, the delay in the control loop is given and/or designed together

✉ R. Sipahi
rifat@coe.neu.edu

D. Tian
tian.do@husky.neu.edu

¹ Department of Mechanical and Industrial Engineering,
Northeastern University, Boston, MA 02115, USA

with the controller while optimizing the real part of the rightmost root. Other contributions in the literature include *delay-dependent stability* where by tuning the controller parameters stability can be achieved within a delay margin $\bar{\tau}$, that is, the system is stable for all delays τ satisfying $\tau \in [0, \bar{\tau}]$ [18]. Complementary to this is the well-known *delay independent stability* (DIS) where the system is stable regardless of the amount of delays, i.e., the delay margin is infinite, $\bar{\tau} \rightarrow +\infty$. Existing work in this direction includes testing whether or not a given LTI system with single or multiple delays exhibits DIS property; see [19–21] and the references therein, or designing certain system parameters such that the LTI system can be made delay-independent stable [22,23].

Furthermore, numerous techniques were published for the “analysis” of the delay parameter space. In these work, the objective is to find out, given system characteristic equation, what the stability/instability decomposition of the system is in any parameter space, including delay τ [3]. This analysis may reveal that the system can be stable in more than one delay interval [4,24]. Such a property can be useful in many applications. For example, larger delays falling into the second delay interval can be more favorable from the viewpoint of a system’s disturbance-rejection capability [25]. Moreover, closed-loop systems cannot be always made delay-independent stable, and hence having multiple stable intervals as options for the system to operate is desirable. In other cases, the maximum achievable delay margin of the closed-loop system is upper bounded $\bar{\tau} < \tau^*$ due to open-loop unstable poles [26] and hence closed-loop system stability cannot be guaranteed for $\tau = \tau' > \tau^*$ even if one reconfigures the controller. A remedy to this would be to design multiple stable delay intervals, one of which containing the delay τ' . In the case of open-loop settings, multiple delay intervals are critical when self-excitations are affected by time delays. One notable example is the delay-induced regenerative chatter where the well-known stability lobes are essentially the stability intervals [4,13] corresponding to proper machining settings including machine tool speed, which is inversely proportional to the delay term.

While having a system with multiple stable delay intervals is attractive as reviewed above, most of the existing work remains to focus on the forward problem where system characteristic equation is given, and stability intervals along τ are to be computed. A control design approach based on these techniques cannot certify stability in pre-set delay intervals, but requires brut force re-checking and plotting, which also limits the design to one or two parameters. As a matter of fact, in many design problems, the reverse problem is more relevant, where one needs to reveal the controller gains by which certain stability intervals along τ can be crafted strategically. Such an effort is extremely complicated mathematically, and if properly constructed, would have practicable impacts on the control of LTI systems.

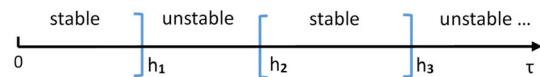


Fig. 1 Desired stability decomposition of the closed loop system in Fig. 2

To the best of our knowledge, this article is the first attempt in the above described design problem with stability “certificates” in pre-set delay intervals. To lay out the main principles of the approach, here we focus on a class of LTI SISO systems for which we create two disjoint stability intervals along τ , as depicted in Fig. 1, by designing multiple coefficients of the corresponding closed-loop characteristic equation. This in essence is achieved by carefully formulating the control design problem in frequency domain and next solving it efficiently using optimization tools.

2 Problem statement

In general, a SISO system with a single delay τ around the control loop, as shown in Fig. 2, has the following characteristic equation:

$$F(s, e^{-s\tau}) = A(s) + B(s)e^{-s\tau} = 0, \tag{1}$$

where $s \in \mathbb{C}$ is the Laplace variable, the delay is non-negative $\tau \geq 0$, and polynomials $A(s)$ and $B(s)$ are nothing but respectively the numerator and denominator of $C(s) \cdot G(s)$. Here, these polynomials are in the general form,

$$\begin{aligned} A(s) &= s^m + a_{m-1}s^{m-1} + \dots + a_1s + a_0 \\ B(s) &= b_n s^n + b_{n-1}s^{n-1} + \dots + b_1s + b_0 \end{aligned} \tag{2}$$

where a_k, b_k are real, $b_n \neq 0$, and $\deg(A(s)) = m \geq \deg(B(s)) = n$ for causality reasons.

When $m > n$, the system is of *retarded* type [4]. This indicates that the highest order derivative in the system dynamics is not affected by any delay terms. The case of $m = n$ corresponds to *neutral* class systems [27,28], which present a number of unique properties related to their spectrum. To keep the presentation compact, in the sequel, we present all the developments first for retarded class systems. Adapting these results for neutral class systems will then easily follow (see Remark 7).

The main problem at hand is that, given $G(s)$, we would like to design $C(s)$ such that the closed-loop system in Fig. 2 with the corresponding characteristic equation (1) is asymptotically stable for all delay values τ in the feedback falling within the pre-determined intervals $\tau \in [0, h_1) \cup (h_2, h_3)$ as depicted in Fig. 1, and marginally stable for delays h_1, h_2 , and h_3 .

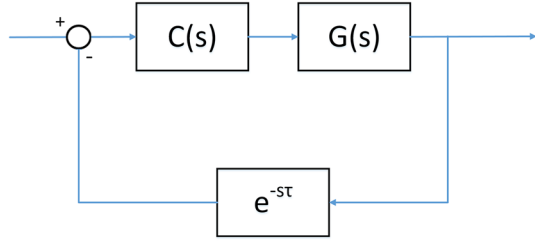


Fig. 2 Model of the SISO LTI system with single delay τ in the feedback

3 Preliminaries

To design a retarded type system to be in stability/instability transition at $\tau = h_1, h_2, h_3$ delay values, we need to study the roots of the characteristic equation (1) that are on the imaginary axis, $s = j\omega$ [4]. We also need to make sure no other crossings occur for delay values interfering the sequence h_1, h_2, h_3 , and that $s = j\omega$ roots tend to cross the imaginary axis toward \mathbb{C}_+ at h_1, h_3 , but toward \mathbb{C}_- at h_2 . Moreover, since for retarded systems, stability of (1) at $\tau = 0$ is preserved as $\tau \rightarrow 0^+$, it suffices to guarantee the stability of $A(s) + B(s)$ such that $A(s) + B(s)e^{-s\tau}$ is also a stable polynomial for $\tau = 0^+$ [4,27]. These knowledge will be the starting point in the following developments.

Note that designing the polynomials $A(s)$ and $B(s)$ to achieve a desired stability decomposition as shown in Fig. 1 is quite complicated, mainly because solving the transcendental equation (1) is impossible. On the other hand, many studies can be followed in order to develop a systematic approach to achieve this nontrivial task (see Sect. 4). Before we present these new results, in the sequel, we first revisit the work of [29], to carry out the mathematics.

3.1 Computing the imaginary crossings

Since from the stability point of view, the most critical roots of (1) are those that are on the imaginary axis, $s = j\omega$, following [29], we can calculate these roots by writing down a new equation by replacing s with $-s$ in (1). That is, $F(s, e^{-s\tau}) = 0$ and $F(-s, e^{s\tau}) = 0$ have the same roots on the imaginary axis. This is as per the symmetricity of system characteristic roots with respect to the real axis of the complex plane. It then follows:

$$\begin{cases} A(s) + B(s)e^{-s\tau} = 0 \\ A(-s) + B(-s)e^{s\tau} = 0 \end{cases} \quad (3)$$

and after the elimination of $e^{-s\tau}$ from (3), one obtains

$$A(s)A(-s) - B(s)B(-s) = 0 \quad (4)$$

which is an even polynomial in s . Then, with $s = j\omega$ and hence $\omega^2 = -s^2$, Eq. (4) can be rearranged into

$$W(\omega^2) = |A(j\omega)|^2 - |B(j\omega)|^2 = 0 \quad (5)$$

whose $\omega^2 > 0$ roots are related to $s = j\omega$ roots of (3) for some $\tau \geq 0$.

Remark 1 The system with the characteristic equation (1) has an imaginary axis root at $s = j\omega^*$ if and only if $W(\omega^{*2}) = 0$ [30]. In the case of multi-input multi-output (MIMO) systems, the characteristic equation may contain commensurate delays. In such cases, imaginary roots of the characteristic equation may constitute only a subset of the roots of the transformed equation $W(\omega^{*2}) = 0$ [31]. One should therefore pay attention to handling such cases when using this elimination technique. Since we do not have commensurate delays in (1), this does not pose any issues in what follows.

Once $W(\omega^2) = 0$ in (5) is solved for $\omega^2 > 0$, the set of solutions, namely, the crossing set, is formed as $\omega = \{\omega_{c1}, \omega_{c2}, \dots, \omega_{cM}\}$, where M is the maximum number of crossings. Since the highest power of $A(s)$ is m , we have that $M \leq m$. Next, the delay values creating an imaginary crossing at $s = j\omega_{cv}$ are calculated. For this, one obtains $e^{j\omega\tau} = -\frac{B(j\omega)}{A(j\omega)}$ from (1), which then yields, for each $\omega = \omega_{cv}$,

$$\tau_{k,v} = \frac{1}{\omega_{cv}} (\angle B(j\omega_{cv}) - \angle A(j\omega_{cv}) + \pi \mp 2k\pi) \quad (6)$$

where $k = 0, 1, \dots, \infty, v = 1, \dots, M$. That is, for each crossing $\omega_{cv}, v = 1, \dots, M$, we have infinitely many delays $\tau_{k,v}$ and the union of these delays $\bigcup_k \bigcup_v \tau_{k,v}$ gives rise to all delays that create imaginary axis crossings of the closed-loop system represented by (1) [24,30].

Remark 2 With the assumption that $\omega_{cv} > 0$ without loss of generality, we then have in general,

$$\omega_{c1} \geq \omega_{c2} \geq \dots \geq \omega_{cM} > 0 \quad (7)$$

where stability of the delay-free system guarantees $\omega_{cM} \neq 0$ for any finite τ .

