Exploring Delay-Based Controllers. A Comparative Study for Stabilizing Angular Positioning

Diego Torres-García, Julián-Alejandro Hernández-Gallardo, César-Fernando Méndez-Barrios, Silviu-Iulian Niculescu

Exploring Delay-Based Controllers. A Comparative Study for Stabilizing Angular Positioning

Diego Torres-García Julián-Alejandro Hernández-Gallardo César-Fernando Méndez-Barrios and Silviu-Iulian Niculescu

Abstract This study aims to perform a comparison of different control schemes, which are characterized by their *delay-based* nature, where the time delay is considered as a parameter of the control law. To facilitate this research, we look at a second-order linear time-invariant (LTI) single-input single-output (SISO) system as our test bench. Moreover, we propose a simple but efficient methodology to compute the *delay margin* for these control schemes, which exploits the continuity of the characteristic roots with respect to the parameters of the characteristic function of the closed-loop system. Furthermore, we evaluate the performance and robustness of the closed-loop system by numerical simulations and physical experiments. By conducting this in-depth analysis, we gain valuable insight into the strengths and limitations of each control scheme, providing a clearer understanding of their real-world applicability.

1 Introduction

Proportional-Integral-Derivative (PID) controllers have long been recognized as one of the most widely used controller structures in various industries [1, 2]. Over several decades, they have consistently demonstrated their effectiveness in dealing with common industrial problems [3]. In parallel, Delay-Based (DB) controllers, such as the Proportional Minus Delay (PMD) [4, 5], the Proportional-Integral-Retard

Diego Torres-García · César-Fernando Méndez-Barrios · Julián-Alejandro Hernández-Gallardo
Universidad Autónoma de San Luis Potosí (UASLP), Facultad de Ingeniería, Dr. Manuel Nava No.8, San Luis Potosí, S.L.P., México e-mail: {diego.torres,fernando.barrios}@uaslp.mx,julian_@ieee.org,

Diego Torres-García · Silviu-Iulian Niculescu
Université Paris-Saclay, CNRS, CentraleSupélec, Inria, Laboratoire des Signaux et Systèmes (L2S, UMR CNRS 8506), F-91192, Gif-sur-Yvette, France e-mail: {diego.torres-garcia,silviu.niculescu}@l2s.centralesupelec.fr

(PIR) [6], more recently the so-called proportional-delayed controller (P-δ) [7], to name a few, where a delay h is used as a control parameter, gained in popularity in recent years. All these controllers are of low complexity because they have a reduced number of parameters. However, delay-based controllers are infinite-dimensional and, as noted in the references mentioned above, in some cases, they have certain advantages over finite dimensional controllers but also certain disadvantages in other cases. We refer to [8] for an overview of time-delay systems and their applications to control theory.

Interestingly, a common feature shared by these two control approaches is that a *derivative-like* action can be implemented using a delay-difference operator. Although this approach is commonly employed to incorporate derivative actions, the selection of the delay parameter is often overlooked in the tuning process. Typically, a value is chosen as "sufficiently small" to approximate the derivative accurately. However, it is important to note that the behavior of a closed-loop system strongly depends on the location of the solutions of its characteristic function. Moreover, these roots are continuous functions of the system parameters (see, for instance, [9] in the finite-dimensional case and [10] in the delay case), which implies that the choice of parameters, including, in this case, the delay parameter, can have a significant impact on the roots location and, therefore, on the overall behavior of the system.

With the above discussion in mind, it is natural to consider a control design that incorporates the derivative action using a delay-difference operator, with the delay parameter h being a control parameter. In this note, we explore four different control schemes based on the well-known finite-difference method. The first case aims to approximate the derivative in the simplest manner, given by:

$$\frac{dy(t)}{dt} \approx \frac{y(t) - y(t-h)}{h}.$$

The second structure employs the three-points backward difference [11] and is defined as follows:

$$\frac{dy(t)}{dt} \approx \frac{3y(t) - 4y(t-h) + y(t-2h)}{2h}.$$

Inspired by these schemes and the P-δ control, we also propose a control scheme of the following form:

$$u(t) = \sum_{j=0}^{n} k_j y(t - jh),$$

with $n \in \{1, 2\}$. Unlike backward difference methods, which aim to approximate the derivative, such structures can bear an *average* derivative action but also provide more *freedom* in their design. It is worth mentioning that, although including a time-delay in the closed-loop system makes it infinite-dimensional, the controller remains of *low-complexity* type since we are considering commensurate delays and therefore adding only one (delay) parameter. The objective of this note is to investigate whether one of the structures introduced above offers advantages, in terms of closed-loop system performance and robustness with respect to the system parameters (including

the delay). In particular, to measure the robustness of the proposed controllers with respect to the delay parameter, we propose methodologies to compute the *delay margin* [12, 10, 13], that is the maximum value of the delay h such that the stability of the closed-loop system can be preserved under the assumption that such a property holds for the system free of delays. In other words, we are explicitly computing the first delay interval guaranteeing stability without analyzing if this interval is the only one guaranteeing the stability in closed-loop.

It is worth mentioning that recent research studies have addressed the issue of "improperly-posedness" associated with these types of approximations. Improperly-posedness refers to the potential instability that may arise in a closed-loop system for "small delays" when replacing the derivative action with its delay-difference counterpart. In order to avoid such problems, we specifically focus our comparative study on systems that are properly-posed (for further details, refer to [14, 15]).

2 Preliminaries

In the following, we focus on LTI-SISO systems, which can be represented by a transfer function of the form:

$$H(s) := \frac{k}{s(Ts+1)}, \tag{1}$$

where s represents the Laplace variable, $k \in \mathbb{R}_+$ is the open-loop gain of the plant, and $T \in \mathbb{R}_+$ is the system time-constant. This particular type of plant is of general industrial interest since it captures the dynamics of several industrial processes such as car suspension systems and pendulum oscillation [16], electric motors [17], reactor batch [18], thermal systems, and heating systems or an air conditioning (HVAC) system [17] (see also [19] and the references therein). We also examine the closed-loop system when a Proportional-Derivative (PD) controller is employed, which has a frequency domain representation given by:

$$C_0(s) = k_p + k_d s, \tag{2}$$

where k_p and k_d represent the proportional and derivative gains, respectively. Such a control scheme is well-suited for the dynamics of the plants under consideration. Firstly, these plants are *integral* in nature, which means they naturally achieve a steady-state error of zero without requiring an integral action in the control scheme. Secondly, incorporating a *derivative action* in a control process is commonly necessary to enhance the speed of such processes, which is a common requirement for this type of plant. In this study, we propose replacing the controller $C_0(s)$ with the DB controllers in Table 2, where $h > 0$ (i.e., $h \in \mathbb{R}_+$) is a time delay, and it is considered as a control parameter. The controllers C_1 and C_2 correspond to the aforementioned approximation of the derivative action, while C_3 and C_4 are inspired

Table 1: DB Control schemes under consideration

Delay-difference derivative	P-δ-alike controller
$C_1(s) = k_p + \frac{k_d}{h}(1 - e^{-hs})$	$C_3(s) = k_0 + k_1 e^{-hs}$
$C_2(s) = k_p + \frac{k_d}{h}\left(\frac{3}{2} - 2e^{-hs} + \frac{1}{2}e^{-2hs}\right)$	$C_4(s) = k_0 + k_1 e^{-hs} + k_2 e^{-2hs}$

by the P-δ controller and aims to simplify the stability analysis while retaining the main properties of a derivative action.

When considering a classical unity feedback loop, the characteristic function of the closed-loop system can be represented as $\Delta_{hi} : \mathbb{C} \to \mathbb{C}$:

$$\Delta_{hi}(s; h) = s(Ts + 1) + kC_i(s), \quad i \in \{1, 2, 3, 4\}, \tag{3}$$

where $C_i(s)$ represents the controller, depending on the subscript i. The characteristic function is a quasi-polynomial and exhibits an infinite number of characteristic roots. In this regard, the closed-loop system is considered exponentially stable if and only if all the roots of $\Delta_{hi}(s; h)$ are located in the open left-half plane of the complex plane ($\mathbb{C}_-$). It is also worth mentioning that these solutions are continuous functions of the delay parameter h.

