Appl. Comput. Math., Vol.25, No.3, 2026, pp.431-451
DOI: 10.30546/1683-6154.25.3.2026.431

A MATLAB TOOLBOX FOR PID STABILIZATION IN TIME-DELAY SYSTEMS: DESIGN AND APPLICATIONS

G.-X. FAN[1], X.-G. LI[1,*], Y.-J. CHEN[1]

Abstract. This paper addresses a general scenario in which the three controller gains (k_P, k_I, and k_D) and the delay (τ) are all treated as free parameters. The goal of this paper is to characterize the stability set in the four dimensional (k_P, k_I, k_D, τ)-parameter space, which poses significant theoretical challenges. We introduce in this paper PID-Design-Delay, a MATLAB-based toolbox developed on the basis of a parameter-space approach. The users only need to input the necessary information through a graphical user interface (GUI). Then, the toolbox automatically performs stability analysis. Various numerical examples and two case studies with application backgrounds are provided to illustrate the toolbox.

Keywords: Time-Delay Systems, Stability, PID Control, MATLAB Toolbox.

AMS Subject Classification: 93B51, 93C35, 93D15.

1. Introduction

Time-delay phenomena occur in a wide range of dynamical systems such as transport, propagation, communication, and population dynamics. Time-delay systems have received significant attention, resulting a rich body of literature in many disciplines.

Stability is a fundamental issue in the area of time-delay systems. A time-delay system is infinite-dimensional, as it has infinitely many characteristic roots. Hence, stability analysis is a non-trivial task, especially when determining the delay stability region without conservatism. There are many relevant monographs (see, e.g., [12, 23, 27]).

For a linear time-delay system, its characteristic function belongs to the class of quasipolynomials $f(\lambda, \tau)$, where λ represents the Laplace variable and τ denotes the delay. The delay τ exists in the exponential terms $e^{-\ell\tau\lambda}$, where ℓ is a positive integer. Owing to this property, it is difficult to apply mathematical tools proposed for finite-dimensional systems, and we need to develop specific mathematical tools for time-delay systems. Most existing studies focus on the case where the delay τ is the only free parameter.

Along with advances in theoretical research, some effective software toolboxes have been developed in the past two decades. They can be applied to stability analysis of time-delay systems, both directly and indirectly, and play an increasingly important role in this field.

The QPmR [35], TRACE-DDE [4] and DDE-BIFTOOL [8] toolboxes are very powerful for spectral analysis of time-delay systems. The YALTA toolbox [3] can also be applied to fractional-order delay systems. Using these toolboxes, the root loci for time-delay systems can be numerically generated, which is greatly helpful for the stability analysis, especially for validating theoretical results. A correct conclusion (drawn based on appropriate theoretical criteria) should be consistent with the distribution of characteristic roots. In other words, if a time-delay system is

[1]State Key Laboratory of Synthetical Automation for Process Industries, and School of Information Science and Engineering, Northeastern University, Shenyang 110819, China
e-mail: fangaoxia0113@163.com, chenyujiao116@163.com
*Corresponding author e-mail: masdanlee@163.com
Manuscript received 25 February 2025.

asymptotically stable for a set of delay values, then the infinitely many characteristic roots lie in the open left-half complex plane when delay belongs to the set.

Time-domain and frequency-domain characteristics of concerning time-delay systems may be analyzed using the Control System Toolbox (see [34]) in MATLAB. A wide range of analysis and control tasks for time-delay systems can be performed by using the LambertW_DDE Toolbox (see [38]), which is based on the Lambert W function approach. It is worth noting that both of these toolboxes are designed for systems with fixed delays.

Among various types of controllers, the proportional-integral-derivative (PID) controller is undoubtedly the most commonly used. The PID controller is widely recognized for its simple structure, straightforward implementation, high applicability, and ease of use [1]. It is reported that the PID controllers are employed in over 95% of industrial processes [2].

The PID stabilization usually refers to the problem of designing a PID controller (more precisely, the proportional gain k_P, the integral gain k_I, and the derivative gain k_D) such that the closed-loop system is asymptotically stable. Various methods have been proposed in the literature for finding the stabilization set in the 3-D space of (k_P, k_I, k_D).

In this paper, we focus on the design of PID controllers for time-delay systems. As discussed above, this problem is of considerable practical and theoretical importance. Research on PID stabilization for closed-loop systems with fixed delays has been extensively studied; see, for example [13, 25, 28, 39]. For the case where the delay is treated as a free parameter, far fewer results are available and have mainly been obtained for low-order systems. See, for instance, [32] for first-order controlled plants and [20] for first-order and second-order controlled plants.

Recently, a generic PID stabilization problem has been studied in [19], where the delay τ and the three controller gains (k_P, k_I, k_D) are all treated as free parameters. The corresponding characteristic function $f(\lambda, \tau)$ involves two types of free parameters: controller parameters k_P, k_I, and k_D appearing in the coefficient functions and delay parameter τ appearing in the exponential terms. This paper focuses on the PID stabilization problem involving these four free parameters.

To appropriately address the high-dimensional parameter space of (k_P, k_I, k_D, τ), a new methodology was proposed in [19]. No assumptions are imposed on the controlled plants, and the critical imaginary roots are allowed to have arbitrarily complex asymptotic behaviors. The parameter-space approach is based on analytical techniques, enabling a comprehensive analysis of the system both qualitatively and quantitatively.

However, as far as we know, it remains a challenging task to effectively solve the PID stabilization problem, in practical applications. This paper focuses on developing a comprehensive software toolbox that enables the analytic approach established in [19] to be implemented automatically. Although a parameter-space approach was proposed in [19], users must manually implement the corresponding algorithms, which is not a trivial task. Especially for users with limited familiarity of the parameter-space approach (or even stability of time-delay systems), they may need to spend considerable time to first learn the underlying theory and prerequisites. Our motivation is to present a user-friendly software toolbox, by using which the involved problem can be solved in a much easier manner. It would be extremely important that the toolbox be accessible to by users with different levels of expertise in the stability analysis of time-delay systems.

In this paper, we introduce a toolbox called PID-Design-Delay, developed based on the parameter-space approach proposed in [19][1]. The whole package is freely accessible at `https://github.com/ETLG-Lab/PID-Design-Delay`. A preliminary conference version of this paper can be found in [9].

The stability studies in (k_P, k_I, k_D, τ)-spaces can be rewritten as complete stability analysis in (k_P, k_I, k_D)-spaces w.r.t. τ by applying the auxiliary characteristic function [7] to the complete

[1]More precisely, the theoretical development is based on the frequency-sweeping mathematical framework (see [17, 18]).

stability problem [18], and the whole stability set w.r.t. τ in (k_P, k_I, k_D)-spaces can be obtained analytically. The parameter-space approach is developed by introducing the complete root classification (see, e.g., [36, 37]) on the basis of the frequency-sweeping approach. Thus, the resulting solution is completed from both theoretical and computational perspectives. To the best of the authors' knowledge, the existing toolboxes are unable to provide a full stability characterization in the (k_P, k_I, k_D, τ)-space.

The PID-Design-Delay toolbox was developed with a user-friendly graphical user interface (GUI) [33] implemented in MATLAB, allowing the user to design PID controllers for closed-loop systems with unknown delays. First, the toolbox does not require the user to have extensive knowledge of the stability problem of time-delay systems. Second, the toolbox is equipped with a GUI to facilitate user interaction. Third, the toolbox can generate the analytic quantitative results, as well as plot the stability figure. The stability results can be understood intuitively by the stability figure.

The remainder of this paper is organized as follows. Section 2 provides preliminaries on the PID stabilization problem for time-delay systems. Section 3 describes the basic operation of the PID-Design-Delay toolbox, and Section 4 describes advanced parameter settings through the menu bar. Two case studies are given in Section 5. Finally, Section 6 concludes the paper.

2. Preliminaries

In this section, we review the fundamental notions related to the PID stabilization problem for time-delay systems. We focus on a closed-loop system, as depicted in Fig.1, which consists of a controlled plant and a PID controller.

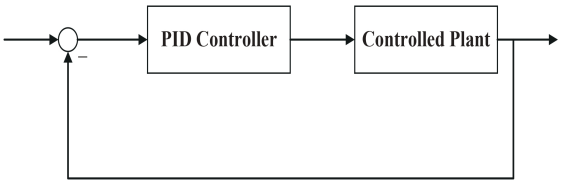


Figure 1. Block diagram of PID control system.

The transfer function that defines the controlled plant can be expressed as:

$$H_0(\lambda) = \frac{H_{\mathrm{N}}(\lambda)}{H_{\mathrm{D}}(\lambda)},$$

where λ denotes the Laplace variable, and the polynomials $H_{\mathrm{D}}(\lambda)$ and $H_{\mathrm{N}}(\lambda)$ are co-prime with real coefficients. Without loss of generality, these polynomials can be written as:

$$H_{\mathrm{D}}(\lambda) = a_0 + a_1\lambda + \cdots + a_{\mathrm{n}}\lambda^{\mathrm{n}}, a_{\mathrm{n}} \neq 0,$$
$$H_{\mathrm{N}}(\lambda) = b_0 + b_1\lambda + \cdots + b_{\mathrm{m}}\lambda^{\mathrm{m}}, b_{\mathrm{m}} \neq 0,$$

with $\mathrm{n} \geq \mathrm{m}$.