3.2 Computing the critical delay values

Denote now by $\tilde{\tau}_v$ the minimum positive delay value τ in (6) with respect to each ω_{cv} . This delay can be expressed as:

$$\tilde{\tau}_v = \frac{1}{\omega_{cv}} \begin{cases} \arctan 2(y, x), & \text{if } \arctan 2(y, x) \geq 0; \\ \arctan 2(y, x) + 2\pi, & \text{if } \arctan 2(y, x) < 0. \end{cases} \quad (8)$$

where

$$x = \Re \left(-\frac{B(j\omega_{cv})}{A(j\omega_{cv})} \right), \quad y = \Im \left(-\frac{B(j\omega_{cv})}{A(j\omega_{cv})} \right)$$

Since as per (6), for each ω_{cv} , there exist infinitely many periodically spaced $\tau_{k,v}$ values with the period $\frac{2\pi}{\omega_{cv}}$, then using (8) each $\tau_{k,v}$ in (6) can be compactly formulated as

$$\tau_{k,v} = \tilde{\tau}_v + \frac{2\pi}{\omega_{cv}}k, \quad k = 0, 1, \dots, \infty, \quad v = 1, \dots, M \quad (9)$$

3.3 Computing the root tendency

For each imaginary crossing at ω_{cv} , $v = 1, \dots, M$, there exist infinitely many delay values $\tau_{k,v}$ as per (9). As the delay τ is slightly increased past the critical delay value $\tau_{k,v}$ is the characteristic root on the imaginary axis $s = j\omega_{cv}$ will be perturbed, either into \mathbb{C}_- or \mathbb{C}_+ . Depending on which direction the root moves, the system will have two more, or less, unstable roots. Root tendency (RT) is aimed at calculating this movement via a sensitivity formula at $j\omega_{cv}$ with respect to $\tau_{k,v}$ [24,30,32]. It is defined as

$$RT_{k,v} = \text{sgn} \left[\Re \left(\frac{ds}{d\tau} \Big|_{\tau=\tau_{k,v}, s=j\omega_{cv}} \right) \right], \quad k = 0, 1, \dots, \infty, \quad v = 1, \dots, M \quad (10)$$

It is known that (10) is invariant for all delays corresponding to a specific crossing of (1), see, e.g. [24]. That is, (10) is invariant with respect to the counter k . This implies that given v , $RT_{k,v}$ is fixed for all k . For this reason, we can write RT_v instead of $RT_{k,v}$.

Moreover, when τ increases slightly past $\tau_{k,v}$, if $RT_v = +1$, then the root at $s = j\omega_{cv}$ crosses the imaginary axis always from left-half plane (LHP) to right-half plane (RHP), i.e., from stable to unstable, for all $\tau_{k,v}$; else if $RT_v = -1$, then the root always crosses from RHP to LHP (from unstable to stable) for all the corresponding delays $\tau_{k,v}$ ($k = 0, 1, 2, \dots$).

Remark 3 [29] If a pair of characteristic roots, $s = \mp j\omega$, only touches the imaginary axis then so $W(\omega^2)$ touches ω^2 -axis. If the characteristic roots cross from LHP(RHP) to RHP(LHP), then $W(\omega^2)$ crosses from below(above) to above(below) the positive ω^2 -axis.

Based on Remark 3, the following relationships hold:

$$RT_v = \text{sgn} \left[\frac{dW(\sigma)}{d\sigma} \right] \Big|_{\sigma=\omega_{cv}^2} = \text{sgn}[W'(\omega_{cv}^2)] \quad (11)$$

Remark 4 In general, for SISO systems, when crossing values ω_{cv} are unique, and sorted in order $\omega_{c1} > \dots > \omega_{cv} > \dots > \omega_{cM}$, then RT_v corresponding to each one of these

crossings have alternating signs. Moreover since (1) represents a retarded class system, $W(\omega^2)$ goes to $+\infty$ with ω increasing. Owing to this, $RT_1 = +1$ at the largest crossing ω_{c1} , $RT_2 = -1$ at ω_{c2} and so on [30]. Special cases must be carefully considered, for example, when ω_{cv} is a tangent point of curve $W(\omega^2)$ [29] or when (1) has multiple roots [33,34]. For the sake of considering the general cases, here we shall assume that crossings $\omega_{cv} > 0$ are distinct.

4 Main results

In this article, the main interest is to design a controller $C(s)$ for an open-loop plant with transfer function $G(s)$ subject to delay, achieving closed-loop stability in two pre-determined delay intervals (Fig. 1). To keep the presentation compact, we first take $G(s)$ as a second-order plant,¹

$$G(s) = \frac{1}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (12)$$

Following the standard definitions, here ζ and ω_n are defined respectively as damping ratio and natural frequency of the open-loop plant.

For the design of $C(s)$, the following conditions must be met:

Condition A. The SISO LTI closed-loop system is stable for the delay-free case ($\tau = 0$);

Condition B. Given h_1, h_2 and h_3 with $0 < h_1 < h_2 < h_3$, the SISO LTI closed-loop system is stable for delays satisfying $\tau \in [0, h_1) \cup (h_2, h_3)$, and marginally stable for h_1, h_2 , and h_3 ;

Condition C. The controller $C(s)$ itself is stable.

4.1 Sufficient conditions

We first establish the sufficient conditions to be satisfied within the design process of $C(s)$. The following steps are considered for this purpose:

Step 1: Assignment of $\tau = h_1, h_2, h_3$ to crossings ω_{cv} .

The system has two separate stable intervals in τ parameter space. Specifically, the first three critical delay values, which cause stability–instability switches, are denoted by h_1, h_2 and h_3 .

A strategic decision here is that the number of crossings M will be set to 2 (see also Step 3), where we have two crossing values ω_{c1} and ω_{c2} . Due to Remark 4, note that we have $\omega_{c1} > \omega_{c2}$, with $s = j\omega_{c1}$ causing destabilizing crossing and $s = j\omega_{c2}$ creating a stabilizing crossing. Accordingly

¹ See discussion at the end of the section on how to extend the approach to higher-order systems.

h_1 and h_3 are assigned to the same crossing ω_{c1} , while being the successive delays values causing this crossing. That is, $\tau_{0,1} = \tilde{\tau}_1 = h_1$, and $\tau_{1,1} = h_3$ in (9), hence $h_3 = h_1 + \frac{2\pi}{\omega_{c1}}$. Next, ω_{c2} is assigned to h_2 , which corresponds to $\tau_{0,2} = \tilde{\tau}_2 = h_2$ in (9).

Moreover, since h_1 is the minimum positive $\tilde{\tau}_1$ value with respect to ω_{c1} and h_3 is the successive critical delay $\tau_{1,1}$, we have the condition $2h_1 < h_3$. In other words, although h_1, h_2, h_3 can be arbitrarily selected by the designer, $2h_1 < h_3$ must still be respected. In general, the initial selection of h_1 and h_3 follows $h_3 = h_3^*$ where

$$h_3^* = \begin{cases} h_3, & \text{if } 2h_1 < h_3; \\ 2h_1 + \lambda, & \text{if } 2h_1 \geq h_3, \end{cases} \tag{13}$$

and λ is a positive number.

Now that h_1 and h_3 are given, ω_{c1} can be calculated from (9) as

$$\omega_{c1} = \frac{2\pi}{h_3 - h_1}.$$

Step 2: Structure of $C(s)$.

Let $C(s)$ be in the form of $\frac{N(s)}{D(s)}$, where $\text{deg}[D(s)] \geq \text{deg}[N(s)]$. Since the design of stability intervals in Fig. 1 relies on the infinite dimensional system characteristic equation, one must make sure there are sufficiently many design parameters to achieve this goal. For example, the polynomials $D(s)$ and $N(s)$ can be expressed as follows:

$$D(s) = s^2 + \alpha_1 s + \alpha_0, \quad N(s) = \beta_2 s^2 + \beta_1 s + \beta_0 \tag{14}$$

where $\alpha_0, \alpha_1, \beta_0, \beta_1, \beta_2 \in \mathbb{R}$ are design parameters.

The characteristics equation in this case becomes

$$F(s, e^{-s\tau}, \alpha_k, \beta_k) = (s^2 + \alpha_1 s + \alpha_0) (s^2 + 2\zeta \omega_n s + \omega_n^2) + (\beta_2 s^2 + \beta_1 s + \beta_0) e^{-s\tau} \tag{15}$$

where the general form in (1) reads

$$A(s) = (s^2 + \alpha_1 s + \alpha_0) (s^2 + 2\zeta \omega_n s + \omega_n^2) \\ B(s) = \beta_2 s^2 + \beta_1 s + \beta_0$$

At this point, we have 6 unknowns $\omega_{c2}, \alpha_0, \alpha_1, \beta_0, \beta_1, \beta_2$ in total, denoted by the vector $\chi = [\omega_{c2}, \alpha_0, \alpha_1, \beta_0, \beta_1, \beta_2]$. Next we formulate the constraints that this vector should comply with.

Step 3: Constraints \mathcal{C} for all specifications.