With the above discussion in mind, consider the following definitions:

Definition 1 (**Spectrum**) Consider the dynamical system represented by the transfer function (1) with closed-loop characteristic function (3). The *spectrum* of Δ_{hi} is the set $\sigma(\Delta_{hi}) \subset \mathbb{C}$ defined by

$$\sigma\left(\Delta_{hi}\right) := \{s^* \in \mathbb{C} : \Delta_{hi}(s^*; h) = 0\}.$$

Definition 2 (**Delay margin**) Consider the closed-loop characteristic function (3) and its spectrum $\sigma(\Delta_{hi})$. Under the assumption that the closed-loop system free of delay is exponentially stable, the *delay margin* $h_{\max}$ is the bound defined by

$$h_{max} := \sup\{\beta \in \mathbb{R}_+ : \sigma\left(\Delta_{\beta i}\right) \subset \mathbb{C}_-\}.$$

Bearing in mind all these definitions, the following section introduces our main results.

3 Main Results

In the following, we aim to compute the delay margin of the closed-loop system for the different controllers under consideration. Since any digital implementation of a PD controller requires a discretization process, the delay margin is of crucial importance.

3.1 Delay-Margin Computation

We have the following propositions:

Proposition 1 ***(Delay margin of the system in closed-loop with*** C_1***):*** *Consider the system* (1) *in closed-loop with the controller* C_1 *and assume that* $k_p \leq \frac{1}{4kT}$*. Then, if* C_0 *is a stabilizing controller, the delay margin* h_{max} *of the closed-loop system is given by:*

$$h_{max} = \frac{2\left(kk_dT\omega^2 - k^2k_dk_p\right)}{k^2k_p^2 - 2kk_pT\omega^2 + T^2\omega^4 + \omega^2}, \tag{4}$$

where $\omega \in \mathbb{R}_+$ *is the smallest value satisfying:*

$$\frac{\pi}{2}(1 - sign(k_d)) = arg\{h_{max}\boldsymbol{i}\omega(T\boldsymbol{i}\omega + 1) + kk_ph_{max} + kk_d\} + h_{max}\omega. \tag{5}$$

Proof Let $\tilde{\Delta}(s;h) := h\Delta_{h1}(s;h)$, which share the same solutions for $h > 0$. Taking $s = \boldsymbol{i}\omega$ and considering $\tilde{\Delta}(\boldsymbol{i}\omega;h) = 0$, it is easy to obtain the following system of equations:

$$k(hk_p + k_d) - kk_d\cos(h\omega) - hT\omega^2 = 0, \tag{6}$$

$$kk_d\sin(h\omega) + h\omega = 0. \tag{7}$$

Note that, as was expected, $h = 0$ is a solution, however, it is not of interest. Hence, by solving for $h \neq 0$ yields to (4). Next, since $s = \boldsymbol{i}\omega$ is a solution when $h = h_{\max}$ we must have:

$$h_{\max}\boldsymbol{i}\omega(T\boldsymbol{i}\omega + 1) + kk_ph_{\max} + kk_d = kk_de^{-h_{\max}\boldsymbol{i}\omega}. \tag{8}$$

Finally, (5) results straightforwardly by considering the argument on both sides of (8). □

Proposition 2 ***(Delay margin of the system in closed-loop with*** C_3***):*** *Consider the system* (1) *in closed-loop with the controller* $C_3(s)$*. If the system is stable for* $h = 0$*, then the delay margin* h_{max} *of the system is given by the smallest value of* $h \neq 0$ *such that:*

$$\cos(h\omega) = \frac{T\omega^2 - kk_0}{kk_1} \wedge \sin(h\omega) = \frac{\omega}{kk_1}, \tag{9}$$

where $\omega \neq 0$ *is given by:*

$$\omega^2 = \frac{2Tkk_0 - 1 \pm \sqrt{1 + 4kT(-k_0 + kk_1^2T)}}{2T^2}. \tag{10}$$