In practice, the control loop typically experiences a delay, denoted as τ, leading to the actual transfer function being represented as $H_0(\lambda)e^{-\tau\lambda}$.

The transfer function of the PID controller is expressed as:

$$k_P + \frac{k_I}{\lambda} + k_D\lambda, \tag{1}$$

where k_P, k_I, and k_D denote the proportional, integral, and derivative gains, respectively.

For the above closed-loop system, the characteristic function is given by:

$$f(\lambda, \tau) = H_{\mathrm{D}}(\lambda)\lambda + H_{\mathrm{N}}(\lambda)(k_I + k_P\lambda + k_D\lambda^2)e^{-\tau\lambda}. \tag{2}$$

The closed-loop system is considered asymptotically stable if and only if all characteristic roots of (2) lie in the open left half-plane [10].

In the case where $\lambda = 0$ is a characteristic root (i.e., $b_0 = 0$), the closed-loop system cannot be asymptotically stable for any $\tau \geq 0$, so the toolbox does not specifically consider the trivial case.

Most existing researches on stability problems can be divided into two categories: The τ-decomposition problem [16] and the D-decomposition problem [26]. The study of τ-decomposition problem is for the case where the delay is free and other parameters are fixed. In contrast, the study of D-decomposition problem is for the case where the delay is fixed and some other parameters are free.

As previously mentioned, we treat the three controller gains (k_P, k_I, and k_D) as well as the delay (τ) in the closed-loop system as free parameters. In our opinion, it is a mixed τ-decomposition problem and D-decomposition problem.

Few studies have investigated the mixed τ-decomposition and D-decomposition method, and it is mainly used to determine delay margin and the generalized delay margin in studies on PID stabilization. In the following, we briefly review two important concepts.

A time-delay system is considered asymptotically stable if

$$\tau \in [0, \bar{\tau}),$$

where $\bar{\tau}$ is known as the delay margin [12].

The notion of delay margin is widely used in studying the stability of time-delay systems (see, e.g., [14, 24, 29–31]). Additionally, it is extensively applied in the studies on PID stabilization (see, e.g., [20, 22, 32]).

However, various examples are given in [19] to show that a plant may be stabilized with more than one interval including or excluding 0. In such cases, the notion inevitably involve conservatism. Therefore, a more general notion needs to be introduced.

A time-delay system is considered asymptotically stable if

$$\tau \in [\underline{\tau}_1, \bar{\tau}_1) \cup \cdots \cup (\underline{\tau}_s, \bar{\tau}_s), 0 = \underline{\tau}_1 < \bar{\tau}_1 < \cdots < \underline{\tau}_s < \bar{\tau}_s,$$

or if

$$\tau \in (\underline{\tau}_1, \bar{\tau}_1) \cup \cdots \cup (\underline{\tau}_s, \bar{\tau}_s), 0 \leq \underline{\tau}_1 < \bar{\tau}_1 < \cdots < \underline{\tau}_s < \bar{\tau}_s,$$

where $\bar{\tau}_s$ is referred to as the generalized delay margin [19].

Despite a fairly long research history on the PID stabilization problem for time-delay systems, understanding an effective method to solve the problem remains difficult for most people.

First, a time-delay system has an infinite number of characteristic roots due to its infinite-dimensional nature, making it difficult to analytically study its stability. Furthermore, the PID stabilization problem is a mixed τ-decomposition problem and D-decomposition problem, involving two types of free parameters: Controller gains and delays. Consequently, this stabilization problem is considerably more complex than a conventional stability problem.

Based on the authors' research and teaching experience, it is quite challenging to learn an analytical approach (e.g., the one proposed in [19]) for the PID stabilization problem, or it would require a significant amount of time.

The underlying theoretical approach of the PID-Design-Delay toolbox, which relies on the analysis of characteristic roots, falls within the frequency-domain framework. In this framework, systems with time-invariant but uncertain parameters can be addressed. However, systems with time-varying parameters generally require a time-domain framework. One may refer to [12] for the distinctions and connections between frequency-domain and time-domain frameworks.

3. Basic operation of PID-Design-Delay toolbox

The PID-Design-Delay toolbox is developed to analyze the stability set in the parameter space of (k_P, k_I, k_D, τ). It is composed of three modes: The proportional-integral (PI) mode, the proportional-derivative (PD) mode, and the proportional-integral-derivative (PID) mode, applicable respectively to the scenarios where the PI, PD, or PID controllers are employed.

One can easily input the necessary information to the toolbox through a graphical user interface (GUI). The PID-Design-Delay toolbox completes the stability analysis, and the results are presented in the MATLAB command window, along with a visualization in the 'Stability Set' figure. In this section, we introduce the basic procedure. We provide Examples 1-3 (for the PI, PD, and PID modes, respectively), which serve as quick-start examples.

3.1. **Starting PID-Design-Delay toolbox.**

To execute the PID-Design-Delay toolbox, first ensure that the toolbox folder is added to the MATLAB path or located in the current folder. Next, enter 'PIDCDTDS' in the MATLAB command window. Alternatively, the toolbox can be run by opening the 'PIDCDTDS.m' file in MATLAB and clicking the 'Run' button. If the toolbox reports that the folder is not in the MATLAB path, click either the 'Change Folder' or 'Add to Path' button.

Once the PID-Design-Delay toolbox is launched, a dialog box appears (as shown in Fig.2), prompting the user to select among the PI, PD, or PID controller types. Depending on the user's selection, a corresponding dialog box will appear: The one for the PI mode (Fig.3(a)), the PD mode (Fig.3(b)), or the PID mode (Fig.3(c)).

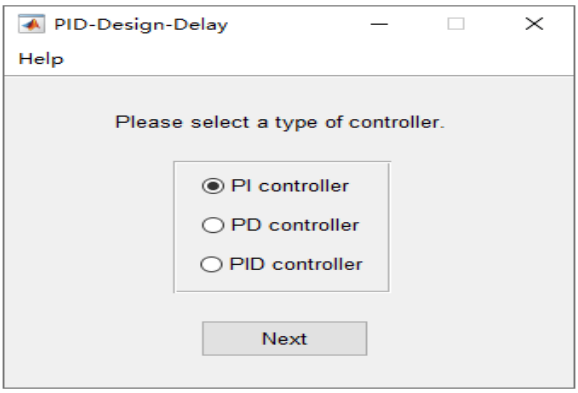


Figure 2. Dialog box for selecting controller type.

3.2. **PI mode.**

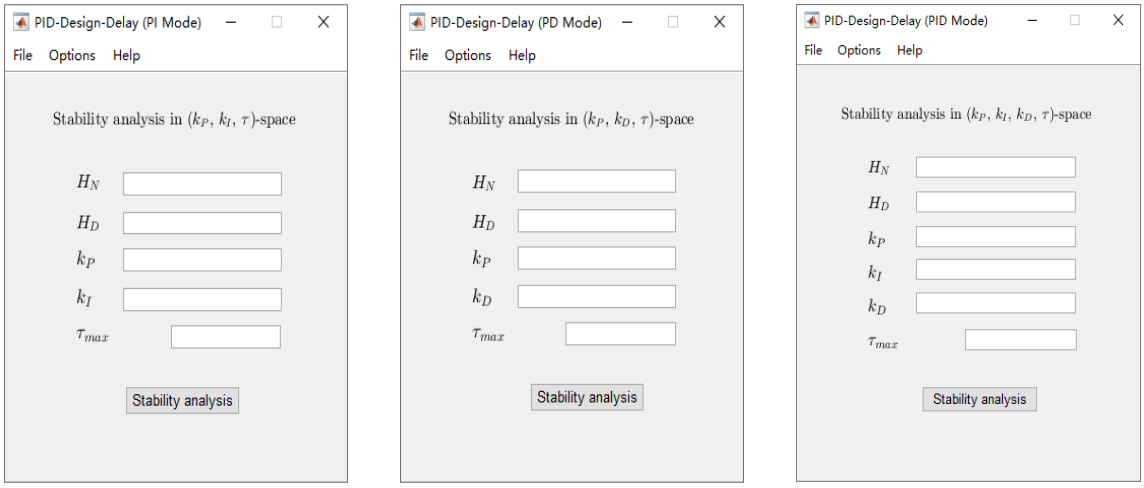
When the 'PI controller' mode is selected, a dialog box apeears as displayed in Fig.3(a).

For the PI controller, the transfer function is as follows

$$k_P + \frac{k_I}{\lambda}. \tag{3}$$

Then, the characteristic function of the closed-loop system is given by:

$$f(\lambda, \tau) = H_{\mathrm{D}}(\lambda)\lambda + H_{\mathrm{N}}(\lambda)(k_I + k_P\lambda)e^{-\tau\lambda}. \tag{4}$$



(a) PI mode (b) PD mode (c) PID mode

FIGURE 3. Dialog boxes for inputting necessary information.