From (15), Walton’s equation can be written as

$$W(\omega^2) = \omega^8 - (-4\omega_n^2 \zeta^2 - \alpha_1^2 + 2\omega_n^2 + 2\alpha_0) \omega^6 \\ + (4\alpha_1^2 \omega_n^2 \zeta^2 - 8\alpha_0 \omega_n^2 \zeta^2 - 2\alpha_1^2 \omega_n^2 + \omega_n^4$$

$$+ 4\alpha_0 \omega_n^2 + \alpha_0^2 - \beta_2^2) \omega^4 - (-4\alpha_0^2 \omega_n^2 \zeta^2 \\ - \alpha_1^2 \omega_n^4 + 2\alpha_0 \omega_n^4 + 2\alpha_0^2 \omega_n^2 \\ - 2\beta_0 \beta_2 + \beta_1^2) \omega^2 + (\alpha_0^2 \omega_n^4 - \beta_0^2) \tag{16}$$

Through (16), we have 2 constraints related to the imaginary axis crossing values:

$$\mathcal{C}_1 : W(\omega_{c1}^2) = 0, \\ \mathcal{C}_2 : W(\omega_{c2}^2) = 0.$$

As for the minimum positive critical delay value, referring to (8), we revise x and y variables as

$$x_k = \Re \left(-\frac{\beta_2 s^2 + \beta_1 s + \beta_0}{(s^2 + \alpha_1 s + \alpha_0)(s^2 + 2\zeta \omega_n s + \omega_n^2)} \right) \Big|_{s=j\omega_{ck}}, \\ y_k = \Im \left(-\frac{\beta_2 s^2 + \beta_1 s + \beta_0}{(s^2 + \alpha_1 s + \alpha_0)(s^2 + 2\zeta \omega_n s + \omega_n^2)} \right) \Big|_{s=j\omega_{ck}}.$$

where $k = 1, 2$. Since the minimum positive critical delay value must be equal to the given delays h_1 and h_2 respectively for ω_{c1} and ω_{c2} , we have two more constraints:

$$\mathcal{C}_3 : h_1 = \begin{cases} \frac{\arctan 2(y_1, x_1)}{\omega_{c1}}, & \text{if } \arctan 2(y_1, x_1) \geq 0; \\ \frac{\arctan 2(y_1, x_1) + 2\pi}{\omega_{c1}}, & \text{if } \arctan 2(y_1, x_1) < 0; \end{cases} \\ \mathcal{C}_4 : h_2 = \begin{cases} \frac{\arctan 2(y_2, x_2)}{\omega_{c2}}, & \text{if } \arctan 2(y_2, x_2) \geq 0; \\ \frac{\arctan 2(y_2, x_2) + 2\pi}{\omega_{c2}}, & \text{if } \arctan 2(y_2, x_2) < 0. \end{cases}$$

In order to guarantee the number of crossings M to be 2, we need to make sure (16) has only 2 distinct positive real roots $\sigma = \omega^2$ as per Remark 4. Firstly, (16) with $\sigma = \omega^2$ can be rewritten as:

$$\sigma^4 + r_3 \sigma^3 + r_2 \sigma^2 + r_1 \sigma + r_0 = 0 \tag{17}$$

where

$$r_3 = -(-4\omega_n^2 \zeta^2 - \alpha_1^2 + 2\omega_n^2 + 2\alpha_0) \\ r_2 = 4\alpha_1^2 \omega_n^2 \zeta^2 - 8\alpha_0 \omega_n^2 \zeta^2 - 2\alpha_1^2 \omega_n^2 + \omega_n^4 + 4\alpha_0 \omega_n^2 + \alpha_0^2 - \beta_2^2 \\ r_1 = -(-4\alpha_0^2 \omega_n^2 \zeta^2 - \alpha_1^2 \omega_n^4 + 2\alpha_0 \omega_n^4 + 2\alpha_0^2 \omega_n^2 - 2\beta_0 \beta_2 + \beta_1^2) \\ r_0 = \alpha_0^2 \omega_n^4 - \beta_0^2$$

Since ω_{c1}^2 and ω_{c2}^2 must be the two roots of (17), one factor of (17) is $(\sigma^2 - (\omega_{c1}^2 + \omega_{c2}^2)\sigma + \omega_{c1}^2 \omega_{c2}^2)$. Then, using factorization in (17), the other factor must be

$$\sigma^2 + (r_3 + \omega_{c1}^2 + \omega_{c2}^2) \sigma + \frac{r_0}{\omega_{c1}^2 \omega_{c2}^2} \tag{18}$$

To make sure $M = 2$, the zeros of (18) in σ should not be positive. Owing to this, we have to check the discriminant of (18):

$$\delta = \left(r_3 + \omega_{c1}^2 + \omega_{c2}^2\right)^2 - \frac{4r_0}{\omega_{c1}^2 \omega_{c2}^2} \tag{19}$$

The following two scenarios are to be considered:

1. Equation (18) has two complex zeros, that is $\delta < 0$. Denote by I_1 the set of all solutions χ satisfying $\delta < 0$.
2. Equation (18) has two negative real zeros, that is, $\delta > 0$ & $r_3 + \omega_{c1}^2 + \omega_{c2}^2 > 0$ & $\frac{r_0}{\omega_{c1}^2 \omega_{c2}^2} > 0$. In this case, the set of solutions satisfying the three inequalities is denoted by I_2 .

Consequently, the following constraint guarantees that $M = 2$ holds,

$$\mathcal{C}_5 : \chi \in I_1 \cup I_2$$

Remark 5 As per Remark 4, and $M = 2$ based on \mathcal{C}_5 , and $\omega_{c1} > \omega_{c2}$ holds as per Step 1, the crossing directions of $j\omega_{c1}$ and $j\omega_{c2}$ are automatically guaranteed and therefore do not need to be explicitly considered. In the case of MIMO problems with commensurate delays, these directions are not necessarily implied and hence must be taken into account explicitly.

Next, from *Condition A* that the delay-free system is stable, Routh’s array on $F(s, 1)$ in (15) can be implemented, where the Number of Sign (NS) changes in the first column of the array elements determines the number of unstable roots of the system [35]. Thus, for delay-free ($\tau = 0$) closed-loop system stability this constraint becomes:

$$\mathcal{C}_6 : NS = 0$$

which is formulated specifically for the problem at hand, as:

$$\begin{aligned} & 2\alpha_0\omega_n\zeta + \alpha_1\omega_n^2 + \beta_1 \\ & - \frac{(2\omega_n\zeta + \alpha_1)(\alpha_0\omega_n^2 + \beta_0)}{2\alpha_1\omega_n\zeta + \omega_n^2 + \alpha_0 + \beta_2 - \frac{2\alpha_0\omega_n\zeta + \alpha_1\omega_n^2 + \beta_1}{2\omega_n\zeta + \alpha_1}} > 0 \\ \& \quad 2\alpha_1\omega_n\zeta + \omega_n^2 + \alpha_0 + \beta_2 - \frac{2\alpha_0\omega_n\zeta + \alpha_1\omega_n^2 + \beta_1}{2\omega_n\zeta + \alpha_1} > 0 \\ \& \quad \alpha_0\omega_n^2 + \beta_0 > 0 \\ \& \quad 2\omega_n\zeta + \alpha_1 > 0 \end{aligned}$$

For *Condition C*, where the controller $C(s)$ must be a stable one, the constraint that the roots of the denominator $s^2 + \alpha_1s + \alpha_0$ of $C(s)$ to be in LHP is given by:

$$\mathcal{C}_7 : \alpha_0 > 0 \quad \& \quad \alpha_1 > 0$$

Finally, from Step 1, we can write down another constraint:

$$\mathcal{C}_8 : \omega_{c2} < \omega_{c1}$$

Now that the constraints \mathcal{C}_{1-8} sufficient to satisfy *Conditions A–C* are established, we next discuss how to compute the controller under these constraints.

4.2 Optimization setup

We now propose to utilize an optimization scheme to calculate $C(s)$ in light of the constraints \mathcal{C}_{1-8} .

Step 4: *fmincon* optimization.

To solve for the vector $\chi = [\omega_{c2}, \alpha_0, \alpha_1, \beta_0, \beta_1, \beta_2]$ satisfying the above listed constraints, we can consider this design problem as an optimization problem. A solution to χ may not exist, and if it does, it may not be unique. Since the aim is to find a solution, multiple solutions are however of no concern.

It can be observed that the type of constraints \mathcal{C}_{1-8} we have matches that of the nonlinear optimization toolboxes, namely *fmincon* tool of Matlab [36]. On the other hand, we so far did not formulate an optimization problem. For this, an objective function f is needed. Among many different ways one can formulate this function, here we propose to minimize the sum of the magnitudes of the coefficients in the controller, that is,

$$f = \alpha_0^2 + \alpha_1^2 + \beta_0^2 + \beta_1^2 + \beta_2^2 \tag{20}$$

Notice that with the above objective function, it now becomes possible to utilize *fmincon* tool in Matlab, as explained next.

Since we have a small-medium scale optimization problem with the number of variables < 100 , Sequential Quadratic Programming method can be used. In this method, *fmincon* solves a Quadratic Programming subproblem at each iteration. An estimate of the Hessian of the Lagrangian is updated at each iteration and a line search is performed using a merit function. An active set strategy is used to solve the programming subproblem.

For the implementation of \mathcal{C}_{1-8} in the standard form for *fmincon* active-set algorithm, tolerances for inequality and upper/lower bound constraints must be set. With Δ_k being small positive values (or vectors with all the entries being small positive values), constraints \mathcal{C}_{1-8} are adapted to the optimizer as follows

$$\begin{aligned} \mathcal{C}_1 & \rightarrow ceq_1(\alpha, \beta) = 0 \\ \mathcal{C}_2 & \rightarrow ceq_2(\chi) = 0 \\ \mathcal{C}_3 & \rightarrow ceq_3(\alpha, \beta) = 0 \\ \mathcal{C}_4 & \rightarrow ceq_4(\chi) = 0 \\ \mathcal{C}_5 & \rightarrow c_1(\chi) \leq -\Delta_1 \end{aligned}$$

$$\begin{aligned} \mathcal{C}_6 &\rightarrow c_2(\alpha, \beta) \leq -\Delta_2 \quad \& \quad A \cdot \chi \leq b - \Delta_3 \\ \mathcal{C}_{7-8} &\rightarrow lb + \Delta_4 \leq \chi \leq ub - \Delta_5 \end{aligned}$$

Remark 6 *fmincon* numerical tool is implemented under several considerations: (i) firstly in the domain of optimization, constraints are assumed to be continuous functions in terms of their arguments; (ii) we have that the obtained result, if any, will be very likely a local minimum; (iii) in general there are no set rules as to how one selects a feasible initial point.

Consideration (i) is neglected here partly because all the constraints, possibly except at some special singularity points, will remain as continuous functions of their arguments. For *consideration (ii)*, since we only care about whether the constraints are satisfied in this problem, the local minimum point χ is equally admissible. Finally, we have to pay attention to *consideration (iii)*, in which initial condition could affect whether this optimization method leads to an admissible solution. A rule for selecting an appropriate initial condition is well stated in the Appendix.