Proof The closed-loop characteristic equation reads as:

$$\Delta_{h3}(s;h) = s(Ts + 1) + k(k_0 + k_1e^{-hs}).$$

Define $f_1(s) := s(Ts + 1 + kk_0)$ and $f_2(s;h) := kk_1e^{-hs}$. Thus, $\Delta_{h3}(s;h) = f_1(s) + f_2(s;h)$. By continuity, we know that if $\Delta_{h3}(s;0)$ is stable, it will remain stable for

a sufficiently small h. Furthermore, if any root crosses from stability to instability, the crossing must occur by the $\boldsymbol{i}\mathbb{R}$ axis. Suppose now that $s = \boldsymbol{i}\omega$ is a solution of $\Delta_{h3}(s;h)$ for some $h \in \mathbb{R}^+$, thus $s = -\boldsymbol{i}\omega$ is also a solution and we can consider the equation $f_1(\boldsymbol{i}\omega)f_1(-\boldsymbol{i}\omega) - f_2(\boldsymbol{i}\omega;h)f_2(-\boldsymbol{i}\omega;h)$, which bears:

$$k^2k_0^2 - k^2k_1^2 + \omega^2(1 - 2kk_0T) + T^2\omega^4 = 0. \tag{11}$$

Direct computations yield to (10). Next, since $\Delta_{h3}(\boldsymbol{i}\omega;h) = 0$, both the real and the imaginary parts have to be zero. From this observation, it is easy to obtain (9). □

Proposition 3 ***(Delay margin of the system in closed-loop with*** C_4***):*** *Consider the system* (1) *in closed-loop with the controller* $C_4(s)$*. If the system is stable for* $h = 0$*, then the delay margin* h_{max} *of the system is given by the smallest value of* $h \neq 0$ *such that:*

$$\cos(h\omega) = \Re\left\{-\frac{k^2(k_0-k_2)(k_0+k_2) - 2kk_0T\omega^2 + T^2\omega^4 + \omega^2}{kk_1(k(k_0-k_2) - \omega(T\omega + \boldsymbol{i}))}\right\}, \tag{12}$$

and:

$$\sin(h\omega) = \Im\left\{\frac{k^2(k_0-k_2)(k_0+k_2) - 2kk_0T\omega^2 + T^2\omega^4 + \omega^2}{kk_1(k(k_0-k_2) - \omega(T\omega + \boldsymbol{i}))}\right\}, \tag{13}$$

hold, where $\omega \neq 0$ *is given by:*

$$\omega^2 = \sqrt{x}, \tag{14}$$

with $x \neq 0$ *being the smallest real solution of:*

$$ax^4 + bx^3 + cx^2 + dx + e = 0,$$

where:

$$\begin{aligned}
a &= T^4,\\
b &= 2T^2 - 4kk_0T^3,\\
c &= 6k^2k_0^2T^2 - k^2k_1^2T^2 - 2k^2k_2^2T^2 - 4kk_0T + 1,\\
d &= -4k^3k_0^3T + 2k^3k_0k_1^2T + 4k^3k_0k_2^2T - 2k^3k_1^2k_2T + 2k^2k_0^2 - k^2k_1^2 - 2k^2k_2^2,\\
e &= k^4k_0^4 - k^4k_0^2k_1^2 - 2k^4k_0^2k_2^2 + 2k^4k_0k_1^2k_2 - k^4k_1^2k_2^2 + k^4k_2^4 + x^2.
\end{aligned}$$

Proof The closed-loop characteristic equation reads:

$$\Delta_{h4}(s;h) = s(sT+1) + kk_0 + kk_1e^{-hs} + kk_2e^{-2hs}.$$

Next, define $f_1(s) = s(sT+1) + kk_0$ and $f_2(s;h) = kk_2$. If $s = \boldsymbol{i}\omega$ is a solution of $\Delta_{h4}(s;h)$, then $s = -\boldsymbol{i}\omega$ is also a solution and we can write the following system of equations:

$$\begin{aligned}
F_1(\boldsymbol{i}\omega;h) &= f_1(-\boldsymbol{i}\omega)\Delta_{h4}(\boldsymbol{i}\omega;h) - f_1(\boldsymbol{i}\omega)\Delta_{h4}(-\boldsymbol{i}\omega;h)e^{-h\boldsymbol{i}\omega},\\
F_2(\boldsymbol{i}\omega;h) &= f_1(\boldsymbol{i}\omega)\Delta_{h4}(-\boldsymbol{i}\omega;h) - f_1(-\boldsymbol{i}\omega)\Delta_{h4}(\boldsymbol{i}\omega;h)e^{h\boldsymbol{i}\omega}.
\end{aligned}$$

