Controller Design for Unstable Time-Delay Systems with Unknown Transfer Functions

Hsun-Heng Tsai 1, Chyun-Chau Fuh 2,*, Jeng-Rong Ho 3, Chih-Kuang Lin 3 and Pi-Cheng Tung 3

1 Department of Biomechatronics Engineering, National Pingtung University of Science and Technology, Pingtung 912301, Taiwan; heng@mail.npust.edu.tw
2 Department of Mechanical and Mechatronic Engineering, National Taiwan Ocean University, Keelung 202301, Taiwan
3 Department of Mechanical Engineering, National Central University, Taoyuan 320317, Taiwan; jrho@ncu.edu.tw (J.-R.H.); t330014@cc.ncu.edu.tw (C.-K.L.); t331166@ncu.edu.tw (P.-C.T.)

* Correspondence: f0005@mail.ntou.edu.tw

Abstract: This study developed a method for designing parallel two-degree-of-freedom proportional-integral-derivative controllers for unstable time-delay processes with unknown dynamic equations. First, a performance index accounting for both transient response performance and disturbance rejection was developed. To obtain useful data even if the output of the system exceeds the allowable range, an effective penalty function was included in the performance index. The N–M simplex method was used to iteratively determine the optimal controller parameters. The proposed approach has the following advantages: (1) it can be used regardless of the stability of the open-loop system; (2) the mathematical model and parameters of the process need not be known in advance; (3) it can be used for processes that include measurement noise; (4) it has good transient response performance and is also robust against external disturbances; and (5) it enables more efficient controller design and reduces costs.

Keywords: control; optimization; penalty function; PID; simplex method; time-delay; unstable

1. Introduction

Time delay is a common real-world phenomenon [1–5] because matter and energy exist in a distributed state. A time delay occurs when matter or energy is transported within a medium. Transmission systems, chemical processes, metallurgical processes, hydraulic and pneumatic systems, power systems, biological systems, the environment, and ecosystems are examples of time-delay systems. The existence of time delay complicates the analysis and control of systems. In particular, designing a controller for a time-delay system whose dynamic equation is unknown is challenging.

In most time-delay processes, proportional-integral-derivative (PID) control is used as the feedback controller due to its simple structure and implementation. The design and adjustment of the PID controller are conceptually intuitive. However, designs required to meet multiple objectives at the same time (for example, both good transient response characteristics and disturbance rejection) may be difficult to achieve in practice [6–10].

Many studies have proposed various methods for adjusting PID parameters or parallel two-degree-of-freedom (2-DOF) PID parameters for unstable time-delay systems. For example, Nasution et al. [11] synthesized an optimal H 2 PID controller from an internal model control design for unstable processes with a single right-half-plane pole and a time delay. Hast and Hägglund [12] proposed a method for finding the set-point weights of a PID controller by using convex optimization techniques. Begum et al. [13] developed analytical tuning rules for PID controllers for unstable first order plus dead time processes. Chakraborty et al. [14] proposed an integral–proportional derivative (I-PD) control

Citation: Tsai, H.-H.; Fuh, C.-C.; Ho, J.-R.; Lin, C.-K.; Tung, P.-C. Controller Design for Unstable Time-Delay Systems with Unknown Transfer Functions. Mathematics 2022, 10, 431. https://doi.org/10.3390/math10030431

Academic Editors: Changhua Lien, Hamid Reza Karimi and Sundarapandian Vaidyanathan

Received: 29 December 2021 Accepted: 26 January 2022 Published: 29 January 2022

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).
strategy for integrator plus time delay processes. The proposed scheme comprised an inner proportional derivative (PD) loop and an outer integral loop for controlling a servo and for regulatory actions. Verma and Padhy [15] studied the effects of exponential weights on square-of-error functions for PID control systems. They demonstrated that the performance of any process can be tailored by applying exponential weights in the objective function. Onat [16] presented a graphical method for tuning the proportional integral (PI)-PD controller parameters for unstable time-delay systems. The proposed method used a new concept, namely the centroid of the convex stability region. The method only requires the coordinates of a few special points related to the curve determining the stable controller parameter area. Zhang et al. [17] proposed an optimized robust control algorithm based on the mirror mapping method for a class of industrial unstable processes with time delays. Raja and Ali [18] proposed tuning rules for the PI-PD control structure for a class of unstable time-delay processes. The proposed method required the tuning of only four controller parameters.

However, most of the proposed methods in these studies have failed to provide rules for tuning the controller parameters for unstable time-delay processes with unknown mathematical models. Therefore, a trial-and-error method for designing an actual controller is often adopted if information on the approximate or nominal dynamic equation of the process is lacking.

Moreover, during the adjustment of the controller parameters, the process output may exceed the allowable range of variation. If this occurs, the experiment must be immediately terminated, and relevant safety or protective measures must be initiated to prevent injury to the operator or damage to the equipment. If a running experiment is terminated, not only are raw materials and consumables wasted but the obtained data are incomplete and must also be discarded. Therefore, designing a high-performance controller for unstable real-world processes is a not only time consuming but also a potentially expensive task.

To overcome these difficulties in designing real controllers, this study formulated an optimal controller design method for an unstable time-delay process with measurement noise and external disturbance. First, a novel performance index (objective function) was established based on the response requirements of the controlled process. The performance index includes only an explicit function of time and output error; thus, no prior knowledge of the mathematical model of the process is required to calculate the index value. Moreover, a penalty function was included to ensure that the performance index is applicable even if the output of the process is outside the allowable range. In mathematical optimization, penalty functions can be used to solve constrained optimization problems. By introducing a weighted penalty for violating constraints, the constraint problem can be transformed into a nonconstrained problem.

The Nelder–Mead (N–M) simplex method was used to search for the minimum value of the performance index and obtain optimal controller parameters. The N–M simplex method is an algorithm modified by Nelder and Mead [19] on the basis of the simplex method originally proposed by Spendley, Hext, and Himsworth [20]. Because the N–M simplex method is easy to implement and does not require derivatives of the objective function, it is suitable for solving optimization problems if derivatives of the objective function are not available or contain noise [21–25].

To verify the feasibility and effects of the proposed method, several unstable time-delay processes with unknown transfer functions were investigated as examples. In each simulation, the system output contained measurement noise and could only be operated within a limited range. In addition, an external disturbance was added during the second half of each simulation to assess the robustness of the system. The simulation results reveal that even if the initial controller parameters cannot stabilize the process when the algorithm reaches the iterative termination condition, the obtained controller parameters can not only stabilize the process but also have good transient response performance and robustness against external disturbances.
The remaining parts of the article are organized as follows. The mathematical models of the open-loop system and the three 2-DOF PID controllers considered in this paper are described in Section 2. Section 3 discusses the performance index with the penalty function proposed in this paper and discusses the N–M simplex method for identifying the optimal controller parameters. A comparison and analysis of the simulation results is presented in Section 4. Finally, concluding remarks are given in Section 5.