Based on the Appendix, a rigorous way of formulating an initial condition follows

$$\begin{cases} \omega_{c2} \rightarrow \text{slightly less than } \omega_{c1}; & \omega_{c2} = \omega_{c1} - \epsilon_3 \\ \alpha_0 \rightarrow \text{slightly less than } \omega_{c1}^2; & \alpha_0 = \omega_{c1}^2 - \epsilon_0 \\ \alpha_1 \rightarrow \epsilon_1 \\ \beta_0, \beta_1, \beta_2 \rightarrow \epsilon_2 \end{cases}$$

where ϵ_l are small positive values. In other words, with the above proposed initial condition, the characteristic equation in (15) becomes

$$F(s, e^{-s\tau}) \cong (s^2 + \epsilon_1 s + (\omega_{c1} - \epsilon_0)^2) (s^2 + 2\zeta\omega_n s + \omega_n^2) + \epsilon_2(s^2 + s + 1)e^{-s\tau} \tag{21}$$

The small gains of $\beta_k = \epsilon_2$ will not let the time delayed term $e^{-\tau s}$ render a crossing, and the system is almost marginally stable for a large interval of τ due to the factor $s^2 + \epsilon_1 s + (\omega_{c1} - \epsilon_0)^2$.

As we tested, this rule for the initial condition works well for a large interval of ω_n , and for ζ from 0 (undamped) to some value above 1 (over-damped); see details in the next section. For some cases, this rule may not work well for undamped systems as this creates potentially an additional crossing at ω_n , violating our design constraints at the very first step of our problem setup. In such cases we may need to propose higher order controllers for $C(s)$, which can be done following the above steps.

Remark 7 In the case when $m = n$ in (2), we have a neutral class system. For such systems, stability may not be preserved as $\tau : 0 \rightarrow 0^+$ and could be lost for all $\tau > 0$, even if the

delay-free ($\tau = 0$) system is stable. To prevent this on the LTI SISO system at hand, an additional condition is needed, which is $|b_n| < 1$ in (2) [27,28]. Remainder of the analysis is the same, following Steps 1–4.

4.3 Extensions to higher-order problems

Designing a controller for higher-order open-loop systems that leads to a closed-loop system with two separate stability intervals is even more challenging. This is mainly due to the fact that arising equations are of higher order making it more difficult to keep system crossings under ‘control’. Indeed, inspecting the above design procedure (Steps 1–4), the only step where the assumption of a second-order plant $G(s)$ becomes a key enabler is where factorization and discriminant are used to prevent any additional unwanted crossings in the closed-loop system (see Examples 1, 2 in Sect. 5). By doing so, any unwanted delays are avoided hence it becomes more feasible to attain the intended stable delay intervals.

In the sequel, we discuss two alternatives to the above manipulation where factorization to smaller factors may not be easily established due to the size of the problem. Notice that the design process in Steps 1–4 is primarily concerned with “preventing” additional unwanted crossings. This in some sense is the safest choice, and should be followed when possible (e.g., in the case of low-order problems). On the other hand, the designer has also the option of “allowing” additional unwanted crossings as long as the delay values corresponding to these crossings do not ruin the intended stability intervals along the delay axis. Allowing such crossings simplifies the design as factorization is no longer necessary. Since “prevent” option is already covered in sufficient detail in Steps 1–4, we further discuss the “allow” option next.

4.3.1 Allow additional unwanted crossings

Since two crossings must exist for the intended design, let $\omega_{c3}, \dots, \omega_{cv}, \dots, \omega_{cM}$ be the list of unwanted crossings, where $\sigma_v = \omega_{cv}^2 > 0$. Then we have three sets of additional constraints for the optimizer:

- (a) Allow unwanted crossings but make sure the minimum positive delay value $\tilde{\tau}_v$ corresponding to each crossing is sufficiently large. That is, unwanted $\tilde{\tau}_v$ should occur outside the range of the intended stability intervals. For this, $\tilde{\tau}_v > h_3$ must hold.
- (b) Let $\tilde{\tau}_v = \epsilon_v + h_3$ with $\epsilon_v \in \mathbb{R}_+$. Since the delay value smaller than the minimum positive delay $\tilde{\tau}_v$ should be negative, $\tilde{\tau}_v - 2\pi/\omega_{cv} < 0$, we can write $\epsilon_v + h_3 - 2\pi/\omega_{cv} < 0$ which implies that the unwanted crossings satisfy $\omega_{cv} < 2\pi/(\epsilon_v + h_3)$. Furthermore, since crossing directions are implied, ω_{c2} must be the second largest crossing, hence a stricter condition $\omega_{cv} < \omega_{c2}$ must hold.

The remaining σ_v roots, $\sigma_{M+1}, \dots, \sigma_m$, are infeasible for a crossing. For example they can be set as negative quantities, $\sigma_v < 0$. Then, one can design a *candidate* Walton’s equation given by

$$\bar{W}(\sigma) := \prod_{v=1}^m (\sigma - \sigma_v) = \sigma^m + \sum_{k=0}^{m-1} \bar{\ell}_k \sigma^k,$$

where $\sigma_1 = \omega_{c1}^2$ is given, and the remaining σ_v are optimization variables. Next, compare this equation with that of Walton’s, $W(\sigma) = \sigma^m + \sum_{k=0}^{m-1} \ell_k \sigma^k$, which leads to another set of constraints as described next.

- (c) To attain the desired and unwanted crossings, as well as the infeasible solutions σ_v , sign of these solutions must be imposed in the optimization, and the coefficients of Walton’s equation must match with those of the candidate equation, $|\bar{\ell}_k - \ell_k| = 0, k = 0, 1, \dots, m - 1$. This imposes a constraint on the nature of the roots of the equation since $\sigma_v, v \geq 2$, are all optimization variables.

Notice that the above three sets of constraints are not different in structure than those already available in the optimization routine as established in Steps 1–4. The set described in (a) is related to minimum delay formulations already available for h_1 and h_2 (see constraints \mathcal{C}_{3-4}). The set described in (b) is similar to the inequality constraint \mathcal{C}_8 considered, and the set in (c) is incorporated as an equality constraint in the solver. In some sense conditions in sets (a), (b) guarantee sparsely repeated delays corresponding to the unwanted crossings, thereby reducing the possibility of these delays interfering with the desired stability intervals, while set (c) gives control over the number of feasible crossings the system could have.

The above approach of safely creating additional unwanted crossings can be over-constraining in some cases since these crossings and their delay values must respect additional conditions. For this reason, optimizing the controller whereby the closed-loop system has many crossings can be more difficult than optimizing for fewer crossings. Hence, the logic line here must be to achieve only the two crossings ω_{c1} and ω_{c2} as the design requires. In the case the formulations in Steps 1–4 indicate that this cannot be feasible, one would then relax the problem with as fewer number of additional unwanted crossings as possible. An example of this scenario is demonstrated in Example 3 in Section 5 with and without using factorization.

4.3.2 A bottom-up approach with stable pole/zero cancellations

Another option is to take a bottom-up approach where one starts with a “building brick” (e.g., a 2nd order plant $G(s)$

with a PID controller $C_1(s)$), and grows the loop transfer function by successively inserting additional controllers $C_k(s), k = 1, \dots, \bar{k}$, into the forward loop. In this approach, the optimization needs to be set up at each step k , and the design process through Steps 1–4 must be re-run while keeping fixed the stability intervals $\tau \in [0, h_1) \cup (h_2, h_3)$. The main idea here is to first generate a higher-order loop transfer function $L(s) = G(s) \prod_{k=1}^{\bar{k}} C_k(s)$ achieving stability for $\tau \in [0, h_1) \cup (h_2, h_3)$ in the closed loop (see Example 3 in Section 5).

Next, consider that the aim is to design a controller $\bar{C}(s)$ for a high-order plant transfer function $\bar{G}(s)$ achieving stability in the closed-loop for $\tau \in [0, h_1) \cup (h_2, h_3)$. Then, this controller is given by $\bar{C}(s) = \frac{L(s)}{\bar{G}(s)}$, where care must be taken with the orders of $L(s)$ such that the controller $\bar{C}(s)$ is proper hence realizable. Moreover, this design approach will work owing to pole/zero cancellations. Hence, one must make sure those cancellations arise only amongst stable poles and zeros, which brings additional constraints both on the plant $\bar{G}(s)$ and loop transfer function $L(s)$. Needless to say, effects of such cancellations under noise and uncertainties must be thoroughly studied before certifying the controller $\bar{C}(s)$ for implementation.

5 Example case studies

5.1 Example 1: a computational study

Take $\zeta = 0.7$ and $\omega_n = 10$ rad/s in $G(s)$ and pick $h_1 = 0.1, h_2 = 0.5, h_3 = 0.7$ in *Condition C* (see also Figs. 1 and 2). Following Steps 1–4, constraints \mathcal{C}_{1-8} can be incorporated into the *fmincon* solver in Matlab.

Specifically, in \mathcal{C}_5 , we reconstruct the feasible interval of $I_1 \cup I_2$ to simplify the way of expressing the constraints. For the two roots of (18) to become complex conjugate, the following constraint denoted by set I_1 must hold,

$$I_1: \left(4\omega_n^2 \zeta^2 + \alpha_1^2 - 2\omega_n^2 - 2\alpha_0 + \omega_{c1}^2 + \omega_{c2}^2 \right)^2 - \frac{4 \left(\alpha_0^2 \omega_n^4 - \beta_0^2 \right)}{\omega_{c1}^2 \omega_{c2}^2} < 0$$

Then for the two roots of (18) to lie in LHP, two constraints denoted by set I_3 must hold as per Routh’s Array,

$$I_3: 4\omega_n^2 \zeta^2 + \alpha_1^2 - 2\omega_n^2 - 2\alpha_0 + \omega_{c1}^2 + \omega_{c2}^2 > 0$$

$$\& \frac{\alpha_0^2 \omega_n^4 - \beta_0^2}{\omega_{c1}^2 \omega_{c2}^2} > 0$$

Obviously, $I_1 \cup I_3 \equiv I_1 \cup I_2$, and the above two constraints are inserted into the *fmincon* solver.

Next, we test whether initial condition works well. Given h_1 and h_3 , we have $\omega_{c1} = \frac{2\pi}{h_3-h_1} = 10.4720$, then the initial condition becomes

$$\begin{cases} \omega_{c2} \rightarrow \text{slightly less than } 10.4720 \\ \alpha_0 \rightarrow \text{slightly less than } 109.6628 \\ \alpha_1 \rightarrow 0 \\ \beta_0, \beta_1, \beta_2 \rightarrow 0.01 \end{cases}$$

The initial points for ω_{c2} and α_0 are then put on a grid ($p \times q$), and slightly varied,

$$\chi_0 = [10.4720 - 0.05 \times p, 109.6628 - 0.1 \times q, 0, 0.01, 0.01, 0.01]$$

The optimization is re-run for $p \times q$ times, where $p = 1, 2, \dots, 20$, $q = 1, 2, \dots, 20$. For these 400 initial points, we find 98.5% of all the solutions of χ converge to the same point $[\omega_{c2}, \alpha_0, \alpha_1, \beta_0, \beta_1, \beta_2] = [8.7378, 91.7096, 0.0010, -1.7621, 111.457, 21.5547]$, while 1.5% of them converge to different solutions. Moreover, there are no issues with convergence. That is, all 400 runs lead to a feasible solution for χ .