The following necessary information needs to be provided by the user in the dialog box shown in Fig.3(a).
H_{N} **edit box**: Enter the coefficients for the numerator of controlled plant, corresponding to the function $H_{\mathrm{N}}(\lambda)$. The coefficients should be entered in the format as "$[b_{\mathrm{m}} \dots b_0]$".
H_{D} **edit box**: Enter the coefficients for the denominator of controlled plant, corresponding to the function $H_{\mathrm{D}}(\lambda)$. The coefficients should be input in the format as "$[a_{\mathrm{n}} \dots a_0]$".
k_P **edit box**: Specify a range for k_P along with the corresponding step size. The input format is "start point : step length : end point".
k_I **edit box**: Specify a range for k_I with the associated step size. The input format is "start point : step length : end point".
τ_{max} **edit box**: Enter a positive value for τ_{max}. The delay τ is treated as a free parameter within the range $\tau \in [0, \tau_{max}]$ by the toolbox.

Remark 1. When the mouse pointer hovers over the 'Stability analysis' button, a tooltip string (Fig.4) will pop up. This message provides helpful guidance for first-time users and will be further clarified with an example (Example 5).

After all the above information has been entered, press 'Stability analysis'. The PID-Design-Delay toolbox will perform the corresponding analysis, which may take some time. Once completed, the stability results will be presented in both the MATLAB command window and the 'Stability Set' figure.

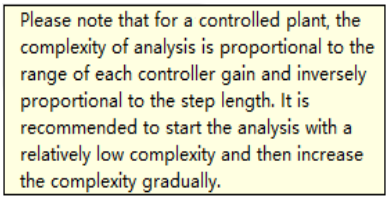


FIGURE 4. Tooltip string associated with 'Stability analysis' button.

The PI controller (3) fails to achieve stabilization in the following two specific cases.

When $\deg(H_{\mathrm{D}}(\lambda)) < \deg(H_{\mathrm{N}}(\lambda))$ and the PI controller is used, the closed-loop system described in (4) is of the advanced type. Accordingly, the closed-loop system definitely cannot be asymptotically stable. Corresponding prompts will be displayed in red in the command window, as illustrated in Fig.5.

When $\deg(H_{\mathrm{D}}(\lambda)) = \deg(H_{\mathrm{N}}(\lambda))$ and the PI controller is used, the closed-loop system described in (4) is of the neutral type. The system stability condition in this case is given by:

$$|k_P| < |\frac{a_{\mathrm{n}}}{b_{\mathrm{m}}}|. \tag{5}$$

The associated indications will be given in the command window, if condition (5) is violated.

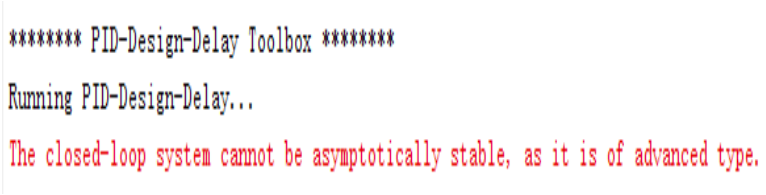


FIGURE 5. Informational messages for the advanced type.

Example 1. Apply the PI controller to the following transfer function of the controlled plant

$$\frac{\lambda - 2}{\lambda - 0.5}.$$

Assume that the stability analysis focuses on the domain $(k_P, k_I, \tau) \in [-2, -0.01] \times [-1.5001, -0.0001] \times [0, 100]$, with the step lengths of 0.01 for both k_P and k_I.

Enter "[1 -2]" in the H_{N} edit box, "[1 -0.5]" in the H_{D} edit box, "-2:0.01:-0.01" in the k_P edit box, "-1.5001:0.01:-0.0001" in the k_I edit box, and "100" in the τ_{max} edit box.

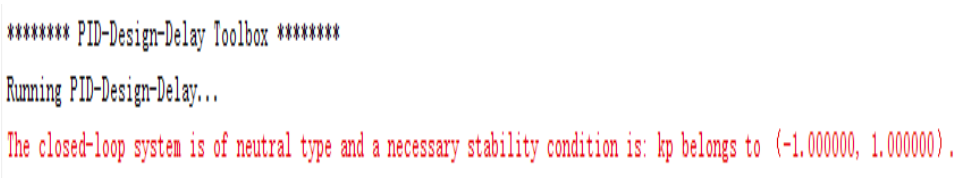


FIGURE 6. Informational message for Example 1.

As the closed-loop system is neutral under the PI controller, the controller gain k_P needs to satisfy the condition (5). After the 'Stability analysis' button is clicked, the informational message of Fig.6 appears to prompt the user to re-enter a valid range for the controller gain k_P.

Accordingly, modify the input in the k_P edit box to "-0.99:0.01:-0.01" ($k_P \in [-0.99, -0.01]$), while leaving all other input values unchanged. Clicking the 'Stability analysis' button starts the analysis within the toolbox. Upon completion, the stability results will be displayed in both the command window and the 'Stability Set' figure. The procedure for applying the PID-Design-Delay toolbox in Example 1 is illustrated in Fig.7. □

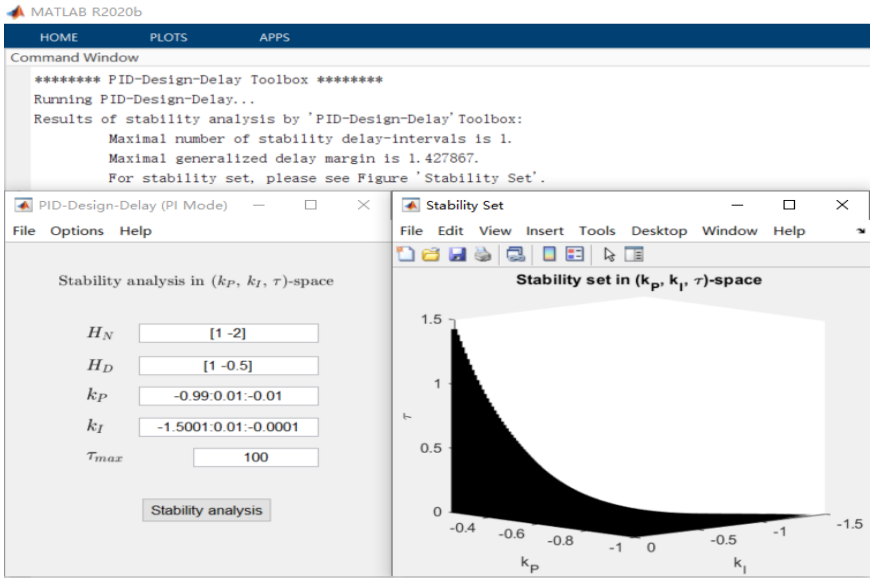


FIGURE 7. Application of PID-Design-Delay toolbox to Example 1.

3.3. **PD mode.** When the 'PD controller' mode is selected, the corresponding dialog box will appear, as shown in Fig.3(b).

For the PD controller, the transfer function is as follows

$$k_P + k_D\lambda. \tag{6}$$

The characteristic function of the closed-loop system is expressed as:

$$f(\lambda,\tau) = H_{\mathrm{D}}(\lambda) + H_{\mathrm{N}}(\lambda)(k_P + k_D\lambda)e^{-\tau\lambda}. \tag{7}$$

The PD controller defined in (6) cannot stabilize the system in the following two specific cases.

When $\deg(H_{\mathrm{D}}(\lambda)) = \deg(H_{\mathrm{N}}(\lambda))$ and a PD controller is used, the closed-loop system described in (7) is of the advanced type. In this case, the closed-loop system definitely cannot be asymptotically stable. As a result, appropriate indications will be provided in the command window, as shown in Fig.5.

When $\deg(H_{\mathrm{D}}(\lambda)) = \deg(H_{\mathrm{N}}(\lambda))+1$ and a PD controller is used, the closed-loop system described in (7) is of the neutral type. The system stability condition in this case is given by:

$$|k_D| < |\frac{a_{\mathrm{n}}}{b_{\mathrm{m}}}|. \tag{8}$$

The associated indications will be given in the command window, if condition (8) is violated.

The PD mode can operate similarly to the PI mode.

Example 2. Apply the PD controller to the following controlled plant

$$\frac{1}{\lambda^2 - 1.4\lambda + 0.48}.$$

The stability analysis focuses on the domain $(k_P, k_D, \tau) \in [-0.5, 0.5] \times [-1.5, 1.5] \times [0, 100]$, with step lengths for k_P and k_D set to 0.001.

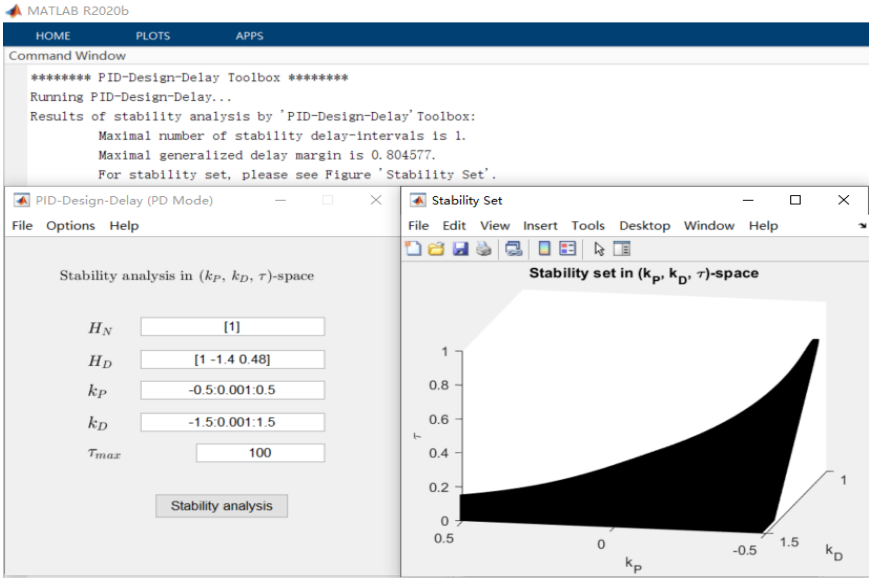


FIGURE 8. Application of PID-Design-Delay toolbox to Example 2.