While this system of equations shares solutions with the characteristic equations, it only presents one exponential term, allowing to repeat the procedure followed for the case of $C_3(s)$. □

Finally, if we consider the system in closed-loop with the controller $C_2(s)$, the following algorithm allows to compute its corresponding delay margin:

Data: $\Delta_{h2}(s;h)$, h_0, δh
Result: $h_{\max}$
Initialize: flag $\leftarrow 1$, $h_{\text{aux}} \leftarrow h_0$;
do
  QPmR($\Delta_{h2}(s;h_{aux})$);
  if $\Delta_{h2}(s;h_{aux})$ *is stable* **then**
    $h_{\text{aux}} \leftarrow h_{\text{aux}} + \delta h$;
  else
    $h_{\max} \leftarrow h_{\text{aux}}$;
    flag $\leftarrow 0$;
  end
while *flag* $= 1$;

Algorithm 1: Delay margin explicit computation.

Note that in order to verify whether $\Delta_{hi}(s;h_{aux})$ is stable or not, we make use of the QPmR algorithm (see [20] for further details). An implementation of the algorithm in Python 3.8 can be found in the Appendices.

Remark 1 Algorithm 1 facilitates the computation of the delay margin independently of the selected controllers. However, whenever feasible, opting for analytical computation is generally advisable for achieving optimal results.

4 Numerical Results

Through this section we first introduce the system under consideration and then we present some numerical results in the form of the output of the system in closed-loop with the different control schemes that are being presented.

4.1 System: QUBE-Servo 2

The system under consideration is the QUBE-Servo 2 (see Figure 1), a servo motor for which we aim to control its angular position. The transfer function that models the relationship between the output $\Theta(s)[rad]$ and the input $V(s)[V]$ reads:

$$\frac{\Theta(s)}{V(s)} = \frac{k}{s(Ts+1)}, \tag{15}$$

Fig. 1 QUBE-Servo 2 ROTARY SERVO EXPERIMENT.

which correspond to the class of system considered in this study.

4.2 Numerical Simulations

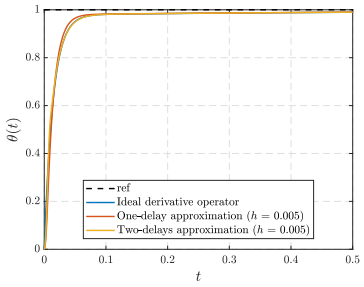
Throughout this section, we present some numerical simulations of the system in closed-loop with the different controllers under study. The plant parameters were obtained via an identification process, $k = 23.65$ and $T = 0.345$, while the controller gains correspond to those indicated in Table 2. Using Algorithm 1 and the methodologies previously mentioned, we obtain the delay margins and the crossing frequency that are shown in Table 2.

Table 2: Delay margin for the different controllers. The different controller gains were chosen such that the delay-free systems have the same right-most root.

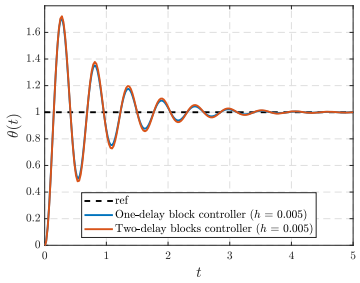
Controller	h_{max}	Crossing frequency ω	Control Gains
$C_1(s)$	$0.07473\ldots$	$43.8984\ldots$	$(k_p, k_d) = (700/473, 1)$
$C_2(s)$	$0.03658\ldots$	$87.1097\ldots$	$(k_p, k_d) = (700/473, 1)$
$C_3(s)$	$0.04410\ldots$	$11.3446\ldots$	$(k_0, k_1) = (1, 1)$
$C_4(s)$	$0.03372\ldots$	$11.2988\ldots$	$(k_0, k_1, k_2) = (1, 0.5, 0.5)$

Note that the smallest tolerance with respect to the delay h corresponds to $h_{max} \approx 0.03$, thus, we can easily consider $h = 0.005$ to avoid losing stability.

From the numerical simulation, it can be observed that the controllers performing an "*approximate*" derivative action achieve a faster response with less overshoot when regulating with an input reference of $\theta = \pi/4$. However, it is worth mentioning that the performance of the PD controller with the derivative approximated via a delay-difference operator can be also achieved by $C_3(s)$ and $C_4(s)$ by simple



(a) System output when in closed-loop with the PD-controller and its two different approximations.