2. System Description

The open-loop system considered in this paper can be represented by the following transfer function:

\[ P(s) = G(s)e^{-Ts}, \]  

where \( s \) denotes the Laplace operator. In addition, the transfer function \( G(s) \) and the time delay \( T \) are unknown. At least one pole of \( G(s) \) is located in the right half of the complex plane, and \( P(s) \) can be stabilized by using the parallel 2-DOF PID controller as presented in Figure 1, where \( r \) is the reference input, \( y \) is the output, \( e = r - y \) is the error, and \( u \) is the control input.

![Figure 1: Block diagram of the system that is controlled by a parallel 2-DOF PID controller.](image)

The parallel 2-DOF PID controller in Figure 1 can be expressed with the following formula:

\[ u = C_1(s)(r - y) + C_2(s)y \]

\[ \equiv \left( k_{p1} + \frac{k_{i1}}{s} + k_{d1}s \right)(r - y) - \left( k_{p2} + k_{d2}s \right)y \]

\[ = \left( k_{p1} + \frac{k_{i1}}{s} + k_{d1}s \right)e - (k_{p2} + k_{d2}s)y \]

Here, \( k_{p1}, k_{i1}, \) and \( k_{d1} \) denote the coefficients (gains) for the proportional, integral, and derivative terms of \( C_1(s) \), respectively. The definition of the coefficients in \( C_2(s) \) is similar to that for \( C_1(s) \), but the integral term is unnecessary in \( C_2(s) \). We discuss and compare the following three commonly used parallel 2-DOF PID controllers:

- **PID controller**
  
  Let \( k_{p1} = K_p, \ k_{i1} = K_i, \ k_{d1} = K_D, \) and \( k_{p2} = k_{i2} = k_{d2} = 0 \). Then, the parallel 2-DOF PID controller degenerates into the following single-DOF PID controller:

  \[ C_1(s) = \frac{K_p}{s} + \frac{K_i}{s} + K_Ds \]

  \[ C_2(s) = 0 \]

- **I-PD controller**
Define $k_{i1} = K_I$, $k_{p2} = K_P$, $k_{d2} = K_D$, and $k_{p1} = k_{d1} = k_{i2} = 0$. The following I-PD controller is then obtained:

$$C_1(s) = \frac{K_I}{s}$$

(7)

$$C_2(s) = K_P + K_D s$$

(8)

- PI-PD controller

Set $k_{p1} = K_C$, $k_{i1} = K_I$, $k_{p2} = K_P$, $k_{d2} = K_D$, and $k_{d1} = k_{i2} = 0$. The following PI-PD controller is obtained:

$$C_1(s) = K_C + \frac{K_I}{s}$$

(9)

$$C_2(s) = K_P + K_D s$$

(10)

In practical applications, the control system in Figure 1 may be subject to external disturbances $d$ and the output signal is also affected by measurement noise $v$. If a filter is added to the output signal, the system block diagram becomes that in Figure 2, where $C_3(s)$ denotes the filter.

![Figure 2. System block diagram of parallel 2-DOF PID control including the external disturbance $d$, measurement noise $v$, and low-pass filter $C_3(s)$.](image)

In this paper, $C_3(s)$ was considered as a first-order low-pass filter with the following form:

$$C_3(s) = \frac{\omega_C}{s + \omega_C}$$

(11)

Here, $\omega_C$ is the cutoff frequency of the low-pass filter. In practical applications, the approximate transfer function of the open-loop system (Equation (1)) must be obtained using parameter identification techniques. However, if the output signal is disturbed by large-magnitude white noise and the system also uses a low-pass filter, the estimated transfer function would have substantial error; in extreme cases, the result may be meaningless. Thus, many controller design methods based on known open-loop system transfer functions cannot be used. Therefore, the designer can only adjust the controller parameters through trial and error based on experience.

Moreover, the output, $y$, of a real system may indicate temperature, pressure, voltage, current, displacement, or velocity. As with these physical quantities, the actual output value may only be allowed to change within a specific range due to the limitations of the device itself or due to safety considerations. If the output, $y$, exceeds the allowable range of variation, the experiment must be stopped immediately. In this situation, the design of
the controller parameters is more difficult and time consuming. To overcome this problem, the N–M simplex method based on a performance index with a penalty function was used to identify the optimal controller parameters. The method is detailed in the next section.

3. Design of Controller Parameters

The N–M simplex method is one of the most famous algorithms for multidimensional unconstrained optimization without derivatives. The N–M simplex method is intended to solve \(N\)-dimensional unconstrained optimization problems with the following form:

\[
\min_{\mathbf{p} \in \mathbb{R}^n} J(\mathbf{p})
\]

Here, \(\mathbf{p}\) is an \(N \times 1\) vector composed of the controller parameters (controller gains) to be designed, and \(J(\mathbf{p})\) is a scalar performance index (also called objective function or cost function).

To design an optimal controller, the performance index must first be determined. The performance index is used in the simplex method to search for the optimal points. The performance index must account for factors such as the dynamic characteristics of the system and the desired control target. This function is usually a positive-valued function; smaller output values indicate that the control system is closer to its optimization goal. Some commonly used performance indexes are as follows:

1. Integral of absolute error (IAE)

\[
J = \int_0^{T_f} |e(t)|\,dt
\]

2. Integral of squared error (ISE)

\[
J = \int_0^{T_f} e^2(t)\,dt
\]

3. Integral of time-weighted absolute error (ITAE)

\[
J = \int_0^{T_f} t|e(t)|\,dt
\]

4. Integral of time-weighted squared error (ITSE)

\[
J = \int_0^{T_f} te^2(t)\,dt
\]

Here, \(T_f\) is the final time of the experiment or simulation.

These performance indexes are typically only a quantitative reference for comparing the performance of different controllers; they do not directly suggest rules or formulas for determining the parameter values of the controllers. For example, most studies regarding the use of PID, I-PD, or PI-PD to control a system with time delay have focused on systems with a transfer function of a specific form by using sufficient numerical calculations or curve fitting methods to obtain an approximately optimal controller parameter formula. To use these methods, the transfer function of the system must first be known. However, estimating the transfer function of an open-loop system that is unstable and has substantial measurement noise is difficult.