Enter "[1]" and "[1 -1.4 0.48]" in the H_{N} and H_{D} edit boxes, respectively. Specify the range of k_P and k_D as "-0.5:0.001:0.5" and "-1.5:0.001:1.5", and enter "100" in the τ_{max} edit box.

Clicking the 'Stability analysis' button starts the analysis, after which the stability results appear in the command window and are simultaneously plotted in the 'Stability Set' figure. The procedure for applying the PID-Design-Delay toolbox to Example 2 is shown in Fig.8. □

3.4. **PID mode.** When the 'PID controller' mode is selected, the corresponding dialog box will appear as displayed in Fig.3(c).

The PID controller given as in (1) cannot stabilize the system in the following two cases.

When $\deg(H_{\mathrm{D}}(\lambda)) = \deg(H_{\mathrm{N}}(\lambda))$ and the PID controller is used, the closed-loop system is of the advanced type. An informational message, similar to that in Fig.5, will appear in this case.

When $\deg(H_{\mathrm{D}}(\lambda)) = \deg(H_{\mathrm{N}}(\lambda))+1$ and the PID controller is used, the closed-loop system is of the neutral type. The system stability is governed by the same necessary condition as given in (8). If this condition is violated, then an informational message, similar to that in Fig.6, is displayed in the command window.

Note that the 'Stability Set' figure in the PID mode differs from those in the PI and PD modes. Color information is introduced in the figure to represent the generalized delay margin.

Example 3. Apply the PID controller to the following controlled plant

$$\frac{\lambda - 2}{\lambda^2 - \frac{1}{2}\lambda + \frac{13}{4}}.$$

The stability analysis focuses on the domain $(k_P, k_I, k_D, \tau) \in [0.2, 1] \times [-1, 1] \times [-2, 2] \times [0, 20]$, with the step lengths for k_P, k_I, and k_D set to 0.01.

Enter "[1 -2]" in the H_{N} edit box, "[1 -1/2 13/4]" in the H_{D} edit box, "0.2:0.01:1" in the k_P edit box, "-1:0.01:1" in the k_I edit box, "-2:0.01:2" in the k_D edit box, and "20" in the τ_{max} edit box.

When the PID controller is applied, the closed-loop system is of neutral type. An informative message, similar to that shown in Fig.6, appears in the command window. Accordingly, modify the input in the k_D edit box to "-0.9:0.01:0.9" ($k_D \in [-0.9, 0.9]$), while leaving all other input values unchanged.

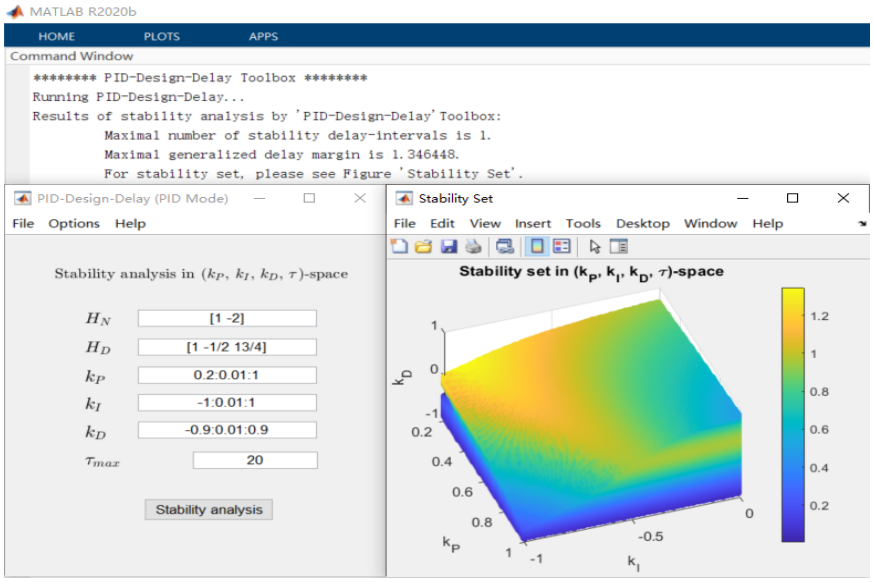


FIGURE 9. Application of PID-Design-Delay toolbox to Example 3.

Once the 'Stability analysis' button is clicked, the toolbox performs the analysis, and the stability results are presented in the command window and the 'Stability Set' figure. The procedure for applying the PID-Design-Delay toolbox to Example 3 is shown in Fig.9. □

3.5. **On Computational time.** The computational time of the PID-Design-Delay toolbox mainly depends on two factors. Generally, it increases with higher system orders as well as smaller step sizes. In the sequel, we present two examples for demonstration.

Example 4. To demonstrate the influence of system order on computational time, we here consider some controlled plants of orders two through six, as listed in Table 1. Here, we employ a PD controller with $k_P \in [-10, 10]$, $k_D \in [-10, 10]$, $\tau \in [0, 50]$, and a step size of 0.1 for both k_P and k_D.

TABLE 1. Computation time (seconds) for Example 4.

Controlled plant	Computation time
$\frac{1}{\lambda^2-4\lambda+3}$	1.552493
$\frac{1}{\lambda^3+2\lambda^2+\lambda-4}$	1.576471
$\frac{1}{\lambda^4+3\lambda^3+2\lambda^2-\lambda+2}$	1.565973
$\frac{1}{\lambda^5+3\lambda^4+4\lambda^3+2\lambda^2-\lambda+3}$	1.631682
$\frac{1}{\lambda^6+2\lambda^5+3\lambda^4+2\lambda^3+2\lambda^2-\lambda+4}$	1.794244

It can be seen that the overall computation time increases, though not significantly, with respect to the system order under the same parameter ranges and step sizes (on a PC with an Intel Core 3.40GHz CPU with 32GB RAM). □

Example 5. To show the effect of step size, we here consider three cases of the system in Example 1.
Case 1: Specify the k_P range as "-0.99:0.01:-0.01" and the k_I range as "-1.5001:0.01:-0.0001".
Case 2: Specify the k_P range as "-0.99:0.005:-0.01" and the k_I range as "-1.5001:0.005:-0.0001".
Case 3: Specify the k_P range as "-0.99:0.001:-0.01" and the k_I range as "-1.5001:0.001:-0.0001".

TABLE 2. Computation time (seconds) and maximal generalized delay margin for Example 5.

	Case 1	Case 2	Case 3
Computation time (s)	5.340897	21.004788	518.107136
Maximal generalized delay margin	1.427867	1.436068	1.436869

Cases 1, 2, and 3 have identical ranges for k_P and k_I, but differ in their step sizes. The computation time and the corresponding maximal generalized delay margin for each case are reported in Table 2. It can be seen that smaller step sizes lead to higher computation time and a more refined analysis of the parameter space. □

In practical applications of the PID-Design-Delay toolbox, we recommend a coarse-to-fine analysis process. One can begin with broad parameter ranges and large step sizes. Then, based on the toolbox outputs, the parameter ranges and step sizes can be gradually reduced.

4. MENU BARS

The menu bars in the PI, PD, and PID modes have the similar functions. For the sake of clarity, we will focus on the menu bar in the PI mode for the subsequent discussion.

The interface features a menu bar with three menus: File, Options, and Help (as illustrated in Fig.3(a)).

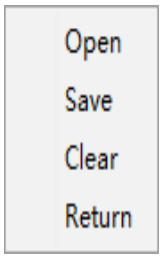


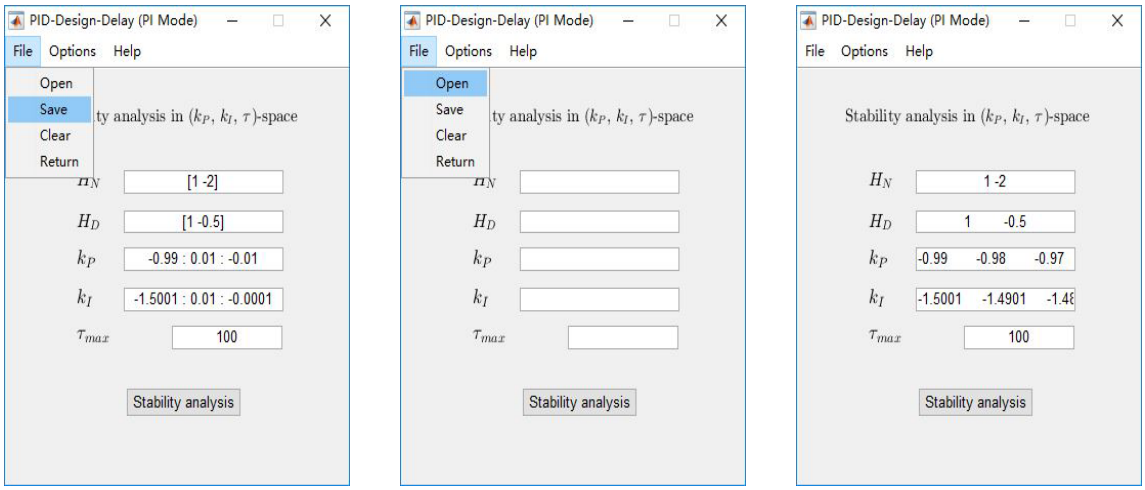
FIGURE 10. File drop-down menu list.