The solution χ that converge at the highest percentage of 98.5% is next taken for further study. In this solution, we have $\omega_{c2} = 8.7378$ rad/s, which is smaller than ω_{c1} as expected. Furthermore, these two crossing frequencies signify the oscillation frequency of the closed-loop system output at the critical delays; ω_{c1} for h_1 and h_3 , and ω_{c2} for h_2 . We next use the respective controller $C(s)$ in Simulink and run unit step response simulation of the closed-loop system with delay values, $\tau = 0, 0.095, 0.1, 0.105, 0.495, 0.5, 0.505, 0.695, 0.7, 0.705$ (s). As simulations indicate in Fig. 3, the stability of the delay-free system and the stability boundary of critical delay values match perfectly with the results obtained from the controller design method. Furthermore, in the time window 15–20s, at the critical values $\tau = h_1 = 0.1$ and $\tau = h_3 = 0.7$, we observe about 8.25 cycles of oscillations, corresponding to approximately about 10.36 rad/s, which matches with $\omega_{c1} = 10.4720$, and similarly at $\tau = h_2 = 0.5$, we observe about 6.75 cycles ≈ 8.48 rad/s, which is very close to the exact solution $\omega_{c2} = 8.7378$.

In order to verify that stability intervals of the closed-loop system match the pre-determined stability “certificates”, rightmost root of the closed-loop characteristic equation is computed using the QPmR toolbox [37] in the interval $\tau \in (0, 1)$. For clarity and relevance, only the real part of the rightmost root $\Re(s_{RM})$ is depicted, see Fig. 4, which reveals that the stability intervals match the intended design $\tau \in [0, 0.1) \cup (0.5, 0.7)$.

Remark 8 Notice that the controller designed above has stable poles but close to the imaginary axis due to the small

α_1 value. This controller works well in the ideal closed-loop model and simulations. However, in the presence of sensor noise that may appear in an experimental system and with possible uncertainties in α_1 , e.g., due to digital implementation, this weakly-damped controller may cause internal stability problems [38]. To prevent this, a positive lower bound on α_1 can be set as another constraint (see Example 2).

5.2 Example 2: an experimental study

We next implement the controller design approach on a Rotary Servo Base Unit speed-control experiment comprising of a DC-motor with gearbox and a shaft mounted encoder, as shown in Fig. 5. In our experimental setup, the Rotary Servo Base Unit is linked to a computer through a National Instruments data acquisition board supported by QUARC embedded system software using Matlab/Simulink toolbox. A fixed sample period of 0.001 sec and relative tolerance of 10^{-8} are utilized with ODE45 time integration scheme in all the experiments.

For speed control, the model of the DC motor is captured by a first order transfer function (TF) from voltage to motor shaft angular velocity, as

$$G_{motor}(s) = \frac{8.4792}{0.028s + 1} \quad (22)$$

A first-order low-pass filter with cut-off frequency of 50 rad/s is designed to block out the high frequency noise, which is inserted into Simulink at the encoder reading channel,

$$G_{filter}(s) = \frac{1}{0.02s + 1} \quad (23)$$

The open loop TF then becomes a second-order system given by that of the DC motor in series with the filter

$$G(s) = G_{motor}(s)G_{filter}(s) = \frac{15141.375}{s^2 + 85.7143s + 1785.715} \quad (24)$$

where $\zeta = 1.01419$ and $\omega_n = 42.25772$. Consistent with Fig. 2, we define the speed as the output of the filter, and a unity negative feedback with an artificial transport time delay is introduced in the Simulink interface. Furthermore, to be consistent with $G(s)$ in (12) which has a unity gain, we insert a gain of $1/15, 141.375$ in front of the controller $C(s)$.

Here, we pick $h_1 = 0.3$, $h_2 = 1.0$, $h_3 = 1.8$ in Condition B, and follow Steps 1–4 as was done in Example 1 to bring the constraints into the *fmincon* solver. Given h_1 and h_3 , we have $\omega_{c1} = \frac{2\pi}{h_3-h_1} = 4.1888$, hence the initial condition becomes

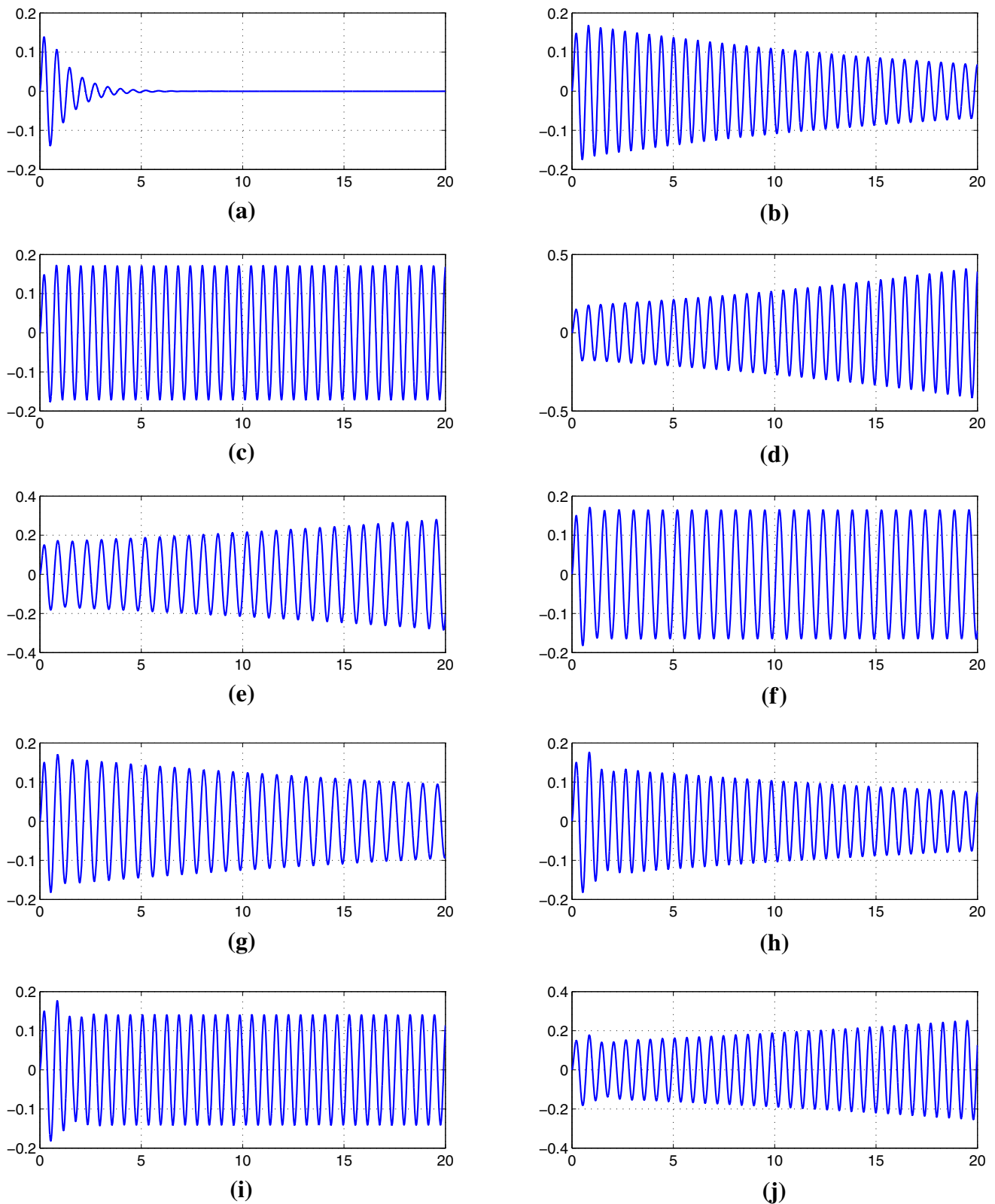


Fig. 3 Example 1: unit step response of the closed loop system simulations at **a** $\tau = 0$, **b** $\tau = 0.095$, **c** $\tau = 0.1$, **d** $\tau = 0.105$, **e** $\tau = 0.495$, **f** $\tau = 0.5$, **g** $\tau = 0.505$, **h** $\tau = 0.695$, **i** $\tau = 0.7$, **j** $\tau = 0.705$, with the con-

troller $C(s)$ designed using the optimization method described in Sect. 4 and the pre-determined stability decomposition $\tau \in [0, 0.1) \cup (0.5, 0.7)$

Fig. 4 Real part of the rightmost root of the closed-loop system in Example 1 as τ varies from 0 to 1 (computed with QPmR toolbox [37])