To overcome this problem, the N–M simplex method with a performance index and a penalty function was used to search for the optimal controller parameters. Before applying the proposed performance index in this study, the variables and symbols in the index are first defined in Table 1.
Table 1. Definitions and descriptions of the variable s and symbols in the proposed performance index.

| Name | Definition and Description |
|------|-----------------------------|
| \(r(t)\) | System reference input (i.e., \(r\) in Figure 2). |
| \(y(t)\) | The actual output of the system (i.e., \(y\) in Figure 2). The allowable range of change is \(y_{\text{min}} \leq y \leq y_{\text{max}}\), where \(y_{\text{min}}\) and \(y_{\text{max}}\) are the lower and upper limits of the output allowable change, respectively. If \(y\) exceeds the allowable range, the experiment or simulation is halted immediately. |
| \(e(t)\) | The output error (i.e., \(e\) in Figure 2) is defined as \(e(t) = r(t) - y(t)\). |
| \(d\) | The disturbance to the system (i.e., \(d\) in Figure 2). |
| \(T_f\) | The start time of each simulation is 0. The simulation end time is \(T_f\). |
| \(T_d\) | The disturbance \(d\) is added to the system at time \(T_d\), and the disturbance continues until \(T_f\). |
| \(W_p\) | The weight of the penalty function. \(W_p\) must be sufficient to punish (or cause to avoid) situations in which the output \(y\) exceeds the allowable range. |

The performance index used in this paper is defined as follows:

\[
J = \begin{cases} 
J_A, & \text{if } y \text{ exceeds the allowable range at } t_A, 0 < t_A \leq T_f \\
J_B, & \text{if } y \text{ exceeds the allowable range at } t_B, T_d < t_B < T_f \\
J_C, & \text{otherwise} 
\end{cases}
\]

where

\[
J_A = \int_0^{t_A} te^2(t)\,dt + W_p(T_f - t_A)
\]

\[
J_B = \int_0^{T_d} te^2(t)\,dt + \int_{T_d}^{t_B} (t - T_d)e^2(t)\,dt + W_p(T_f - t_B)
\]

\[
J_C = \int_0^{T_d} te^2(t)\,dt + \int_{T_d}^{T_f} (t - T_d)e^2(t)\,dt
\]

\(J_A\) and \(J_B\) contain penalty functions. The weight \(W_p\) is then determined. First, the most conservative situation is considered in which the error is equal to the maximum value \((y_{\text{max}} - y_{\text{min}})\) from the beginning and maintained for a time \(t_A = t_B = (T_f - \Delta t)\), where \(\Delta t\) is the sampling time. The integral term in Equation (18) is then approximately equal to \(0.5(y_{\text{max}} - y_{\text{min}})^2\Delta t^2\). To ensure that the penalty function is useful, the following conditions must be satisfied:

\[
\frac{1}{2}T_f^2(y_{\text{max}} - y_{\text{min}})^2 < W_p\Delta t
\]

That is,

\[
W_p > \frac{T_f^2(y_{\text{max}} - y_{\text{min}})^2}{2\Delta t}
\]

After an appropriate performance index is defined, the N–M simplex method can be applied to systematically search for optimal controller parameters.

The N–M simplex method generates a simplex sequence; each simplex is composed of \(N + 1\) different vertices \(p_0, \ldots, p_N\), and the corresponding performance index values are \(J_0, \ldots, J_N\), respectively. All vertices \(p_0, \ldots, p_N\) are assumed to satisfy \(J_0 \leq \cdots \leq J_N\) after sorting according to the performance indexes. Next, \(\bar{p}\) and \(\Delta p\) are defined as follows:

\[
\bar{p} = \frac{1}{N} \sum_{i=1}^{N} p_i
\]
In each iteration, the N–M simplex method is used to examine one or more of four different \( \sigma \) values along the line \( \bar{p} + \sigma \Delta \bar{p} \). These four values are denoted as \( \alpha, \pm \beta \), and \( \gamma \) and must satisfy the following condition:

\[
0 < \beta < 1, \quad \gamma > 1, \quad 0 < \alpha < \gamma
\]

In practical applications, the typical values are \( \alpha = 1, \beta = 0.5 \), and \( \gamma = 2 \) \cite{19,22,23}. The values of \( \alpha, \gamma, \beta \), and \( -\beta \) yield the reflection point \( p_r \), the expansion point \( p_e \), the outer contraction point \( p_o \), and the inner contraction point \( p_{oc} \), respectively. The values of the performance index at these four points are denoted as \( J_r, J_e, J_o, \) and \( J_{oc} \), respectively. If none of the four points is superior to the current lowest performance point \( p_0 \), the algorithm shrinks the points \( p_1, \ldots, p_N \) toward the lowest \( p_0 \), thereby producing a new simplex. During the shrinking process, each \( p_i \) is replaced by \( 0.5(p_0 + p_i) \) for \( i = 1, \ldots, n \). Upon the completion of the shrinking process, a new iteration is automatically triggered. This iterative process is continued until the specified termination criteria are satisfied. Table 2 lists the definitions of related names, symbols, and equations in the N–M simplex method.

**Table 2. Definitions of related names, symbols, and equations in the N–M simplex method.**

| Name            | Parameter | Equation       | Performance Index |
|-----------------|-----------|----------------|-------------------|
| Reflection      | \( \sigma = \alpha = 1 \) | \( p_r = \bar{p} + \Delta \bar{p} \) | \( J_r = J(p_r) \) |
| Expansion       | \( \sigma = \gamma = 2 \) | \( p_e = \bar{p} + 2\Delta \bar{p} \) | \( J_e = J(p_e) \) |
| Outer contraction| \( \sigma = \beta = 0.5 \) | \( p_o = \bar{p} + 0.5\Delta \bar{p} \) | \( J_o = J(p_o) \) |
| Inner contraction| \( \sigma = -\beta = -0.5 \) | \( p_{oc} = \bar{p} - 0.5\Delta \bar{p} \) | \( J_{oc} = J(p_{oc}) \) |
| Shrink          |           | \( p_i = 0.5(p_0 + p_i) \) | \( J_i = J(p_i) \) |

As a simplified example of the N–M simplex method, consider a two-dimensional space \( (N = 2; \bar{p}_i \in \mathbb{R}^2, i = 0,1,2) \). First, three points that are not on the same straight line are randomly generated or preliminarily selected, and a monomer with \( p_0, p_1, \) and \( p_2 \) is formed (Figure 3). The performance index corresponding with the vertex \( p_i \) is represented by \( J_i \). After all the vertices have been sorted according to their performance index size, \( J_0 \leq J_1 \leq J_2 \) is satisfied.

In each iteration of the N–M simplex method, an attempt is made to find a superior point through a simplified linear search or shrink method to replace the worst point in the simplex (i.e., \( p_2 \)). In the linear search part, some or all of the following four points are considered in each iteration: \( p_r, p_e, p_o, p_{oc} \). This iterative step is repeated to approximate the best solution. The basic steps of the N–M simplex method are summarized in Table 3.
Figure 3. Diagram of various possible candidate points during each of N–M simplex iteration: (a) linear search; (b) shrink.