File: The drop-down list for the File menu, displayed in Fig.10, includes four options: Open, Save, Clear, and Return.
Open: Loads an existing MAT-file containing previously saved inputs for the dialog box depicted in Fig.3(a).
Save: Saves the current inputs from the dialog box to a MAT-file.
Clear: Removes all the inputs from the current dialog box.
Return: Takes the user back to the previous dialog box (i.e., that shown in Fig.2).

Example 6. To facilitate the implementation when previously entered information needs to be reused, the 'Save' and 'Open' buttons can be used. For illustration, consider again the task described in Example 1.

Once all the required information has been entered into the dialog box, the input data can be saved into a MAT-file (Example1.mat), as depicted in Fig.11(a) and Fig.12.

When the previously saved inputs need to be loaded, click the 'Open' button (Fig.11(b)) and choose the appropriate saved file (Fig.12). This process restores the earlier input information, as illustrated in Fig.11(c).□



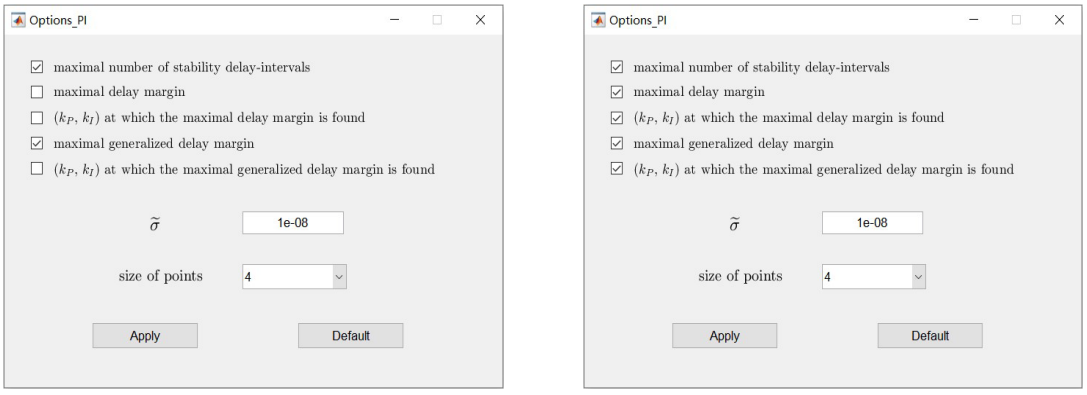
(a) Saving information (b) Opening MAT-file (c) Invoked information

FIGURE 11. Using 'Save' and 'Open' buttons for Example 6.



FIGURE 12. Saved MAT-file for Example 6.

Options: When 'Options' is selected, the dialog box displayed in Fig.13(a) appears. It contains three types of boxes: Five check boxes, an edit box, and a list box.



(a) Default options (b) All check boxes ticked

FIGURE 13. Dialog box for options in PI mode.

The following are the five checkboxes:

- 'maximal number of stability delay-intervals' check box;
- 'maximal delay margin' check box;
- '(k_P, k_I) at which the maximal delay margin is found' check box;
- 'maximal generalized delay margin' check box;
- '(k_P, k_I) at which the maximal generalized delay margin is found' check box.

Upon selecting a check box (denoted by '$\surd$'), the corresponding result will appear in the command window after the stability analysis is finished.

'$\tilde{\sigma}$' edit box: This box is used to set a small positive value for $\tilde{\sigma}$.

Define $\sigma \triangleq \{\min |\mathrm{Re}(\lambda)| : f(\lambda,\tau) = 0\}$. The system is considered to be in or near the state of "having critical imaginary roots when $\tau = 0$" within the domain $\sigma \leq \tilde{\sigma}$.

'size of points' list box: This box allows the user to set the size of points for the 'Stability Set' figure.

'Apply' button: This button activates the modified options.

'Default' button: This button restores the default options.

Based on the default options illustrated in Fig.13(a), the stability results of Examples 1, 2, and 3 are presented in Figs.7-9.

Help: When 'Help' is selected, a brief introduction to the PI mode will be displayed.

Example 7. Consider again the controlled plant from Example 3. Now, we tick all the check boxes as displayed in Fig.13(b). The analytic results are displayed in the command window (see Fig.14), showing that the maximal generalized delay margin exceeds the maximal delay margin.

The maximal delay margin is achieved as 0.935151 at $(k_P = 0.9, k_I = -0.01, k_D = 0.19)$, with the associated stability interval $[0, 0.935151)$.

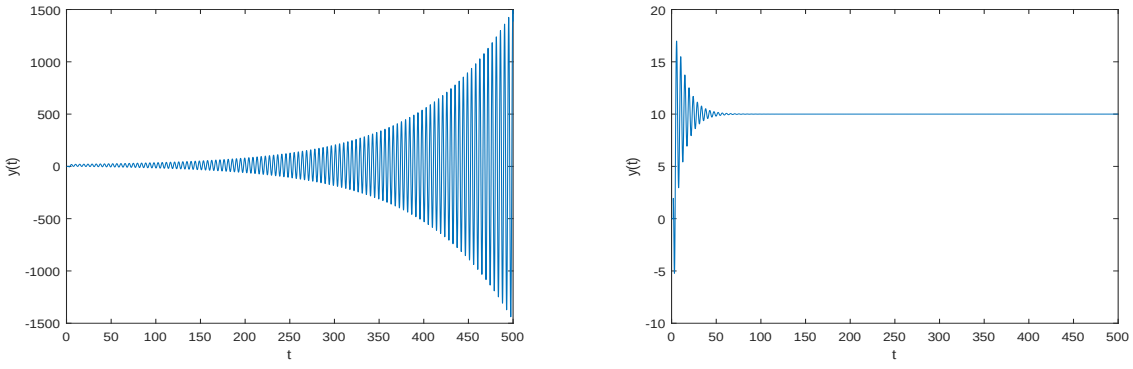
The maximal generalized delay margin is achieved as 1.346448 at $(k_P = 0.2, k_I = -0.48, k_D = 0.19)$, with the associated stability interval $(0.815754, 1.346448)$.

Objectively, it is rather difficult to obtain the above results without using the PID-Design-Delay toolbox.

To validate the results, we present the "$y(t)$ vs. t" plots (where $y(t)$ is the output) for $\tau = 0.8$ (Fig.15(a)), $\tau = 0.9$ (Fig.15(b)), $\tau = 1.3$ (Fig.16(a)), and $\tau = 1.4$ (Fig.16(b)), under the controller parameters $(k_P = 0.2, k_I = -0.48, k_D = 0.19)$. □

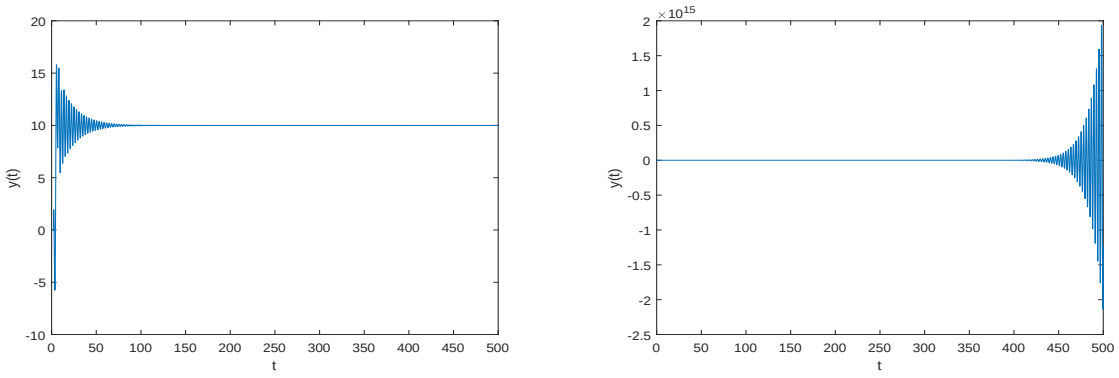
```
Command Window
******** PID-Design-Delay Toolbox ********
Running PID-Design-Delay...
Results of stability analysis by 'PID-Design-Delay' Toolbox:
         Maximal number of stability delay-intervals is 1.
         Maximal delay margin is 0.935151.
         Maximal delay margin is found at (kp=0.900000,ki=-0.010000,kd=0.190000).
         Maximal generalized delay margin is 1.346448.
         Maximal generalized delay margin is found at (kp=0.200000,ki=-0.480000,kd=0.190000).
         For stability set, please see Figure 'Stability Set'.
```

FIGURE 14. Stability results for Example 7.



(a) $\tau = 0.8$ (b) $\tau = 0.9$

FIGURE 15. $y(t)$ vs. t when $\tau = 0.8$ and $\tau = 0.9$ for Example 7.