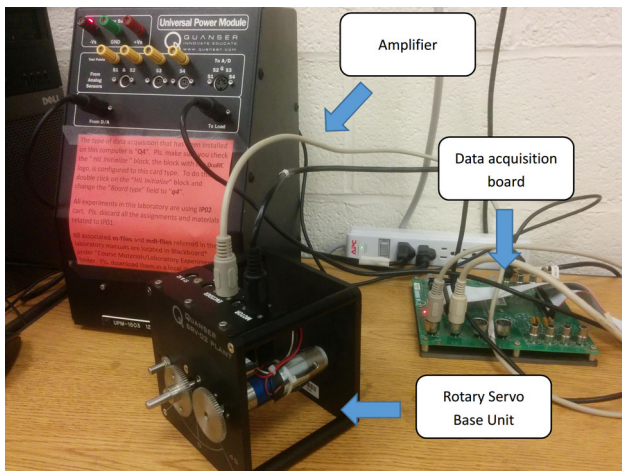
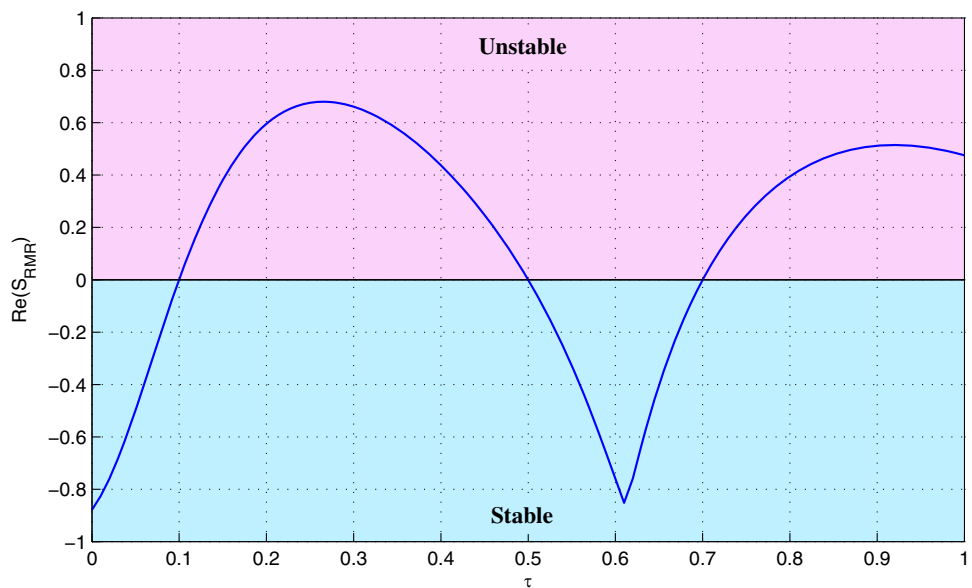


Fig. 5 Experimental setup of Rotary Servo Base Unit with amplifier and data acquisition board

$$\begin{cases} \omega_{c2} \rightarrow \text{slightly less than } 4.1888 \\ \alpha_0 \rightarrow \text{slightly less than } 17.5460 \\ \alpha_1 \rightarrow 0 \\ \beta_0, \beta_1, \beta_2 \rightarrow 0.001 \end{cases}$$

Solution set using this initial condition is computed as $[\omega_{c2}, \alpha_0, \alpha_1, \beta_0, \beta_1, \beta_2] = [3.4992, 15.5508, 0.0010, 16, 057.60, 853.6112, 891.4406]$. This corresponds to the following numerical expression of the controller

$$C_a(s) = \frac{891.4406s^2 + 853.6112s + 16,057.60}{15,141.375(s^2 + 0.001s + 15.5508)} \quad (25)$$

Notice that this controller has very low damping $\alpha_1 \approx 0$, and therefore may not be appropriate from internal stability point of view. Therefore, we change the constraint

on α_1 , now setting $\alpha_1 \geq 0.2$ in our optimization. This yields $[\omega_{c2}, \alpha_0, \alpha_1, \beta_0, \beta_1, \beta_2] = [3.2884, 15.0340, 0.2000, 14, 697.23, 1033.999, 722.7142]$, revising the numerical expression of the controller to

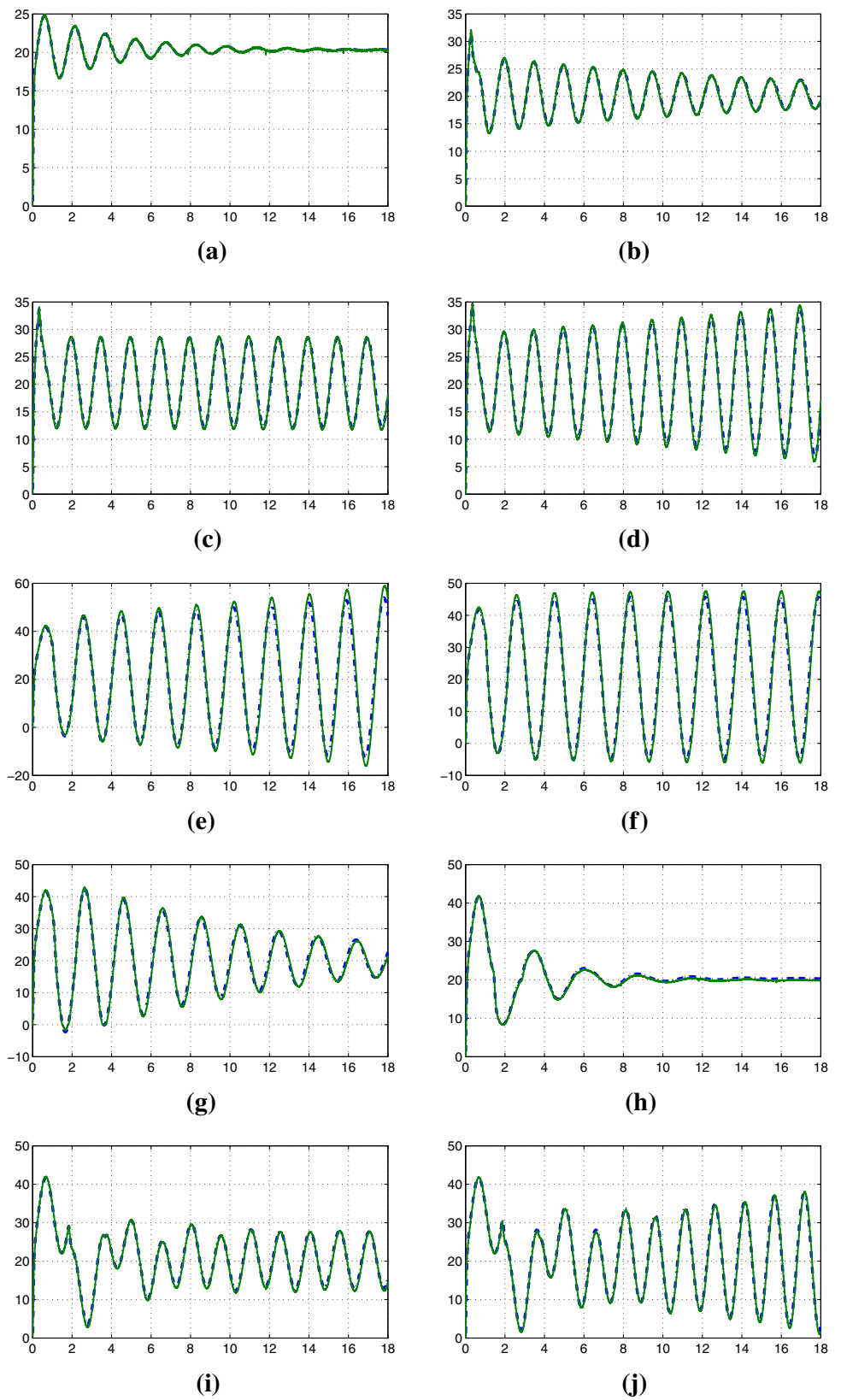
$$C_b(s) = \frac{722.7142s^2 + 1033.999s + 14,697.23}{15,141.375(s^2 + 0.2s + 15.0340)} \quad (26)$$

We next test the validity of the controller using a step input with a gain of 57.52, at $\tau = 0, 0.25, 0.3, 0.32, 0.99, 1.0, 1.05, 1.4, 1.8, 1.85$ in both simulations and experiments, as shown in Fig. 6. In the figure, we see that except some minimal discrepancies, which may be due to slight variations in DC motor dynamics and nonlinearities, the tendency of curves and stability properties can be clearly observed. Interestingly, at $\tau = 1.4$, when the operating point is inside the second stable interval, the output of the closed-loop system has fairly good settling time and transient characteristics, which is worthy of future study.

5.3 Example 3: higher-order loop transfer function

The developed controller design approach can also be repeatedly implemented to design cascaded controllers, see also Sect. 4.3.2. Consider the DC motor system in Example 2, where we designed $C(s) = C_a(s)$ in (25). While this controller is sufficient to achieve the design requirements, we now propose to insert a PID controller $C_{pid}(s) = \frac{k_d s^2 + k_p s + k_i}{s}$ into the closed-loop, in cascade to $C_a(s)$, as depicted in Fig. 7. In this setting, we aim to tune the PID gains k_p, k_i and k_d , but we still wish to maintain the same stability intervals as in Example 2. We find out however that h_1, h_2, h_3 values in Example 2 cannot be attained with the addition of the PID

Fig. 6 Example 2: step response of the motor-shaft angular velocity with controller $C_b(s)$ in both experiments and simulations at delay values **a** $\tau = 0$, **b** $\tau = 0.25$, **c** $\tau = 0.3$, **d** $\tau = 0.32$, **e** $\tau = 0.99$, **f** $\tau = 1.0$, **g** $\tau = 1.05$, **h** $\tau = 1.4$, **i** $\tau = 1.8$, **j** $\tau = 1.85$ (Experimental data: *solid line*; Simulation data: *dashed line*). Experimental results and model predictions are superimposed. Control design procedure utilizes the pre-determined stability decomposition of $\tau \in [0, 0.3) \cup (1.0, 1.8)$



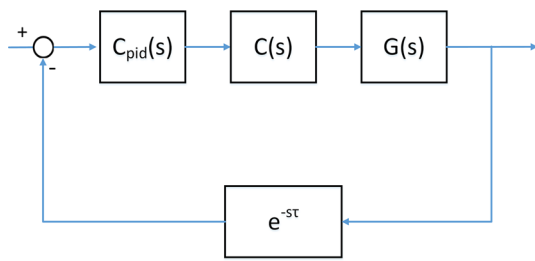


Fig. 7 Model of Example 3

controller, mainly because we have only three parameters to design, which are not sufficiently many for the over constrained optimization problem at hand. We therefore set free α_0 and α_1 in C_a , to relax the parameter space.

The system shown in Fig. 7 has the following characteristic equation:

$$\begin{aligned} & (s^2 + \alpha_1 s + \alpha_0) (s^2 + 85.7147s + 1785.7149) s \\ & + (891.4406s^2 + 853.6112s + 16,058) \\ & \times (k_d s^2 + k_p s + k_i) e^{-s\tau} = 0, \end{aligned} \quad (27)$$

which can be easily converted to Walton's equation from (16). Since the constant term in $W(\omega^2)$ function is negative, odd number of positive real roots of $\sigma = \omega^2$ exists as per *Descartes rule of signs* [39]. Since ω_{c1} and ω_{c2} are enforced as crossings, we have at least three crossings in this closed-loop system.