Table 3. Steps for search for the optimal controller parameters, \( \mathbf{p} \), using the N–M simplex method.

| Step | Description |
|------|-------------|
| Step 1 | Choose an initial \( \mathbf{p}_0, \ldots, \mathbf{p}_N \) and execute Subroutine to obtain \( J_0, \ldots, J_N \). |
| Step 2 | Sort \( \mathbf{p}_0, \ldots, \mathbf{p}_N \) such that \( J_0 \leq \cdots \leq J_N \). |
| Step 3 | If the termination criteria are satisfied, go to Step 20. |
| Step 4 | Let \( \mathbf{p} = \frac{1}{N} \sum_{i=0}^{N-1} \mathbf{p}_i \) and \( \Delta \mathbf{p} = \mathbf{p} - \mathbf{p}_N \). |
| Step 5 | (Reflection) Let \( \mathbf{p}_e = \mathbf{p} + \Delta \mathbf{p} \) and execute Subroutine to obtain \( J_r \). |
| Step 6 | If \( J_r < J_0 \), go to Step 7; else go to Step 11. |
| Step 7 | (Expansion) Let \( \mathbf{p}_e = \mathbf{p} + 2 \Delta \mathbf{p} \) and execute Subroutine to obtain \( J_e \). |
| Step 8 | If \( J_e < J_r \), go to Step 9; else go to Step 10. |
| Step 9 | Let \( \mathbf{p}_N = \mathbf{p}_e \) and \( J_N = J_e \), then go to Step 2. |
| Step 10 | Let \( \mathbf{p}_N = \mathbf{p}_r \) and \( J_N = J_r \), then go to Step 2. |
| Step 11 | If \( J_r < J_N \), go to Step 12; else go to Step 16. |
| Step 12 | (Outer contraction) Let \( \mathbf{p}_e = \mathbf{p} + 0.5 \Delta \mathbf{p} \) and execute Subroutine to obtain \( J_c \). |
| Step 13 | If \( J_c < J_r \), go to Step 14; else, go to Step 15. |
| Step 14 | Let \( \mathbf{p}_N = \mathbf{p}_e \) and \( J_N = J_c \), then go to Step 2. |
| Step 15 | Let \( \mathbf{p}_N = \mathbf{p}_r \) and \( J_N = J_r \), then go to Step 2. |
| Step 16 | (Inner contraction) Let \( \mathbf{p}_{cc} = \mathbf{p} - 0.5 \Delta \mathbf{p} \) and execute Subroutine to obtain \( J_{cc} \). |
| Step 17 | If \( J_{cc} < J_N \) go to Step 18; else, go to Step 19. |
| Step 18 | Let \( \mathbf{p}_N = \mathbf{p}_{cc} \) and \( J_N = J_{cc} \), then go to Step 2. |
| Step 19 | (Shrink) Let \( \mathbf{p}_i = 0.5(\mathbf{p}_0 + \mathbf{p}_i) \) and execute Subroutine to obtain \( J_i \) where \( i = 1, \ldots, N \), then go to Step 2. |
| Step 20 | Print out \( \mathbf{p}_0 \) and \( J_0 \). |

Subroutine

Simulate or experiment using the given parameter \( \mathbf{p} \) to obtain the corresponding index \( J \).

4. Simulation Results

To examine the feasibility and effectiveness of the proposed method, simulations were run for the following cases. Each open-loop system was an unstable process with a time delay. The reference input for all cases was set as \( r = 1 \), and the allowable range of variation in the output was \(-1 \leq y \leq 3\). Moreover, \( d \) denotes a constant disturbance to the system. The start time of each simulation was defined as 0. The simulation completion (end) time is denoted as \( T_f \). The disturbance \( d \) was added to the system at time \( T_d = 0.5 T_f \), and the disturbance continued until \( T_f \). All systems used a first-order low-pass
filter, such as that expressed in Equation (11), and the cutoff frequency \( \omega_c \) depended on the system and noise characteristics.

4.1. Case 1

Consider an unstable time-delay process with the following transfer function:

\[
P(s) = \frac{1}{s-1} e^{-0.2s}
\]  

(26)

In this case, the sampling time was 0.001, and the preset end time \( T_f \) was 10. The target value of the output, \( y \), was \( r = 1 \), and the lower limit and upper limits for \( y \) were \( y_{\text{min}} = -1 \) and \( y_{\text{max}} = 3 \), respectively. The measurement noise, \( v \), was normally distributed with mean 0 and standard deviation 0.1. The cutoff frequency for the first-order low-pass filter was \( \omega_c = 100 \). An external disturbance \( d = 1.5 \) was added to the system at time \( T_d = 5 \). The termination condition of the N–M simplex algorithm was 50 iterations. The performance index is defined in Equations (17)–(20), where the coefficient of the penalty function was set to \( W_p = 10^6 \) according to Inequality (22).

4.1.1. PID Controller

Because the PID controller in this case had three parameters, four sets of controller parameters (gains) were first arbitrarily designated as the four vertices of the initial simplex. Table 4 lists the four sets of initial controller parameters and their corresponding performance indexes. Figure 4 lists the time responses after these four sets of initial parameters were used to control the system.

| \( i \) | \( \mathbf{p}_i = [K_P, K_I, K_D] \) | \( J_i = J(\mathbf{p}_i) \) |
|------|-----------------|-------------|
| 0    | \( \mathbf{p}_0 = [6, 12, 0.8] \) | \( J_0 = 7.8990 \times 10^6 \) |
| 1    | \( \mathbf{p}_1 = [2, 8, 1.0] \) | \( J_1 = 8.6430 \times 10^6 \) |
| 2    | \( \mathbf{p}_2 = [4, 10, 1.2] \) | \( J_2 = 8.6670 \times 10^6 \) |
| 3    | \( \mathbf{p}_3 = [8, 6, 0.2] \) | \( J_3 = 9.3660 \times 10^6 \) |

Figure 4. Time responses after these four sets of initial PID controller parameters were used to control the system.
As indicated by the time responses in Figure 4, all four sets of initial PID controller parameters listed in Table 4 exceeded the allowable range before $T_f$. Outputs that exceed the allowable safety range earlier have a larger corresponding performance index (i.e., worse controller parameters). Therefore, the proposed penalty function achieved the expected effect.

Subsequently, the aforementioned four sets of parameters were used as the four vertices of the initial simplex. The maximum number of iterations was set as 50. The result for the obtained controller is presented in Figure 5. The parameters of the controller were as follows:

$$ p = [K_p \ K_i \ K_o] = [5.0420 \ 5.0580 \ 0.30729] $$

The corresponding performance index was $J = 3.2556 \times 10^{-1}$, far smaller than the performance indexes of the original four sets of controllers (Table 4).

Thus, even if the initial controller parameters result in system instability, our proposed performance index can be used with the N–M algorithm to effectively obtain a set of stable, superior controller parameters.

![Figure 5](image)

**Figure 5.** Time response of System (26) controlled by the PID controller. Gains were found using the N–M simplex method based on the initial vertices given in Table 4.