(a) $\tau = 1.3$ (b) $\tau = 1.4$

FIGURE 16. $y(t)$ vs. t when $\tau = 1.3$ and $\tau = 1.4$ for Example 7.

5. Case studies

In this section, we utilize the PID-Design-Delay toolbox to investigate a direct current (DC) motor drive system with speed control and a multi-agent system.

5.1. **A speed-controlled DC motor drive system.** Consider a DC motor drive system with speed control as shown in Fig.17, where $n(t)$ represents the motor speed and U_n^* denote the reference voltage speed. The system comprises four primary components: A (separately-excited) DC motor, a (thyristor) converter, a tachogenerator, and a PI controller.

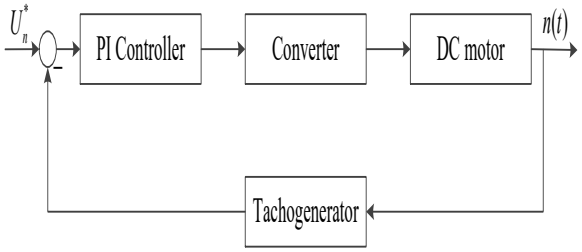


FIGURE 17. Block diagram of DC motor drive system.

The dynamics of the DC motor can be expressed by the following transfer function:

$$\frac{\mathrm{b}_0}{\mathrm{a}_2\lambda^2 + \mathrm{a}_1\lambda^2 + 1},$$

where b_0, a_2, and a_1 represent real numbers.

The thyristor converter is described by the model $\mathrm{c}_0 e^{-\tau\lambda}$, with c_0 denoting the gain and τ representing the phase delay.

The tachogenerator produces a DC voltage as the speed feedback for the DC motor, which can be modeled as a gain (d_0). Further details regarding the DC motor drive system are provided in [15].

In this configuration, the speed $n(t)$ of DC motor is controlled, and the desired speed is $\frac{U_n^*}{\mathrm{d}_0}$. The system achieves the desired speed, i.e., $n(t) \to \frac{U_n^*}{\mathrm{d}_0}$, if and only if the closed-loop system is asymptotically stable.

For the DC motor drive system, its characteristic function can be expressed as:

$$f(\lambda,\tau)=\mathrm{a}_2\lambda^3+\mathrm{a}_1\lambda^2+\lambda+\mathrm{b}_0\mathrm{c}_0\mathrm{d}_0(k_P\lambda+k_I)e^{-\tau\lambda}. \tag{9}$$

By comparing the characteristic functions in (4) and (9), it can be observed that the stability of DC motor system is equivalent to the closed-loop system depicted in Fig.1, where the transfer function of the controlled plant is given by:

$$H_0(\lambda)=\frac{\mathrm{b}_0\mathrm{c}_0\mathrm{d}_0}{\mathrm{a}_2\lambda^2+\mathrm{a}_1\lambda+1}. \tag{10}$$

The PI controller for the above speed-controlled DC motor drive system (10) can be designed through the parameter-space approach (see [5]). Utilizing the PID-Design-Delay toolbox, we directly compute the stability set within the (k_P,k_I,τ)-space.

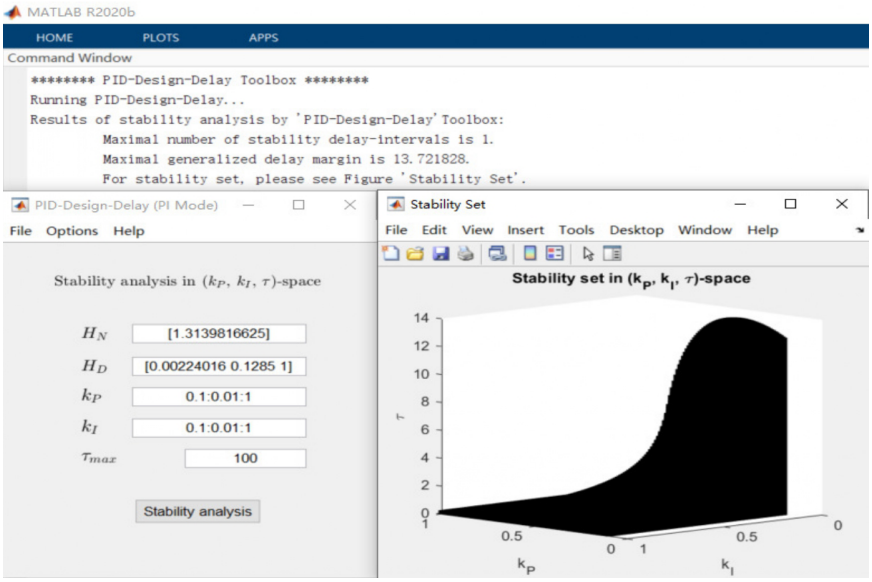


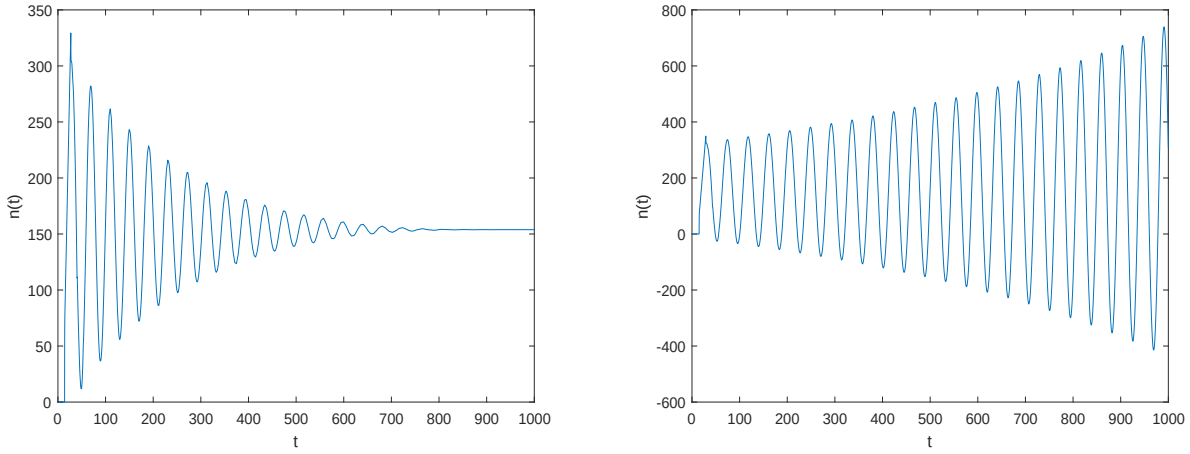
FIGURE 18. Application of PID-Design-Delay toolbox to Example 8.

Example 8. Consider the DC motor system of [5], with the transfer function

$$\frac{1.3139816625}{0.00224016\lambda^2+0.1285\lambda+1}. \tag{11}$$

We treat the controller gains k_P and k_I, and the delay τ, all as free parameters. According to the transfer function (11), we input "[1.3139816625]" and "[0.00224016 0.1285 1]" in the H_{N} and H_{D} edit boxes. We set the ranges for the controller gains by entering "0.1:0.0001:1" and "0.1:0.01:1" in the k_P and k_I edit boxes, and set the maximum delay by entering "100" in the τ_{max} box.

Click the 'Stability analysis' button, and the analysis is automatically performed by the PID-Design-Delay toolbox. The corresponding results are presented in the command window and 'Stability Set' figure (see Fig.18). The speed $n(t)$ converges to $\frac{U_n^*}{\mathrm{d}_0}$ if and only if (k_P,k_I,τ) belongs to the stability set depicted in the 'Stability Set' figure.



(a) $\tau = 13$ (b) $\tau = 14$

FIGURE 19. $n(t)$ vs. t for Example 8.

The stability interval of the closed-loop system is observed to be $[0, 13.721828)$ when $k_P = 0.34$ and $k_I = 0.1$. To verify the above analysis, we provide the "$n(t)$ vs. t" plots (letting $U_n^* = 10$) for $\tau = 13$ and $\tau = 14$ in Fig.19(a) and Fig.19(b), respectively. □

5.2. **A second-order multi-agent system.** Consider a second-order multi-agent system [6]

$$\begin{cases} \dot{x}_i(t) = v_i(t), \\ \dot{v}_i(t) = \alpha v_i(t) + \beta x_i(t) + u_i(t), \end{cases} i = 1, 2, \ldots, N, \tag{12}$$

where $x_i(t)$ and $v_i(t)$ are respectively the position state and velocity state for the ith agent, the parameters α and β are real numbers.

The multi-agent system (12) is said to achieve consensus if under any initial conditions,

$$\lim_{t\to\infty} \|x_i(t) - x_j(t)\| = 0, \ \lim_{t\to\infty} \|v_i(t) - v_j(t)\| = 0,$$

for all $i, j = 1, 2, \ldots, N$.

Here, we select the PD controller, and the consensus control protocol can be represented as:

$$u_i(t) = -k_P \sum_{j\in N_i} a_{ij}(x_i(t-\tau) - x_j(t-\tau)) - k_D \sum_{j\in N_i} a_{ij}(v_i(t-\tau) - v_j(t-\tau)), \tag{13}$$

where $\tau \geq 0$ is the communication delay, and k_P and k_D are respectively the proportional and derivative gains.