(b) System output when subjected to controllers C_3 and C_4.

Fig. 2: System output when simulating the closed-loop considering the different proposed controllers.

choosing $k_0 = k_p + k_d/h$ and $k_1 = -k_d/h$ in $C_3(s)$, and $k_0 = k_p + 3k_d/(2h)$, $k_1 = -2k_d/h$ and $k_2 = k_d/(2h)$ in $C_4(s)$.

It is important to note that the simulations do not account for any mechanical constraints inherent to the physical system, such as the maximum rotation velocity it can achieve. As a result, the observed speeds in the case of the controllers $C_1(s)$ and $C_2(s)$ may not be achievable in the actual system. To address this issue, the subsequent section presents experiments conducted on the physical plant.

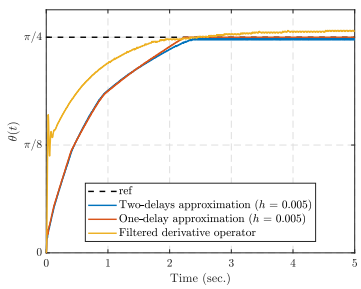
5 Experimental Results

The same experiments that were performed numerically were also done in the real plant. In the case of the physical experiment, not only the system output but also the control output was measured. The implementation does not allow control outputs outside of the interval $[-10, 10]$, and therefore one have to ensure that the control signal is not outside of such a range to assure the expected performance.

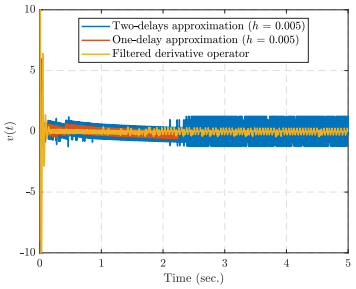
Remark 2 Note that a pure derivative action cannot be implemented in a real plant, it was implemented with the scheme: $k_d N/(1 + N/s)$ where $N = 100$. The larger value of N is the better approximation of the derivative action. This, together with the one-delay approximation, is the most common way to implement a derivative action.

As expected from the numerical simulations, the chosen values of the controller gains as well as the delay value achieve regulation in the real plant. The experimental results can be observed in the following site :

https://sites.google.com/view/jhernandezg/principal/doctorado/publicaciones/delay-difference

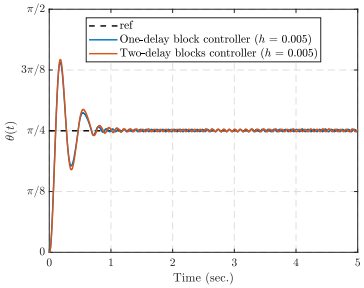


(a) Plant output when in closed-loop with the controllers C_0 implemented with a filter, C_1 and C_2.

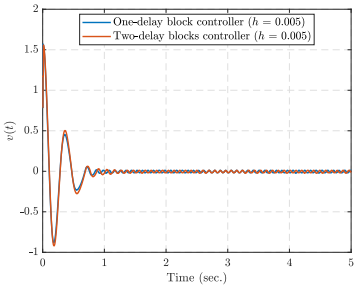


(b) Control output of the filtered PD controller and both approximations.

Fig. 3: System output considering the different proposed controllers.



(a) Plant output when in closed-loop with the controllers C_3 and C_4.



(b) Control output of the controllers C_3 and C_4.

Fig. 4: Real plant output considering the different proposed controllers.

5.1 Verification of the delay margin

The accuracy of the delay margin computations can be readily verified by progressively increasing the value of h within the closed-loop system. As illustrated in Figures 5, a gradual increment in the delay h leads to the emergence of oscillations. These oscillations correspond to the pair of complex conjugate solutions that approach the imaginary axis.

Furthermore, it is evident that the frequency of the oscillations, as depicted in Table 2, is consistent with expectations. Specifically, when considering controllers $C_1(s)$ and $C_2(s)$, the oscillation frequency is higher compared to when using $C_3(s)$ and $C_4(s)$. These results align with the corresponding values of ω obtained from the previous analysis.