### 4.1.2. I-PD Controller

The I-PD controller used in this case also had three parameters; thus, four sets of controller parameters were arbitrarily designated as the four vertices of the initial simplex. The four sets of initial controller parameters and their corresponding performance indexes are listed in Table 5. Figure 6 presents the time responses after these four sets of initial parameters were used to control the system.

| $i$ | $p_i = [K_i \ K_p \ K_o]$ | $J_i = J(p_i)$ |
|-----|--------------------------|----------------|
| 0   | $p_0 = [7 \ 6 \ 0.5]$    | $J_0 = 5.6286 \times 10^{-1}$ |
| 1   | $p_1 = [3 \ 7 \ 0.5]$    | $J_1 = 1.6343 \times 10^{0}$ |
| 2   | $p_2 = [5 \ 2 \ 0.2]$    | $J_2 = 5.3141 \times 10^{1}$ |
| 3   | $p_3 = [15 \ 6 \ 0.2]$   | $J_3 = 6.0510 \times 10^{6}$ |
As indicated in Figure 6, the parameters $p_0$, $p_1$, and $p_2$ could maintain the system output within the allowable range until $T_f$; therefore, the penalty function values were 0 and the corresponding index values were small. However, $p_3$ caused the system output to exceed the allowable range before $T_f$. Therefore, the penalty function achieved the expected effect in that the corresponding index value was much larger than those for $p_0$, $p_1$, and $p_2$.

The aforementioned four sets of parameters were used as the four vertices of the initial simplex. The parameters obtained using the N–M simplex optimal search method are as follows:

$$p = [K_I, K_P, K_D] = [9.7159, 6.2491, 0.41445]$$

(28)

The time response using the obtained controller parameters is presented in Figure 7. The corresponding performance index was $J = 5.2350 \times 10^{-5}$; this index was lower than those of the original four sets of controllers.

Figure 6. Time response with these four sets of initial I-PD controller parameters.

Figure 7. Time response of System (26) controlled by the I-PD controller with gains obtained using the N–M simplex method based on the initial vertices given in Table 5.
4.1.3. PI-PD Controller

Because the PI-PD controller used in this case had four gains, five sets of controller gains were first arbitrarily designated as the five vertices of the initial simplex. The five sets of gains and their corresponding performance indexes are presented in Table 6. The time responses of the system are presented in Figure 8.

Table 6. Initial PI-PD controller gains and their corresponding performance indexes for the system (26).

| \(i\) | \(p_i = [K_c \ K_I \ K_p \ K_d]\) | \(J_i = f(p_i)\) |
|------|--------------------------------|----------------|
| 0    | \(p_0 = [3 \ 2 \ 0.5 \ 0.5]\)  | \(J_0 = 6.2384 \times 10^{-1}\) |
| 1    | \(p_1 = [3 \ 10 \ 0.8 \ 0.3]\) | \(J_1 = 1.9547 \times 10^0\) |
| 2    | \(p_2 = [5 \ 10 \ 0.4 \ 1.0]\) | \(J_2 = 1.2092 \times 10^1\) |
| 3    | \(p_3 = [5 \ 8 \ 1.0 \ 1.0]\)  | \(J_3 = 5.1920 \times 10^6\) |
| 4    | \(p_4 = [3 \ 5 \ 2.5 \ 2.5]\)  | \(J_4 = 9.1870 \times 10^6\) |

Figure 8. Time responses after these five sets of initial PI-PD controller gains were used to control System (26).

Among these, the parameters \(p_0\), \(p_1\), and \(p_2\) could maintain the system output within the allowable range until \(T_f\); therefore, all their penalty function values were 0 and the corresponding index values were small. Both \(p_3\) and \(p_4\) caused the system output to exceed the allowable range before \(T_f\). Therefore, the penalty function achieved the expected effect; the index values were larger than those for the other parameter sets. This result also indicated that parameter sets that caused earlier simulation interruption have larger corresponding index values (indicating worse controller parameters).

Similarly, the aforementioned five sets of controller parameters were applied as the vertices of the initial simplex (Figure 9). The resulting controller parameters obtained by using the N–M simplex optimal search method were as follows:

\[
p \triangleq [K_c \ K_I \ K_p \ K_d] = [2.9726 \ 6.5343 \ 2.6429 \ 0.46122]
\]

The corresponding performance index was \(J = 1.1277 \times 10^{-1}\); this value was lower than those of the original five sets of controllers.
Figure 9. Time response of System (26) controlled by the PI-PD controller. The controller parameters were obtained using the N–M simplex method with the initial vertices given in Table 6.

4.2. Case 2

We consider a second-order unstable time-delay process with the following transfer function:

\[ P(s) = \frac{1}{(2s - 1)(0.5s + 1)} e^{-0.5s} = \frac{1}{s^2 + 1.5s - 1} e^{-0.5s} \] (30)

In this case, the sampling time was 0.005 and the preset end time, \( T_f \), was 50. The target value of the output, \( y \), was \( r = 1 \), and \( y_{\min} = -1 \) and \( y_{\max} = 3 \). The measurement noise, \( v \), was normally distributed with a mean of 0 and standard deviation of 0.1. The cutoff frequency of the first-order low-pass filter was \( \omega_c = 20 \). The external disturbance \( d = -1 \) was injected into the system at \( T_d = 25 \). The termination condition of the N–M simplex algorithm was 30 iterations. The performance index was defined as in Equations (17)–(20), and the coefficient of the penalty function was set to \( W_p = 10^7 \) according to Inequality (22).

4.2.1. PID Controller

We arbitrarily selected four sets of parameters as the four vertices for the initial simplex for System (30). Table 7 presents four sets of initial controller parameters and their corresponding performance indexes. The time responses obtained using these four sets of controllers are presented in Figure 10.

Table 7. Four sets of initial PID controller parameters used by System (30) and their corresponding performance indexes.

| \( i \) | \( \mathbf{p}_i \) | \( J_i = J(\mathbf{p}_i) \) |
|-----|-------|-----------------|
| 0   | \( \mathbf{p}_0 = [2.0 \quad 0.1 \quad 2] \) | \( J_0 = 1.0225 \times 10^2 \) |
| 1   | \( \mathbf{p}_1 = [1.5 \quad 0.2 \quad 4] \) | \( J_1 = 1.9751 \times 10^2 \) |
| 2   | \( \mathbf{p}_2 = [1.0 \quad 0.2 \quad 3] \) | \( J_2 = 2.5865 \times 10^8 \) |
| 3   | \( \mathbf{p}_3 = [2.5 \quad 0.1 \quad 5] \) | \( J_3 = 4.2925 \times 10^8 \) |
Figure 10. Output responses of System (30) controlled by the four sets of PID parameters listed in Table 7.

For System (30), the optimal PID controller parameters obtained by the N–M simplex method were as follows:

\[
p = [K_p, K_i, K_d] = [3.3231, 0.53297, 1.6756]
\]  

(31)

The output response using these controller parameters is presented in Figure 11, and its corresponding performance index value was \( J = 7.9370 \).

Figure 11. Time response of System (30) controlled by the PID controller with gains obtained using the N–M simplex method and based on the initial vertices given in Table 7.
4.2.2. I-PD Controller

Similarly, four sets of parameters as the four vertices of the initial simplex were arbitrarily selected to design the I-PD controller for System (30). The four sets of initial controller parameters and their corresponding performance indexes are presented in Table 8. The time responses obtained using these four sets of controllers are presented in Figure 12.