Moreover, we have h_1 and h_3 assigned to ω_{c1} , while h_2 is associated with ω_{c2} . Since the function $W(\omega^2)$ is of order-5 with respect to $\sigma = \omega^2$, in this case, using *factorization* in $W(\omega^2)$, we have one factor to be $\sigma^2 - (\omega_{c1}^2 + \omega_{c2}^2)\sigma + \omega_{c1}^2 \omega_{c2}^2$, and the other factor is of order-3. Since negative of the *discriminant* of a 3^{rd} order polynomial indicates one real pole and two complex poles [40], using this condition, we can make sure at most one positive real root exists in the remaining factor, rendering in the worst case one additional unwanted crossing ω_{c3} . Note that imposing σ to be complex is simple to implement with the help of discriminant. One can alternatively restrict σ from being positive, to rule out the presence of unwanted crossings.

Next, we constraint ω_{c3} to be as small as possible such that it does not cause frequent solutions of delay values in (9). For this, we have $\omega_{c3} < \omega_{c2}$ in the optimization environment. This is done as a necessary condition to prevent those delay values to undesirably interfere with the pre-selected delay intervals $[0, h_1) \cup (h_2, h_3)$, consistent with Sect. 4.3.1. Consequently, we have the order $\omega_{c1} > \omega_{c2} > \omega_{c3}$ with root crossing directions implied (see Remarks 4, 5) as destabilizing at ω_{c1} and ω_{c3} , and stabilizing at ω_{c2} .

For sufficiency such that one can guarantee that the two disjoint stability intervals $[0, h_1)$ and (h_2, h_3) are guaranteed to exist in τ domain, the minimum positive delay value $\tilde{\tau}_3$

corresponding to ω_{c3} must appear in τ domain but at a value greater than h_3 . Hence, a constraint is set for this as $\tilde{\tau}_3 > h_3$. The minimum positive delay values for ω_{c1} and ω_{c2} are carried over from Example 2. At last, NS is checked for delay-free closed-loop system stability.

Considering all the above constraints, another *fmincon* solver is set with a new unknown vector $\chi = [k_p, k_i, k_d, \alpha_0, \alpha_1, \omega_{c2}, \omega_{c3}]$. The initial condition is easily extracted from the feasible solution point in Example 2, which is $k_p \rightarrow 1$, $k_i \rightarrow 0$, $k_d \rightarrow 0$, $\alpha_0 \rightarrow 15.5508$, $\alpha_1 \rightarrow 0.0001$ and $\omega_{c3} \rightarrow 0$. With the objective function

$$f = k_p^2 + k_i^2 + k_d^2,$$

the optimization yields the following solution point

$$\chi = [1.2412, 1.3285, 0.0392, 15.0702, 0.0743, 3.0201, 1.1784]$$

rendering $\omega_{c3} = 1.1784$ relatively small as desired. Moreover, it returns $\tilde{\tau}_3 = 2.0620$, which is the smallest delay value for which the system characteristic roots will be on the imaginary axis at $s = \mp j\omega_{c3}$. This delay value, as designed, is greater than $h_3 = 1.8$ and hence does not interfere with the pre-determined stability intervals.

Next, with the PID gains found as $k_p = 1.2412$, $k_i = 1.3285$, and $k_d = 0.0392$, and the controller $C(s)$ as

$$C(s) = \frac{891.4406s^2 + 853.6112s + 16,057.60}{15,141.375(s^2 + 0.0743s + 15.0702)}$$

we perform simulations of the closed loop in Fig. 7 in Simulink, which present agreement in system stability with the pre-determined stable delay intervals $\tau \in [0, 0.3) \cup (1, 1.8)$ (simulations suppressed to conserve space).

Finally, we remove the factorization condition in the above setting, and instead revisit Sect. 4.3.1 to setup a new optimization problem where constraint sets (a), (b) are implemented and we allow only one additional unwanted crossing ω_{c3} hence $M = 3$ in set (c). In light of this setting, we have five coefficients to match since $m = 5$, and moreover the optimization variables are collected by the vector $\chi = [k_p, k_i, k_d, \alpha_0, \alpha_1, \omega_{c2}, \omega_{c3}, \sigma_4, \sigma_5]$. The end result using the same optimization function f as above leads to the following optimal solution

$$\chi = [1.0384, 0.2282, 0.0128, 15.4742, 0.0015, 3.4488, 0.1661, -828.3114, -2815.0664]$$

which indicates that the PID controller gains are given by $k_p = 1.0384$, $k_i = 0.2282$, $k_d = 0.0128$, and the new controller reads

$$C(s) = \frac{891.4406s^2 + 853.6112s + 16,057.60}{15,141.375(s^2 + 0.0015s + 15.4742)}$$

As designed, the additional crossing $\omega_{c3} = 0.1661$ is less than ω_{c2} and moreover its corresponding minimum positive delay value $\tilde{\tau}_3 = 13.3599$ (not shown in the above) is larger than h_3 . Further, the two solutions σ_4 and σ_5 generate infeasible crossings as they are both negative as imposed by the structure of candidate Walton’s equation \overline{W} . Implementation of these controllers in Simulink and QPmR validates that indeed the intended stability intervals are achieved.

6 Conclusion

An eigenvalue-based optimization scheme is developed in this article to design controllers for a class of LTI SISO systems subject to delay, with the aim to achieve stability of the closed loop in two pre-determined disjoint intervals along the delay axis. This challenging reverse problem is successfully solved over several examples and demonstrated on an experimental study, offering a strong foundation toward solving more complex problems involving multiple stability intervals. It is critical to note that, driven by applications, control problems are becoming increasingly more complex, involving the design of multiple parameters. While optimization tools to this end were in majority utilized to solve Linear Matrix Inequalities (LMIs) stemming from Lyapunov stability theory, formulation of the control problem in the frequency domain and connecting it with optimization tools offers new opportunities. This was indeed recognized by a recent study [21] where frequency domain conditions of DIS property were tested using LMIs. In future studies, we will focus on extending the proposed method to the case of MIMO systems considering additional constraints as highlighted in Remark 1. We will also systematically investigate the control of unstable open-loop plants following the footprints in [26], and design higher-order controllers for higher-order plants.

Appendix: selecting initial conditions for the optimization

To find out the best way to select the initial condition, we first reduce the number of constraints and attempt to search for an initial condition point P satisfying some special cases. These cases are stated as follows:

- I. At $\tau = 0$, the delay-free system $F(s, 1)$ is stable. In this case, the set of solutions to χ is denoted by Q_0 .
- II. Denoted by Q_1 is the set of possible solutions χ rendering a crossing ω_{c1} at a critical delay h_1 .

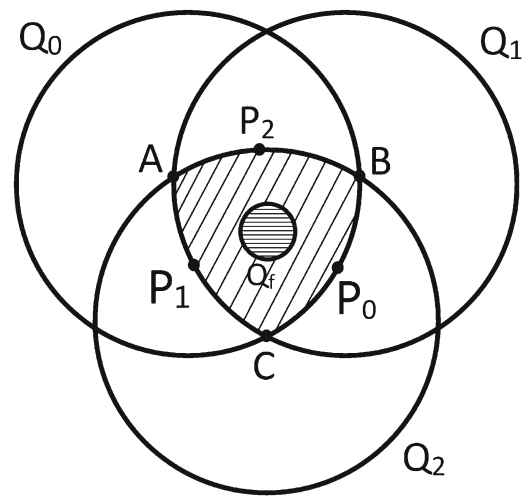


Fig. 8 Venn diagram of Q_0, Q_1 and Q_2 in the “Appendix”

- III. Denoted by Q_2 is the set of possible solutions χ rendering a crossing ω_{c2} at a critical delay h_2 .

Note that the three conditions I–III above do not guarantee the stability decomposition in Fig. 1, but only guarantee a crossing at h_1 and h_2 , and the stability of the delay-free system. In Fig. 8, the intersection area of sets Q_0, Q_1 and Q_2 is schematically pictured by \overline{ABC} . Since the constraints corresponding to Q_0, Q_1 and Q_2 are necessary conditions of \mathcal{C}_{1-8} , the feasible solution set satisfying \mathcal{C}_{1-8} , denoted by Q_f , must lie inside the region \overline{ABC} . Thus, a good initial condition would be to pick a point P inside of \overline{ABC} .

When there are multiple design parameters, such as the case here, it may not be easy to select a point inside the parametric region \overline{ABC} . Nevertheless, it is quite practical to select a point nearby the boundaries of \overline{ABC} . Fortunately, there exist several ways to calculate the parametric settings, e.g., the points at P_0, P_1, P_2 . The point P_0 satisfying Condition I is trivial to obtain from the stability analysis of the delay-free system. The points P_1 and P_2 follow the same type of mathematical calculations, and guarantee a single crossing either at h_1 and h_2 . Conditions on system parameters guaranteeing a crossing can easily be obtained using frequency sweeping methods, see, e.g., [18]. Simply, set $s = j\omega, \tau = h_1$ or $\tau = h_2$ in (15), and as ω increases from zero to an upper bound, one can solve the real and imaginary part of the system characteristic equation from which an implicit formula can be obtained on system parameters satisfying the crossing $s = j\omega$ at τ .