Subsequently, we reformulate the consensus problem as a stability problem. To make this subsection self-contained, we review some basic notations and concepts in graph theory [11]. Let $G = (V, \varepsilon, \mathcal{A})$ be an undirected graph of order N, where $V = \{1, 2, \ldots, N\}$ is the set of nodes, $\varepsilon \subset \{(i, j) : i, j \in V\}$ is the set of undirected edges, and $\mathcal{A} = [a_{ij}]$ is an $N \times N$ weighted adjacency matrix. For each edge $(i, j) \in \varepsilon$, agent j can obtain information from agent i. The neighborhood set N_i denotes the set of agents that can receive information from agent i, where $N_i \triangleq \{j|(i, j) \in \varepsilon\}$. For the adjacency matrix $\mathcal{A} = [a_{ij}]$, it follows that $a_{ij} > 0$ and $a_{ii} = 0$ for all $i, j \in N_i$. The in-degree matrix of the graph G is denoted by $D \triangleq \text{diag}(d_1, d_2, \ldots, d_N)$, where $d_i = \sum_j a_{ij}$. The Laplacian matrix is $L = D - \mathcal{A}$, and we denote the eigenvalues of L by $\lambda_i(L)$ with $0 = \lambda_1(L) \leq \lambda_2(L) \leq \cdots \leq \lambda_N(L)$.

We define the state vectors as $x(t) = [x_1(t), x_2(t), \ldots, x_N(t)]^T$, $v(t) = [v_1(t), v_2(t), \ldots, v_N(t)]^T$, and $\varphi(t) = [x^T(t), v^T(t)]^T$. The composite dynamics for the multi-agent system (12) can be expressed as:

$$\dot{\varphi}(t) = (A \otimes I_N)\varphi(t) - (B \otimes L)\varphi(t-\tau),$$

where $A = \begin{bmatrix} 0 & 1 \\ \beta & \alpha \end{bmatrix}$, $B = \begin{bmatrix} 0 & 0 \\ k_P & k_D \end{bmatrix}$, L is the Laplacian matrix, and the symbol $\otimes$ represents the Kronecker product.

Next, we define $\hat{x}_i(t) = x_i(t) - x_1(t)$ and $\hat{v}_i(t) = v_i(t) - v_1(t)$ for $i = 2, 3, \ldots, N$. The error state vector is given by $\hat{\varphi}(t) = [\hat{x}^T(t), \hat{v}^T(t)]^T$, with $\hat{x}(t) = [\hat{x}_2(t), \hat{x}_3(t), \ldots, \hat{x}_N(t)]^T$ and $\hat{v}(t) = [\hat{v}_2(t), \hat{v}_3(t), \ldots, \hat{v}_N(t)]^T$.

Thus, the dynamics of the error can be expressed as:

$$\dot{\hat{\varphi}} = (A \otimes I_{N-1})\hat{\varphi}(t) - (B \otimes \hat{L})\hat{\varphi}(t-\tau), \tag{14}$$

where $\hat{L} = \begin{bmatrix} d_2 & -a_{23} & \cdots & -a_{2N} \\ -a_{32} & d_3 & \cdots & -a_{3N} \\ \vdots & \vdots & \ddots & \vdots \\ -a_{N2} & -a_{N3} & \cdots & d_N \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \begin{bmatrix} a_{12} & a_{13} & \cdots & a_{1N} \end{bmatrix}$.

The consensus of the multi-agent system (12) under the protocol (13) is achieved if and only if the system (14) is asymptotically stable.

For system (14), its characteristic function is given by:

$$f(\lambda, \tau) = \det(\lambda I_{2(N-1)} - (A \otimes I_{N-1}) + (B \otimes \hat{L})e^{-\tau\lambda}).$$

Furthermore, $f(\lambda, \tau)$ can be decomposed as $f(\lambda, \tau) = \prod_{i=2}^{N} f_i(\lambda, \tau)$, where

$$f_i(\lambda, \tau) = \lambda^2 - \alpha\lambda - \beta + (k_D\lambda + k_P)\lambda_i(L)e^{-\tau\lambda}.$$

Define $\mathcal{S}_i$ as the stability τ-set for $f_i(\lambda, \tau)$. Then, the consensus τ-set for system (12) under the PD consensus control protocol (13), denoted by $\mathcal{C}$, is given by:

$$\mathcal{C} = \mathcal{S}_2 \cap \mathcal{S}_3 \cap \ldots \cap \mathcal{S}_N. \tag{15}$$

The PD controller for the above system can be designed by using the parameter-space approach (see [6]). Utilizing the PID-Design-Delay toolbox, we directly compute the stability set within the (k_P, k_D, τ)-space.

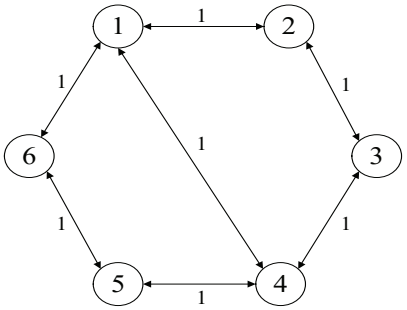


FIGURE 20. Graph for Example 9.

Example 9. Consider the second-order multi-agents system (12) in Example 6 of [21] with $\alpha = 1.1$, $\beta = -0.3$, $k_P = 0.301$, and $k_D = 1.2$. The graph is given in Fig.20, with the adjacency

matrix $\mathcal{A}$ and the Laplacian matrix L given below

$$\mathcal{A}=\begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, L=\begin{bmatrix} 3 & -1 & 0 & -1 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ -1 & 0 & -1 & 3 & -1 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & 0 & -1 & 2 \end{bmatrix}.$$

We have that

$$f_i(\lambda,\tau)=\lambda^2-1.1\lambda+0.3+(k_D\lambda+k_P)\lambda_i(L)e^{-\tau\lambda},\ i=2,3,\ldots,6,$$

where $\lambda_1(L)=0$, $\lambda_2(L)=1$, $\lambda_3(L)=2$, $\lambda_4(L)=\lambda_5(L)=3$, and $\lambda_6(L)=5$.

We apply the PID-Design-Delay toolbox by entering the following values into the corresponding edit boxes: "[1]" for $\lambda_2(L)=1$, "[2]" for $\lambda_3(L)=2$, "[3]" for $\lambda_4(L)=\lambda_5(L)=3$, and "[5]" for $\lambda_6(L)=5$ in the H_N edit box, "[1 -1.1 0.3]" in the H_D edit box, "0.301" in the k_P edit box, "1.2" in the k_D edit box, and "10" in the τ_{max} edit box.

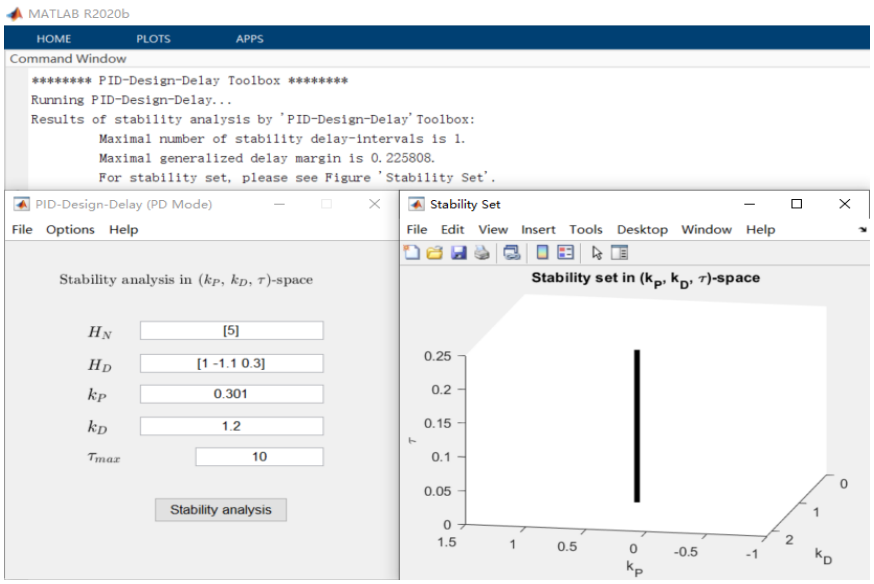


FIGURE 21. Application of PID-Design-Delay toolbox to Example 9.

Then, the PID-Design-Delay toolbox can calculate $\mathcal{S}_i$ as shown in Fig.21 (where $\mathcal{S}_6$ is calculated): $\mathcal{S}_2=[0,0.239509)$, $\mathcal{S}_3=[0,0.433270)$, $\mathcal{S}_4=\mathcal{S}_5=[0,0.337821)$, and $\mathcal{S}_6=[0,0.225808)$.

According to (15), we have that $\mathcal{C}=[0,0.225808)$, which is consistent with the result of [21]. □

6. CONCLUSION

This paper presents PID-Design-Delay, a MATLAB-based toolbox developed for the PID stabilization of time-delay systems. The three controller gains (k_P, k_I, and k_D) and the delay (τ) are treated as free parameters. Developed based on a new parameter-space approach, the toolbox enables automatic stability analysis and systematically determines the whole stability set in the parameter space of (k_P,k_I,k_D,τ). Additionally, the PID-Design-Delay toolbox has demonstrated effective in practical applications, including the stability analysis of DC motors

and multi-agent systems. These applications demonstrate the effectiveness of the toolbox and its potential for broader adoption.