Table 8. Four sets of initial I-PD controller parameters and their corresponding performance indexes for System (30).

| i  | \( \mathbf{p}_i = [K_i \ K_P \ K_D] \) | \( J_i = J(\mathbf{p}_i) \)      |
|----|---------------------------------|--------------------------------|
| 0  | \( [0.5 \ 3 \ 3] \)             | \( J_0 = 3.2987 \times 10^1 \) |
| 1  | \( [2.0 \ 5 \ 2] \)             | \( J_1 = 2.8433 \times 10^3 \) |
| 2  | \( [1.0 \ 1 \ 4] \)             | \( J_2 = 2.9420 \times 10^8 \) |
| 3  | \( [1.5 \ 7 \ 1] \)             | \( J_3 = 4.2865 \times 10^8 \) |

Figure 12 reveals that both \( \mathbf{p}_3 \) and \( \mathbf{p}_2 \) caused range violations before \( T_f \). The corresponding performance indexes were thus large. Although the controller parameters \( \mathbf{p}_1 \) and \( \mathbf{p}_0 \) did not cause range violations, the output corresponding to parameter \( \mathbf{p}_1 \) gradually diverged, and the output corresponding to the parameter \( \mathbf{p}_0 \) was convergent. Therefore, the performance index of \( \mathbf{p}_0 \) was smaller than that of \( \mathbf{p}_1 \).

The aforementioned four sets of parameters were used as the four vertices of the initial simplex. The parameters obtained using the N–M simplex optimal search method were as follows:

\[
\mathbf{p} = [K_i \ K_P \ K_D] = [1.9890 \ 4.1851 \ 2.4074] \tag{32}
\]

The time response of System (30) with the parameters given in (32) is presented in Figure 13. The corresponding performance index was \( J = 8.1780 \), which is significantly lower than those of the initial four sets of controllers.
Figure 13. Time response of System (30) controlled by the I-PD controller with gains obtained by
the N–M simplex method and based on the initial vertices given in Table 8.

4.2.3. PI-PD Controller

Table 9 lists five sets of initial parameters and corresponding performance indexes
for the PI-PD controller for System (30). The time response of the system is presented in
Figure 14.

Table 9. Initial PI-PD controller gains and their corresponding performance indexes for System (30).

| i  | $p_i = [K_C \ K_I \ K_P \ K_D]$ | $J_i = f(p_i)$   |
|----|---------------------------------|------------------|
| 0  | $p_0 = [0.1 \ 1.0 \ 3 \ 1.0]$   | $J_0 = 1.1415 \times 10^1$ |
| 1  | $p_1 = [1.0 \ 2.0 \ 4 \ 2.0]$   | $J_1 = 4.6450 \times 10^7$ |
| 2  | $p_2 = [0.2 \ 2.0 \ 2 \ 2.0]$   | $J_2 = 2.2615 \times 10^8$ |
| 3  | $p_3 = [0.5 \ 0.5 \ 5 \ 4.0]$   | $J_3 = 3.8950 \times 10^8$ |
| 4  | $p_4 = [2.0 \ 3.0 \ 1 \ 0.5]$   | $J_4 = 4.7660 \times 10^8$ |

Figure 14. Time responses using these five sets of initial PI-PD controller gains to control System
(30).
The aforementioned five sets of controller parameters were used as the vertices of the initial simplex. The resulting controller gains obtained using the N–M simplex optimal search method were as follows:

\[ \mathbf{p} = [K_C \ K_I \ K_p \ K_d] = [0.64483 \ 1.6878 \ 3.4654 \ 2.4357] \]  

(33)

The time response of System (30) controlled by the above PI-PD controller is presented in Figure 15. The corresponding performance index was \( J = 2.1582 \), substantially lower than those of the initial four sets of controllers in Table 9.

### Figure 15

Time response of System (30) controlled by the PI-PD controller. Controller parameters were obtained with the N–M simplex method using the initial vertices given in Table 9.

4.3. Case 3

Consider another second-order unstable time-delay process with the following transfer function:

\[ P(s) = \frac{-0.4s + 1}{(3s - 1)(0.5s + 1)} e^{-0.5s}, \quad \text{PI-PD} \]  

(34)

In this case, the sampling time was 0.005, and the preset control end time, \( T_f \), was 50. The target value of the output, \( y \), was \( r = 1 \), and \( y_{\min} = -1 \) and \( y_{\max} = 3 \). The measurement noise, \( v \), was normally distributed and had a mean of 0 and standard deviation of 0.1. The cutoff frequency of the first-order low-pass filter was \( \omega_c = 20 \). The external disturbance \( d = -1 \) was injected into the system at time \( T_d = 25 \). The termination condition of the N–M simplex algorithm was 30 iterations. The performance index is defined as in Equations (17)–(20), and the coefficient of the penalty function was set to \( W_p = 10^7 \) according to Inequality (22).

4.3.1. PID Controller

Four sets of initial PID controller gains and their corresponding performance indexes are listed in Table 10. The time responses obtained by using these four sets of controllers are displayed in Figure 16.

### Table 10

Four sets of initial PID controller parameters used by System (34) and their corresponding performance indexes.
In this case, the optimal PID controller parameters searched by the N–M simplex method were as follows:

\[
\mathbf{p} = [K_p, K_i, K_d] = [2.2624, 0.27294, 0.47510] \quad (35)
\]

The time response of the system using this controller parameter is presented in Figure 17, and its corresponding performance index value was \( J = 38.623 \).

\[
P(s) = \frac{-0.4s + 1}{(3s - 1)(0.2s + 1)}e^{-0.5s}, \quad \text{PID}
\]

Figure 16. Output responses of System (34) controlled by the initial PID gains listed in Table 10.

Figure 17. Time response of System (34) controlled by the PID controller with gains obtained using the N–M simplex method and based on the initial vertices given in Table 10.
4.3.2. I-PD Controller

The four sets of initial controller parameters and their corresponding performance indexes for the I-PD controller for System (34) are listed in Table 11. The time responses obtained by using these controllers are displayed in Figure 18.

**Table 11.** Four sets of initial I-PD controller parameters and their corresponding performance indexes for System (34).

| $i$ | $p_i = [K_I \ K_P \ K_D]$ | $J_i = J(p_i)$ |
|-----|----------------------------|-----------------|
| 0   | $p_0 = [0.2 \ 3 \ 0.3]$    | $J_0 = 3.8643 \times 10^1$ |
| 1   | $p_1 = [0.5 \ 2 \ 0.2]$    | $J_1 = 1.4464 \times 10^2$ |
| 2   | $p_2 = [0.4 \ 4 \ 0.1]$    | $J_2 = 3.3170 \times 10^8$ |
| 3   | $p_3 = [0.3 \ 1 \ 0.4]$    | $J_3 = 3.9800 \times 10^8$ |

![Figure 18](image.png)

Figure 18. Time responses of System (34) controlled by the four sets of initial I-PD controller parameters listed in Table 11.