The significance of these observed phenomena lies in the understanding that the selection of the delay value for controller implementation cannot be arbitrary. Rather,

it is crucial to account for both margins and crossing frequencies during the design process to mitigate undesirable system behaviors. Depending on the application's requirements, one may prioritize mitigating high-frequency oscillations, minimizing overshoot values, or optimizing the time taken to reach a steady state.

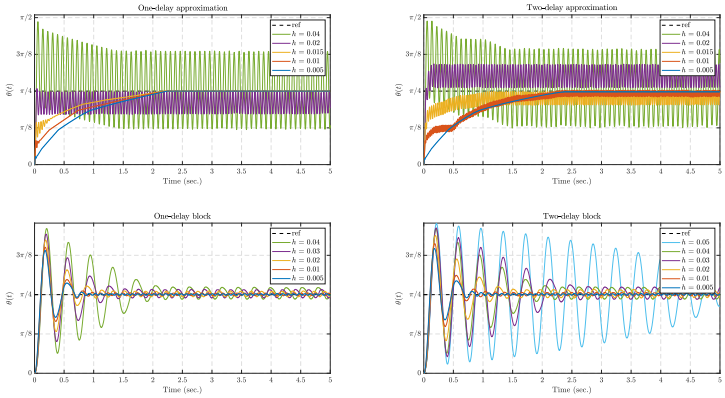


Fig. 5: Experimental proofs of the delay margin.

6 Concluding Remarks

The efficacy of various delay-based controllers has been demonstrated through numerical simulations and real-world experimentation. We also emphasized how the delay-margin may change when changing the control scheme. Moreover, we accentuate the role played by the selection of the delay parameter, and its impact on system stability and performance.

Appendix 1: Algorithm Implementation

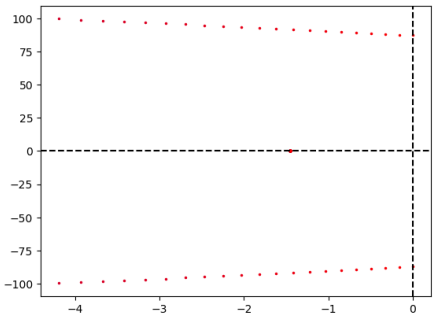
The following code is a Python implementation of Algorithm 1. For the Python implementation of the QPmR algorithm, please refer to the repository available at the following link: https://github.com/DSevenT/QPmR

```
sol = []
i = -1
h = np.linspace(0.000001, 1, 3500)
for tau in h:
    qp = T*s**2+s+k*(kp+kd*(3-4*exp(-tau*s)+exp(-2*tau*s))/(2*tau))
```

```
        r = QPmR(qp, R, 0.1*np.pi/2.5,0.0000001)
        sol.append(r)
        i = i+1
        if all(j < 0 for j in sol[i].real):
            continue
        else:
            print('Critical value of h:' + str(tau))
            break
    figure_1 = plt.figure("Figure 1")
    b = 1
    r = 0
    for n in range(len(sol)):
        plt.plot(sol[n].real, sol[n].imag, 'o', color=[r, 0, b], markersize=1.5)
        if b>0:
            b = b-1/len(sol)
        else:
            b = b+1/len(sol)
        if r <1:
            r = r+1/len(sol)
        else:
            r = r-1/len(sol)
    plt.axvline(x = 0, color = 'k', ls = '--')
    plt.axhline(y = 0, color = 'k', ls = '--')
    plt.show()
```

This implementation generates Figure 6, which illustrates the stability crossing of the system roots as the value of the delay parameter h increases.

Fig. 6 Delay margin detection through Algorithm 1.



Appendix 2: Practical Implementation

The physical implementation of the controller can be done using the QUARC Real-Time Control Software in Simulink (for more information we refer to their website: https://quanser.com/products/quarc-real-time-control-software/), using the scheme depicted on Figure 7.