Application of the above general guidelines on the specific problem at hand leads to the following

$$F_{\Re}(\omega, \alpha_k, \beta_k, \tau) = \omega^4 + (-2\alpha_1\omega_n\zeta - \cos(\tau\omega)\beta_2 - \omega_n^2 - \alpha_0)\omega^2$$

$$+ \sin(\tau\omega)\beta_1\omega + \alpha_0\omega_n^2 + \cos(\tau\omega)\beta_0 \quad (28)$$

$$F_{\Im}(\omega, \alpha_k, \beta_k, \tau) = (-2\omega_n\zeta - \alpha_1)\omega^3 \\ + \sin(\tau\omega)\beta_2\omega^2 + (2\alpha_0\omega_n\zeta + \alpha_1\omega_n^2 \\ + \cos(\tau\omega)\beta_1)\omega - \sin(\tau\omega)\beta_0 \quad (29)$$

Since it is not easy to solve (28), (29) simultaneously, we propose to pick $\alpha_0 = \omega_{c1}^2$ and $\alpha_1 = 0$ to simplify (28), (29) as

$$F_{\Re}(\omega, \beta_0, \beta_1, \beta_2, \tau) = \omega^4 \\ + (-\cos(\tau\omega)\beta_2 - \omega_n^2 - \omega_{c1}^2)\omega^2 \\ + \sin(\tau\omega)\beta_1\omega + \omega_{c1}^2\omega_n^2 + \cos(\tau\omega)\beta_0 \quad (30)$$

$$F_{\Im}(\omega, \beta_0, \beta_1, \beta_2, \tau) = -2\omega_n\zeta\omega^3 + \sin(\tau\omega)\beta_2\omega^2 \\ + (2\omega_{c1}^2\omega_n\zeta + \cos(\tau\omega)\beta_1)\omega - \sin(\tau\omega)\beta_0 \quad (31)$$

For the set Q_1 and Q_2 , at $\tau = h_1$, $\omega = \omega_{c1}$ and at $\tau = h_2$, $\omega = \omega_{c2}$ respectively. Then, the following four equations obtained from (30) and (31) hold.

$$F_{\Re}(\omega_{c1}, \beta_0, \beta_1, \beta_2, h_1) = 0 \\ F_{\Im}(\omega_{c1}, \beta_0, \beta_1, \beta_2, h_1) = 0 \\ F_{\Re}(\omega_{c2}, \beta_0, \beta_1, \beta_2, h_2) = 0 \\ F_{\Im}(\omega_{c2}, \beta_0, \beta_1, \beta_2, h_2) = 0 \quad (32)$$

Since ω_{c1} , h_1 and h_2 are known in Step 1, a particular solution of ω_{c2} , β_0 , β_1 , β_2 can be extracted from the above four equations, that is, $\omega_{c2} \rightarrow \omega_{c1}$, $\beta_0 = \beta_2\omega_{c1}^2$, $\beta_1 \rightarrow 0$. In this case, we can pick $\beta_2 \rightarrow 0^+$, then $\beta_0 \rightarrow 0^+$. With the choice of small $\beta_k = \epsilon$, the coefficient of the exponential term $e^{-s\tau}$ will be vanishing, hence this point is at the boundary of the set Q_1 , e.g., point P_1 , and also at the boundary of the set Q_2 , e.g., point P_2 .

Considering the set Q_0 , with the same rules as above in picking α_k and β_k values, when $\alpha_0 = \omega_{c1}$, $\alpha_1 = 0$, $\beta_0, \beta_1, \beta_2 \rightarrow 0^+$ and $\tau = 0$, the characteristic equation in (15) approaches to

$$F(s, 1) \cong (s^2 + \omega_{c1}^2)(s^2 + 2\zeta\omega_n s + \omega_n^2) \quad (33)$$

which comprises of an oscillating term due to $s^2 + \omega_{c1}^2$, and a stable term due to $s^2 + 2\zeta\omega_n s + \omega_n^2$. Hence the system is marginally stable. This solution point is at the boundary of the set Q_0 , e.g., point at P_0 .

Since our parametric space is actually 6-dimensional, the three points P_0 , P_1 and P_2 in Fig. 8 are indeed the same point, located at the boundary of the intersection area bounded by \widehat{ABC} . This point is therefore a good initial condition point to be used in the optimization scheme.

Furthermore, since \mathcal{C}_8 gives a constraint for ω_{c2} , it therefore makes sense ω_{c2} is slightly less than ω_{c1} for initial

condition. In some sense, this lets the optimization go only in one direction (reduce ω_{c2} further) as iterations take place along the ω_{c2} direction, consistent with Remark 4.

For parameters β_0 , β_1 and β_2 in the numerator, they are assigned with a small positive value ϵ as explained above. This corresponds to a weak controller, which initially guarantees that α_0, α_1 assignment above initially achieves its goal of satisfying Q_0 as well as Q_1 and Q_2 .

References

- Erneux T (2009) Applied delay differential equations, surveys and tutorials in the applied mathematical sciences, vol 3. Springer, New York
- Gu K, Chen J, Kharitonov VL (2003) Stability of time-delay systems. Control Engineering, Birkhäuser, Boston
- Sipahi R, Niculescu SI, Abdallah CT, Michiels W, Gu K (2011) Stability and stabilization of systems with time delay. IEEE Control Syst Mag 31(1):38–65
- Stépán G (1989) Retarded dynamical systems: stability and characteristic functions, Pitman research notes in mathematics series, vol 210. Longman Scientific & Technical Co-publisher John Wiley & Sons Inc, New York
- He Y, Wang QG, Lin C, Wu M (2007) Delay-range-dependent stability for systems with time-varying delay. Automatica 43(2):371–376
- Li X, Gao H (2011) A new model transformation of discrete-time systems with time-varying delay and its application to stability analysis. IEEE Trans Autom Control 56(9):2172–2178
- Michiels W, Niculescu SI (2007) Stability and stabilization of time-delay systems: an eigenvalue-based approach. Advances in Design and Control, SIAM, Philadelphia
- Abdallah CT, Dorato P, Benites-Read J, Byrne R (1993) Delayed positive feedback can stabilize oscillatory systems. In: American control conference, 1993. pp 3106–3107
- Bozorg M, Davison EJ (2006) Control of time delay processes with uncertain delays: time delay stability margins. J Process Control 16(4):403–408
- Emami T, Watkins JM (2009) Robust performance characterization of PID controllers in the frequency domain. WSEAS Trans Syst Control 4(5):232–242
- Silva GJ, Datta A, Bhattacharyya SP (2001) PI stabilization of first-order systems with time delay. Automatica 37(12):2025–2031
- Silva GJ, Datta A, Bhattacharyya SP (2002) New results on the synthesis of PID controllers. IEEE Trans Autom Control 47(2):241–252
- Yi S, Nelson P, Ulsoy A (2007) Delay differential equations via the matrix Lambert W function and bifurcation analysis: application to machine tool chatter. Math Biosci Eng 4(2):355
- Yi S, Nelson PW, Ulsoy AG (2013) Proportional-integral control of first-order time-delay systems via eigenvalue assignment. IEEE Trans Control Syst Technol 21(5):1586–1594
- Michiels W, Vyhldal T (2005) An eigenvalue based approach for the stabilization of linear time-delay systems of neutral type. Automatica 41(6):991–998
- Ramírez A, Mondié S, Garrido R, Sipahi R (2015) Design of proportional-integral-retarded (PIR) controllers for second-order LTI systems. IEEE Trans Autom Control 61(6):1688–1693
- Vyhldal T, Michiels W, McGahan P, Zitek P (2009) Stability impact of small delays in state derivative feedback applied to vibration suppression. Control Eng Pract 17(3):382–393

18. Sipahi R (2014) Delay-margin design for the general class of single-delay retarded-type LTI systems. *Int J Dyn Control* 2(2):198–209
19. Delice II, Sipahi R (2009) Exact upper and lower bounds of crossing frequency set and delay independent stability test for multiple time delayed systems. In: 8th IFAC workshop on time-delay systems. Sinaia, Romania
20. Delice II, Sipahi R (2012) Delay-independent stability test for systems with multiple time-delays. *IEEE Trans Autom Control* 57(4):963–972
21. Li X, Gao H, Gu K (2016) Delay-independent stability analysis of linear time-delay systems based on frequency discretization. *Automatica* 70:288–294
22. Nia PM (2014) Control-parameter-space classification for delay-independent-stability of linear time-invariant time-delay systems; theory and experiments. PhD Dissertation, Northeastern University, Boston
23. Nia PM, Sipahi R (2013) Controller design for delay-independent stability of linear time-invariant vibration systems with multiple delays. *J Sound Vib* 332(14):3589–3604
24. Olgac N, Sipahi R (2002) An exact method for the stability analysis of time-delayed linear time-invariant (LTI) systems. *IEEE Trans Autom Control* 47(5):793–797
25. Olgac N, Ergenc AF, Sipahi R (2005) Delay scheduling: a new concept for stabilization in multiple delay systems. *J Vib Control* 11(9):1159–1172
26. Middleton RH, Miller DE (2007) On the achievable delay margin using LTI control for unstable plants. *IEEE Trans Autom Control* 52(7):1194–1207
27. Datko R (1978) A procedure for determination of the exponential stability of certain differential–difference equations. *Q Appl Math* 36:279–292
28. Hale JK, Lunel SMV (1993) Introduction to functional differential equations, applied mathematical sciences, vol 99. Springer, New York
29. Walton K, Marshall JE (1987) Direct method for TDS stability analysis. *IEE Proc D-Control Theory Appl* 134(2):101–107
30. Cooke KL, Van Den Driessche P (1986) On zeroes of some transcendental equations. *Funkcialaj Ekvacioj* 29(1):77–90
31. Sipahi R, Olgac N (2006) Stability robustness of retarded LTI systems with single delay and exhaustive determination of their imaginary spectra. *SIAM J Control Optim* 45(5):1680–1696
32. Rekasius ZV (1980) A stability test for systems with delays. In: Proceedings of joint automatic control conference, San Francisco, CA Paper No. TP9-A
33. Gumussoy S, Michiels W (2012) Root locus for SISO dead-time systems: a continuation based approach. *Automatica* 48(3):480–489
34. Sipahi R, Olgac N (2003) Degenerate cases in using the Direct Method. *J Dyn Syst Meas Control* 125(2):194–201
35. Ogata K (2001) Modern control engineering, 4th edn. Prentice Hall, Upper Saddle River
36. Coleman TF, Zhang Y (2008) Optimization toolbox 4 [user's Guide]. MathWorks. http://www.mathworks.co.uk/help/pdf_doc/optim/optim_tb.pdf. Accessed 12 Aug 2016
37. Vyhřídál T, Zítek P (2014) QPmR-quasi-polynomial root-finder: Algorithm update and examples. In: Delay systems. Springer International Publishing, Cham, Switzerland, pp 299–312
38. Zhou K, Doyle JC (1998) Essentials of robust control. Prentice hall modular series for engineering. Prentice Hall, Upper Saddle River
39. Curtiss DR (1918) Recent extensions of descartes' rule of signs. *Ann Math* 19(4):251–278
40. Gelfand IM, Kapranov MM, Zelevinsky AV (1994) Discriminants, resultants, and multidimensional determinants. Mathematics, Birkhäuser, Boston