The aforementioned four sets of parameters were then used as the four vertices of the initial simplex. The controller parameters obtained using the N–M simplex optimal search method were as follows:

$$p \triangleq [K_I \ K_P \ K_D] = [0.34863 \ 2.5022 \ 0.25064] \quad (36)$$

The time response of the system controlled by the I-PD with gain as in (32) is displayed in Figure 19. The corresponding performance index was $J = 15.254$, which was lower than those of the initial four sets of controllers.
Figure 19. Time response of System (34) controlled by the I-PD controller with gains obtained using the N–M simplex method based on the initial vertices given in Table 11.

4.3.3. PI-PD Controller

Similarly, five sets of initial controller parameters and their corresponding performance indexes for the PI-PD controller for System (34) are listed in Table 12. The time response of the system is displayed in Figure 20.

Table 12. Initial PI-PD controller gains and corresponding performance indexes for System (34).

| i | $\mathbf{p}_i = [K_C, K_I, K_P, K_D]$ | $J_i = J(\mathbf{p}_i)$ |
|---|---------------------------------|-------------------|
| 0 | $\mathbf{p}_0 = [0.1, 1.0, 3.0, 1]$ | $J_0 = 1.4203 \times 10^1$ |
| 1 | $\mathbf{p}_1 = [0.4, 1.0, 1.5, 2]$ | $J_1 = 1.5060 \times 10^8$ |
| 2 | $\mathbf{p}_2 = [0.2, 2.0, 2.5, 2]$ | $J_2 = 1.5095 \times 10^8$ |
| 3 | $\mathbf{p}_3 = [0.5, 0.5, 1.0, 1]$ | $J_3 = 2.1820 \times 10^8$ |
| 4 | $\mathbf{p}_4 = [0.3, 1.5, 2.0, 3]$ | $J_4 = 4.5125 \times 10^8$ |

Figure 20. Time responses using these five sets of initial PI-PD controller gains to control System (34).
The aforementioned five sets of controller parameters were used as the vertices of the initial simplex. The resulting gains obtained using the N–M simplex optimal search method are as follows:

\[ p \triangleq [K_C, K_I, K_P, K_D] = [0.063756, 0.96157, 3.1812, 1.4933] \]  

Figure 21 is the time response of System (34) controlled by the above PI-PD controller. The corresponding performance index is \( J = 5.5301 \); this is also substantially lower than those of the initial controller parameters presented in Table 12.

\[ P(s) = \frac{-0.4s + 1}{(3s - 1)(0.2s + 1)} e^{-0.5s}, \quad \text{PI-PD} \]

\[ J = 5.5301e + 00 \]

Figure 21. Time response of System (34) controlled by the PI-PD controller with controller parameters obtained using the N–M simplex method with initial vertices as in Table 12.

4.4. Summary of Cases 1 to 3

The simulation results reveal that, for an unstable time-delay system with an unknown transfer function, the performance index proposed in this paper combined with the N–M simplex algorithm can be effectively used to design stable controller parameters with superior performance. The performance indexes corresponding to the various successful controllers used in Cases 1 to 3 are listed Table 13.

Table 13. Performance indexes corresponding to the obtained controllers from Cases 1–3.

| System                              | Controller | Performance |
|-------------------------------------|------------|-------------|
| Case 1: \( P(s) = \frac{1}{s-1} e^{-0.2s} \) | PID        | \( J = 0.32556 \) |
|                                     | I-PD       | \( J = 0.52350 \) |
|                                     | PI-PD      | \( J = 0.11277 \) |
| Case 2: \( P(s) = \frac{1}{(2s-1)(0.5s+1)} e^{-0.5s} \) | PID        | \( J = 7.9370 \) |
|                                     | I-PD       | \( J = 8.1780 \) |
|                                     | PI-PD      | \( J = 2.1582 \) |
| Case 3: \( P(s) = \frac{-0.4s + 1}{(3s - 1)(0.2s + 1)} e^{-0.5s} \) | PID        | \( J = 38.623 \) |
|                                     | I-PD       | \( J = 15.254 \) |
|                                     | PI-PD      | \( J = 5.5301 \) |

\(^1\) Relative minimum of the performance indexes obtained for a system with three different controllers.
Table 13 reveals that, with the same performance index, the performance of the PI-PD controller is superior to that of the other two controllers because the PI-PD controller has one more adjustable parameter than the other two controllers do. To identify a controller with better performance, a 2-DOF PID controller with a complete set of five independently adjustable parameters could be used as in Equation (4). The method proposed in this paper could then be used to obtain optimal controller parameters.

5. Conclusions

In this paper, a method for designing parallel 2-DOF PID controllers for unstable time-delay processes with unknown transfer functions was proposed. First, a performance index including both the target value response performance and disturbance rejection was developed. Moreover, to gain value from the incomplete data obtained if the output of the system exceeds the allowable range, an effective penalty function was included in the performance index. The N–M simplex method was then used to iteratively search for optimal controller parameters. The N–M simplex method is advantageous because the algorithm is easy to implement and the system dynamic equations do not need to be known in advance.

To verify the feasibility and effects of the proposed method, several unstable time-delay processes with unknown transfer functions were used as example cases. In these simulations, the output signal of each system contained measurement noise, and the control input was also subject to an external disturbance. Moreover, to increase the similarity to real-world processes, a limit was set on the range of the system output. If the output exceeded the allowable range, the simulation or experiment could be terminated immediately. Although the data obtained from terminated simulations were incomplete, the penalty function enabled the calculation of a meaningful performance index value. Thus, the N–M simplex algorithm could identify the optimal parameters smoothly and recursively. This feature can not only reduce waste in practical applications but also enable the efficient design of controller parameters with superior performance. The simulation results reveal that even if the initial controller parameters result in system instability, the closed-loop system with the final parameters is not only stable but also has good transient response performance.

This paper presents an optimal controller parameter search method for unstable time-delay systems with an unknown transfer function. It has the following advantages: (1) it can be used for unstable processes in an open-loop system; (2) the dynamic equations of the systems can be unknown; (3) it can be used in systems that include measurement noise; (4) it can balance transient response performance and robustness against external disturbances; and (5) its systematic nature makes design more efficient.

Future designers encountering constraints in a controlled system or attempting to find a global optimal controller could modify or add an integral term or penalty function to the performance index in accordance with the concepts proposed in this paper. Design goals could be achieved in combination with global optimization methods.