7. Acknowledgments

This work was partially supported by Fundamental Research Funds for the Central Universities (No. N2104007).

References

[1] Ang, K.H., Chong, G., Li, Y. PID control system analysis, design, and technology, *IEEE Trans. Control Syst. Technol.*, Vol.13, No.4, 2005, pp.559-576.

[2] Åström, K.J., Hägglund, T. *PID Controllers: Theory, Design, and Tuning*, Instrument Society of America, North Carolina, 1995, 343 p.

[3] Avanessoff, D., Fioravanti, A.R., Bonnet, C. YALTA: A Matlab toolbox for the H_∞-stability analysis of classical and fractional systems with commensurate delays, In: *5th IFAC Symposium on System Structure and Control*, Vol.46, No.2, 2013, pp.839-844.

[4] Breda, D., Maset, S., Vermiglio, R. TRACE-DDE: A tool for robust analysis and characteristic equations for delay differential equations. In: *Topics in Time Delay Systems: Analysis, Algorithms and Control*, pp.145-155, Springer, Berlin, Vol.388, 2009, 418 p.

[5] Chen, J.-X., Fan, G.-X., Li, X.-G., Mao, Z.-Z. Stability analysis for DC motor drive systems with time delays. In: *34th Chinese Control and Decison Conference (CCDC)*, 2022, pp.13-18.

[6] Chen, Y.-J., Li, X.-G., Fan, G.-X., Zhang, Y. Consensus for a class of multiagent systems with time delays: A systematic study in parameter space, *IEEE Trans. Autom. Control*, Vol.69, No.3, 2024, pp.1796-1803.

[7] Cooke, K.L., Van Den Driessche, P. On zeroes of some transcendental equations, *Funkc. Ekvacioj*, Vol.29, No.1, 1986, pp.77-90.

[8] Engelborghs, K., Luzyanina, T., Roose, D. Numerical bifurcation analysis of delay differential equations using DDE-BIFTOOL, *ACM Trans. Math. Softw.*, Vol.28, No.1, 2002, pp.1-21.

[9] Fan, G.-X., Chen, Y.-J., Li, X.-G., Niculescu, S.-I. PID-Design-Delay: A MATLAB toolbox for stability parameter-space characterization, In: *24th International Conference on System Theory, Control and Computing (ICSTCC)*, 2020, pp.13-18.

[10] Farid, G., Benjamin, C.K. *Automatic Control Systems*, Wiley, New Jersey, 2010, 942 p.

[11] Godsil, C., Royle, G.F. *Algebraic Graph Theory*, Springer, New York, 2001, 443 p.

[12] Gu, K., Kharitonov, V.L., Chen, J. *Stability of Time-Delay Systems*, Birkäuser, Boston, 2003, 356 p.

[13] Hohenbichler, N. All stabilizing PID controllers for time delay systems, *Automatica*, Vol.45, No.11, 2009, pp.2678-2684.

[14] Ju, P., Zhang, H. Further results on the achievable delay margin using LTI control, *IEEE Trans. Autom. Control*, Vol.61, No.10, 2016, pp.3134-3139.

[15] Krishnan, R. *Electric Motor Drives: Modeling, Analysis, and Control*, Prentice Hall, New Jersey, 2001, 652 p.

[16] Lee, M.S., Hsu, C.S. On the τ-decomposition method of stability analysis for retarded dynamical systems, *SIAM J. Control*, Vol.7, No.2, 1969, pp.242-259.

[17] Li, X.-G., Niculescu, S.-I., Cela, A. *Analytic curve frequency-sweeping stability tests for systems with commensurate delays*, Springer, London, 2015, 130 p.

[18] Li, X.-G., Niculescu, S.-I., Cela, A., Zhang, L., Li, X. A frequency-sweeping framework for stability analysis of time-delay systems, *IEEE Trans. Autom. Control*, Vol.62, No.8, 2017, pp.3701-3716.

[19] Li, X.-G., Niculescu, S.-I., Chen, J.-X., Chai, T. Characterizing PID controllers for linear time-delay systems: A parameter-space approach, *IEEE Trans. Autom. Control*, Vol.66, No.10, 2021, pp.4499-4513.

[20] Ma, D., Chen, J. Delay margin of low-order systems achievable by PID controllers, *IEEE Trans. Autom. Control*, Vol.64, No.5, 2019, pp.1958-1973.

[21] Ma, D., Tian, R., Zulfiqar, A., Chen, J., Chai, T. Bounds on delay consensus margin of second-order multiagent systems with robust position and velocity feedback protocol, *IEEE Trans. Autom. Control*, Vol.64, No.9, 2019, pp.3780-3787.

[22] Michiels, W., Engelborghs, K., Vansevenant, P., Roose, D. Continuous pole placement for delay equations, In: *2nd IFAC Workshop on Linear Time Delay Systems*, Vol.33, No.23, 2000, pp.145-150.

[23] Michiels, W., Niculescu, S.-I. *Stability, Control, and Computation for Time-Delay Systems: An Eigenvalue-Based Approach*, SIAM, Philadelphia, 2014, 435 p.

[24] Middleton, R.H., Miller, D.E. On the achievable delay margin using LTI control for unstable plants, *IEEE Trans. Autom. Control*, Vol.52, No.7, 2007, pp.1194-1207.
[25] Moărescu, I.-C., Méndez-Barrios, C.-F., Niculescu, S.-I., Gu, K. Stability crossing boundaries and fragility characterization of PID controllers for SISO systems with I/O delays, In: *Proceedings of the 2011 American Control Conference*, 2011, pp.4988-4993.
[26] Neimark, J. D-subdivisions and spaces of quasi-polynomials, *Prikl. Mat. Mekh.*, Vol.13, No.5, 1949, pp.349-380.
[27] Niculescu, S.-I. *Delay Effects on Stability: A Robust Control Approach*, Springer-Verlag, Heidelberg, 2001, 388 p.
[28] Ou, L.-L., Zhang, W.-D., Yu, L. Low-order stabilization of LTI systems with time delay, *IEEE Trans. Autom. Control*, Vol.54, No.4, 2009, pp.774-787.
[29] Özbay, H. Strongly stabilizing controller design for systems with time delay, *Appl. Comput. Math.*, Vol.23, No.3, 2024, pp.392-403.
[30] Öztürk, D., Özbay, H. Analysis of an extended gene regulatory network model with time delay, *Appl. Comput. Math.*, Vol.22, No.2, 2023, pp.225-245.
[31] Qi, T., Zhu, J., Chen, J. Fundamental limits on uncertain delays: When is a delay system stabilizable by LTI controllers? *IEEE Trans. Autom. Control*, Vol.62, No.3, 2017, pp.1314-1328.
[32] Silva, G.J., Datta, A., Bhattacharyya, S.P. *PID Controllers for Time-Delay Systems*, Birkhäuser, Boston, 2005, 330 p.
[33] The Mathworks. *Creating Graphical User Interfaces*, MathWorks Inc., Natick, 2000.
[34] The Mathworks. *Control System Toolbox*, MathWorks Inc., Natick, 2011.
[35] Vyhlidal, T., Zitek, P. Mapping based algorithm for large-scale computation of quasi-polynomial zeros, *IEEE Trans. Autom. Control*, Vol.54, No.1, 2009, pp.171-177.
[36] Xia, B., Yang, L. *Automated Inequality Proving and Discovering*, World Scientific, Singapore, 2016, 344 p.
[37] Yang, L. Recent advances on determining the number of real roots of parametric polynomials, *J. Symb. Comput.*, Vol.28, No.1, 1999, pp.225-242.
[38] Yi, S., Duan, S., Nelson, P.W., Ulsoy, A.G. The lambert W function approach to time delay systems and the LambertW_DDE Toolbox, In: *10th IFAC Workshop on Time Delay Systems*, Vol.45, No.14, 2012, pp.114-119.
[39] Yi, S., Nelson, P.W., Ulsoy, A.G. Proportional-integral control of first-order time-delay systems via eigenvalue assignment, *IEEE Trans. Control Syst. Technol.*, Vol.21, No.5, 2013, pp.1586-1594.



Gao-Xia Fan - is currently pursuing her Ph.D. degree in Automatic Control at Northeastern University, China. She received her B.Sc. degree in Automatic Control from Heilongjiang University, China, in 2018 and the M.Sc. degree in Automatic Control from Northeastern University, China, in 2020. Her current research focuses on analytic curves and their applications.



Xu-Guang Li - is a Full Professor at Northeastern University, China. He obtained his Ph.D. degree in Automatic Control from Shanghai Jiaotong University, China, in 2008. He was a postdoc at ESIEE, Paris, France, and CNRS-Supelec, France, in 2008 and 2009. His main research interests include the stability of time-delay systems, analytic curves, and robotic fine manipulation.



Yu-Jiao Chen - is currently pursuing her Ph.D. degree in Automatic Control at Northeastern University, China. She received the B.Sc. degree in Automation from the Changchun Institute of Technology, China, in 2019, and the M.Sc. degree in Control Engineering from Northeastern University, China, in 2022. Her main research interests include time-delay systems, multi-agent systems, and collision detection and path planning of robot arms.