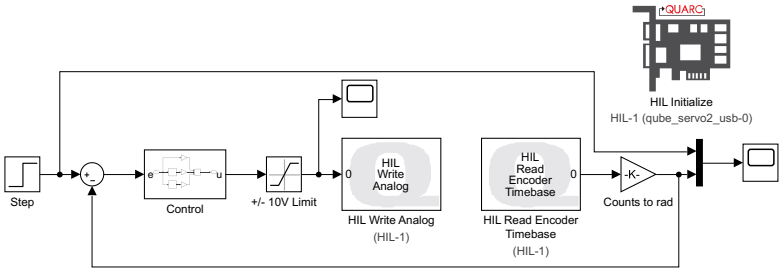


Fig. 7: Implementation of the controller using Simulink.

References

1. K. J. Åström and T. Hagglund. *PID controllers: theory, design, and tuning*. Instrument society of America Research Triangle Park, NC, 2 edition, 1995.
2. A. O'Dwyer. *Handbook of PI and PID Controller Tuning Rules*. Imperial College Press (ICP), London, 3rd edition, 2009.
3. G. J. Silva, A. Datta, and S.P. Bhattachaiyya. Pid stabilization of first-order systems with time delay. *PID Controllers for Time-Delay Systems*, pages 161–190, 2005.
4. H. Suh and Z. Bien. Proportional minus delay controller. *IEEE Transactions on Automatic Control*, 24(2):370–372, 1979.
5. H. Suh and Z. Bien. Use of time-delay actions in the controller design. *IEEE Transactions on Automatic Control*, 25(3):600–603, 1980.
6. A. Ramírez, S. Mondié, R. Garrido, and R. Sipahi. Design of proportional-integral-retarded (pir) controllers for second-order lti systems. *IEEE Transactions on Automatic Control*, 61(6):1688–1693, 2015.
7. J.-E. Hernández-Díez, C.-F. Méndez-Barrios, and S.-I. Niculescu. Proportional-delayed controllers design for lti-systems: a geometric approach. *International Journal of Control*, 91(4):907–925, 2018.
8. R. Sipahi, S.-I. Niculescu, C. Abdallah, W. Michiels, and K. Gu. Stability and stabilization of systems with time delay. *IEEE Control Systems Magazine*, 31(1):38–65, 2011.
9. M. Marden. *Geometry of polynomials*. Number 3. American Mathematical Soc., 1949.
10. W. Michiels and S.-I. Niculescu. *Stability, Control, and Computation for Time-Delay Systems. An Eigenvalue-Based Approach (2. ed.)*. Advances in design and control. SIAM, Philadelphia, 2014.
11. R.J. LeVeque. *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*. SIAM, 2007.
12. K. Gu, V.L. Kharitonov, and J. Chen. *Stability of time-delay systems*. Control Engineering. Birkhäuser Boston, Inc., Boston, MA, 2003.
13. D. Ma and J. Chen. Delay margin of low-order systems achievable by pid controllers. *IEEE Transactions on Automatic Control*, 64(5):1958–1973, 2018.
14. C.-F. Méndez-Barrios, S.-I. Niculescu, A. Martínez-González, and Adrián Ramírez. Characterizing some improperly posed problems in proportional-derivative control. *International Journal of Robust and Nonlinear Control*, 32(18):9452–9474, 2022.
15. D. Torres-García, C.-F. Méndez-Barrios, S.-I. Niculescu, and A. Martínez-González. Delay-difference approximation of pd-controllers. insights into improperly-posed closed-loop systems. *IFAC-PapersOnLine*, 55(40):79–84, 2022.
16. Norman S Nise. *Control systems engineering*. John Wiley & Sons, 2020.

17. S Salivahanan, R Rengaraj, and GR Venkatakrishnan. *Control systems engineering*. Pearson, 2014.
18. Dale E Seborg, Thomas F Edgar, Duncan A Mellichamp, and Francis J Doyle III. *Process dynamics and control*. John Wiley & Sons, 2016.
19. Julián-Alejandro Hernández-Gallardo, Adrian-Josue Guel-Cortez, Emilio J. Gonzalez-Galvan, and César-Fernando Méndez-Barrios. Designing Proportional Delayed Integral Control for Fast Regulation in Second-Order Systems: A Geometric Approach. In *2023 9th International Conference on Control, Decision and Information Technologies (CoDIT)*, pages 01–06, 2023.
20. T. Vyhlidal and P. Zítek. Mapping based algorithm for large-scale computation of quasi-polynomial zeros. *IEEE Transactions on Automatic Control*, 54(1):171–177, 2009.