Author Contributions: Conceptualization, H.-H.T. and C.-C.F.; software, C.-C.F.; validation, H.-H.T. and C.-C.F.; resources, P.-C.T.; writing—original draft preparation, H.-H.T. and C.-C.F.; writing—review and editing, H.-H.T., C.-C.F., J.-R.H., C.-K.L. and P.-C.T.; project administration, C.-C.F. and P.-C.T.; funding acquisition, P.-C.T. and H.-H.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Ministry of Science and Technology, Taiwan, ROC, under Grant MOST 110-2218-E-008-

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.
References

1. Fridman, E. Introduction to Time-Delay Systems: Analysis and Control; Springer: Berlin/Heidelberg, Germany, 2014.
2. Birs, I.; Muresan, C.; Nascu, L.; Ionescu, C. A Survey of Recent Advances in Fractional Order Control for Time Delay Systems. IEEE Access 2019, 7, 30951–30965, https://doi.org/10.1109/ACCESS.2019.2902567.
3. Liu, K.; Selivanov, A.; Fridman, E. Survey on Time-Delay Approach to Networked Control. Annu. Rev. Control. 2019, 48, 57–79, https://doi.org/10.1016/j.arcontrol.2019.06.005.
4. Sun, J.; Su, G.; Chen, X. Data-Driven Controller Synthesis for Parameters Unknown Linear-Delay Systems with Input Constraint. In Proceedings of the 33rd Chinese Control and Decision Conference (CCDC), Kunming, China, 22–24 May 2021; IEEE: Piscataway, NJ, USA; pp. 5915–5920, https://doi.org/10.1109/CCDC52312.2021.9601515.
5. Briones, O.A.; Rojas, A.J.; Sbarbaro, D. Generalized Predictive PI Controller: Analysis and Design for Time Delay Systems. In Proceedings of the 2021 American Control Conference (ACC), New Orleans, LA, USA, 25–28 May 2021; IEEE: Piscataway, NJ, USA; pp. 2509–2514, https://doi.org/10.23919/ACC50511.2021.9482789.
6. Ravikishore, C.; Praveen Kumar, D.T.V.; Padma Sree, R. Enhanced Performance of PID Controllers for Unstable Time Delay Systems Using Direct Synthesis Method. Indian Chem. Eng. 2021, 63, 293–309, https://doi.org/10.1080/00194506.2020.1736650.
7. Balagué, V.; González, A.; Garcia, P.; Blanes, F. Enhanced 2-DOF PID Controller Tuning Based on an Uncertainty and Disturbance Estimator with Experimental Validation. IEEE Access 2021, 9, 99092–99102, https://doi.org/10.1109/ACCESS.2021.3099057.
8. Feng, Z.Y.; Guo, H.; She, J.; Xu, L. Weighted Sensitivity Design of Multivariable PID Controllers via a New Iterative LMI Approach. J. Process. Control. 2022, 110, 24–34, https://doi.org/10.1016/j.jprocont.2021.11.016.
9. Tognetti, E.S.; de Oliveira, G.A. Robust State Feedback-Based Design of PID Controllers for High-Order Systems with Time-Delay and Parametric Uncertainties. J. Control. Autom. Elec. Syst. 2022, 33, 1–11, https://doi.org/10.1007/s40313-021-00846-2.
10. Dubey, V.; Goud, H.; Sharma, P.C. Role of PID Control Techniques in Process Control System: A Review. In Data Engineering for Smart Systems; Springer: Berlin/Heidelberg, Germany, 2022; pp. 659–670.
11. Nasution, A.A.; Jeng, J.C.; Huang, H.P. Optimal H∞ IMC-PID Controller with Set-Point Weighting for Time-Delayed Unstable Processes. Ind. Eng. Chem. Res. 2011, 50, 4567–4578, https://doi.org/10.1021/ie1021688.
12. Hast, M.; Hägglund, T. Optimal Proportional–Integral–Derivative Set-Point Weighting and Tuning Rules for Proportional Set-Point Weights. IET Control. Theory Appl. 2015, 9, 2266–2272, https://doi.org/10.1049/iet-cta.2015.0171.
13. Begum, K.G.; Rao, A.S.; Radhakrishnan, T.K. Maximum Sensitivity Based Analytical Tuning Rules for PID Controllers for Unstable Dead Time Processes. Chem. Eng. Res. Des. 2016, 109, 593–606, https://doi.org/10.1016/j.cherd.2016.03.003.
14. Chakraborty, S.; Ghosh, S.; Naskar, A.K. I-PD Controller for Integrating plus Time-Delay Processes. IET Control. Theory Appl. 2017, 11, 3137–3145, https://doi.org/10.1049/iet-cta.2017.0112.
15. Verma, B.; Padhy, P.K. Optimal PID Controller Design with Adjustable Maximum Sensitivity. IET Control. Theory Appl. 2018, 12, 1156–1165, https://doi.org/10.1049/iet-cta.2017.1078.
16. Onat, C. A New Design Method for PI-PD Control of Unstable Processes with Dead Time. ISA Trans. 2019, 84, 69–81, https://doi.org/10.1016/j.isatra.2018.08.029.
17. Zhang, G.; Tian, B.; Zhang, W.; Zhang, X. Optimized Robust Control for Industrial Unstable Process via the Mirror-Mapping Method. ISA Trans. 2019, 86, 9–17, https://doi.org/10.1016/j.isatra.2018.10.040.
18. Raja, G.L.; Ali, A. New PI-PD Controller Design Strategy for Industrial Unstable and Integrating Processes with Dead Time and Inverse Response. J. Control. Autom. Electr. Syst. 2021, 32, 266–280, https://doi.org/10.1007/s40313-020-00679-5.
19. Nelder, J.A.; Mead, R. A Simplex Method for Function Minimization. Comput. J. 1965, 7, 308–313, https://doi.org/10.1093/comjnl/7.4.308.
20. Spendly, W.; Hext, G.R.; Himsworth, F.R. Sequential Application of Simplex Designs in Optimization and Evolutionary Operation. Technometrics 1962, 4, 441–461, https://dx.doi.org/10.1080/00401706.1962.10490033.
21. Lagarias, J.C.; Reeds, J.A.; Wright, M.H.; Wright, P.E. Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions. SIAM J. Optim. 1998, 9, 112–147, https://doi.org/10.1137/S1052623496303470.
22. McKinnon, K.I.M. Convergence of the Nelder-Mead Simplex Method to a Nonstationary Point. SIAM J. Optim. 1998, 9, 148–158, https://doi.org/10.1137/S1052623496303482.
23. Price, C.J.; Coope, I.D.; Byatt, D. A Convergent Variant of the Nelder-Mead Algorithm. J. Optim. Theory Appl. 2002, 113, 5–19, https://doi.org/10.1023/A:1014849028575.
24. Fuh, C.C.; Tsai, H.H. Applying the Simplex Method to Design the Optimal Parameters of a Sliding Mode Controller for a System with Input Saturation and Unknown Disturbance Bound. IEEE Access 2021, 9, 149100–149109, https://doi.org/10.1109/ACCESS.2021.3117030.
25. Iziçi, D.; Hekimoğlu, B.; Ekinchi, S. A New Artificial Ecosystem-Based Optimization Integrated with Nelder-Mead Method for PID Controller Design of Buck Converter. Alex. Eng. J. 2022, 61, 2030–2044, https://doi.org/10.1016/j.aej.2021.07.037.