Results in Control and Optimization 12 (2023) 100289

Contents lists available at ScienceDirect

Results in Control and Optimization

journal homepage: www.elsevier.com/locate/rico

A Particle Swarm Optimization tuned nonlinear PID controller with improved performance and robustness for First Order Plus Time Delay systems

Stefanos Charkoutsis *, Mohamed Kara-Mohamed

Faculty of Engineering and Technology, Liverpool John Moores University, 3 Byrom St, Liverpool, L3 3AF, England, United Kingdom

ARTICLE INFO

Keywords:
Nonlinear
PID
NLPID
Delay
FOPTD
PSO
MATLAB/Simulink

ABSTRACT

The Proportional, Integral, and Derivative (PID) controller is a ubiquitous controller within industry. The conventional PID controller can struggle to provide a satisfactory response for the nonlinear systems faced by industry. In addition, conventional PID controllers have a trade-off between performance and robustness, where they cannot compensate for both without compromising stability or speed. In this paper, a novel Nonlinear gains Proportional, Integral, and Derivative (NLPID) control algorithm is proposed as a practical control strategy that shows improvements in the simultaneous set-point tracking and disturbance rejection, to control nonlinear systems. The paper shows the performance and robustness of the proposed controller for the case of a First Order Plus Time Delay (FOPTD) system, which heavily exists in industry. The Particle Swarm Optimization (PSO) algorithm is used to tune the proposed NLPID controller. The performance of the proposed NLPID controller is simulated and compared against established controllers in literature such as conventional PID, two degree of freedom PID, and Smith Predictor PID controllers in MATLAB/Simulink for an FOPTD system, with various uncertainties and disturbances. This study shows that the proposed NLPID controller maintains faster settling and rise time, with no overshoot and excellent disturbance rejection, without compromising stability or speed, and is robust against parametric, additive, and multiplicative uncertainties.

1. Introduction

The Proportional, Integral, and Derivative (PID) controller takes the form of three gains, combining linearly the past errors (integration), present errors (proportional), and the future estimates of error (derivative). The transfer function and time-domain representations of the conventional PID control are given, respectively, as follows:

$$K_{\mathrm{PID}}(s) = k_{\mathrm{p_c}} + k_{\mathrm{i_c}}\frac{1}{s} + k_{\mathrm{d_c}}s \quad u_{\mathrm{PID}}(t) = k_{\mathrm{p_c}}\epsilon(t) + k_{\mathrm{i_c}}\int_0^{t_f}\epsilon(t)\,dt + k_{\mathrm{d_c}}\dot{\epsilon}(t) \tag{1}$$

where $k_{\mathrm{p_c}}, k_{\mathrm{i_c}}$, and $k_{\mathrm{d_c}}$ are the proportional, integral, and derivative gains, respectively, $\epsilon(t)$ is the feedback error and t_f is the integration time.

The PID controller is one of the most ubiquitous control systems that exist among all feedback control systems [1]. The linearity and simplicity of the controller make it useful in industrial applications, which is why it has received a lot of attention from

* Corresponding author.
E-mail address: S.Charkoutsis@2021.ljmu.ac.uk (S. Charkoutsis).

https://doi.org/10.1016/j.rico.2023.100289
Received 13 June 2023; Received in revised form 11 August 2023; Accepted 30 August 2023
Available online 2 September 2023

researchers and engineers over the years [2]. It is well known that industry faces nonlinear systems with process disturbances and modelling uncertainty [3,4]. The PID controller forms a single-degree-of-freedom (1DoF) control structure that poses a trade-off between performance and robustness [3,5]. Once the PID controller has been tuned for optimal disturbance rejection, an overshoot will likely appear at the set-point response, and once it has been tuned to eliminate the overshoot, a slow disturbance rejection will likely be observed [1,3,6]. This can be improved by tuning the controller to find a balance between minimizing overshoot and producing fast disturbance rejection [2]. However, this makes it necessary for the appearance of overshoot with large input costs in systems of higher than first order and nonlinear systems [1,3,6]. To reduce the need for complex tuning algorithms, gain scheduling and real-time algorithms have been proposed as a remedy to the issue of using a 1DoF PID controller [2,7]. However, these algorithms can take large memory in the controller hardware, which can increase costs [1,2,7,8].

An alternative method of eliminating overshoot without compromising on disturbance rejection is by using a two-degree-of-freedom (2DoF) PID control structure [1,3,5,6]. This allows one to independently tune a second loop for robust disturbance rejection while maintaining the tuning of the PID controller in the other loop to eliminate overshoot at the set-point response [1,3,6]. However, this requires complex, multi-objective tuning algorithms and requires a larger parameter search space that takes long computational time and large memory [2,6,9]. Research has been conducted to reduce tuning efforts and maintain the effectiveness of the 2DoF structure, with an improved control methodology that uses linear functions to express variable PID gains, based on time and feedback-error [10–13]. Using this method, one can avoid the excessive usage of computer memory to store different tuning values, and instead use a function to compute the new tuning based on the feedback error [10–13]. Although this can overcome many of the PID challenges and improve systems, industry is suffering from nonlinear dynamics that cannot be adequately compensated by linear function gains PID controllers [10,14,15].

An alternative commonly proposed in research is the use of nonlinear functions to describe the PID gains, which is also known as a Nonlinear PID (NLPID) control structure [11–13,16]. Many NLPID controllers have been proposed to provide an efficient control alternative for specific nonlinear systems [11–13,15]. However, NLPID control has been under an active and continuing field of research for the development of industrial NLPID controllers that can work for a class of nonlinear systems [11–13,17].

Modern NLPID controllers have had a resurgence in research and industrial applications, with the use of Passivity based theory and an enlarged set of nonlinear functions that have increased the scope of research. In more recent research, a different set of nonlinear functions have been proposed, where the proportional, integral, and derivative functions are all described by a unique method. One such example is the use of the Gaussian error function to compute the gains, where the proportional and derivative gains are increasing with increasing error, while the integral gain is reduced with increased error [18]. An extension to the Popov criterion-based NLPID control design has also been proposed for a larger set of nonlinear functions, including exponential and error power functions [19]. An improved version of the nonlinear tracking differentiator together with Han's nonlinear PID controller have also been proposed with the tracking differentiator used both in feedback and at set-point [20]. Moreover, a nonlinear PID controller has been proposed that generates large proportional gain at large error that reduces for small errors, while its integral gain is zero at large errors and increases at steady-state [21]. This controller works together with a nonlinear derivative function that depends on the error and error rate and has been shown to improve performance when compared to conventional methods [21]. A nonlinear PID controller that utilizes only a scalable integral nonlinearity has a stability proof and provides adequate responses to linear and delay-type systems [15]. A classical PID controller with derivative filtering has been combined with nonlinear functions in the feedforward and feedback loops to compensate for the intrinsic nonlinear unstable dynamics of a magnetic levitation system, improving transient response and disturbance rejection [16]. A nonlinear PID controller has also been used for the temperature control of a nonlinear Continuous Stirred Tank Reactor (CSTR) using a local model with internal model control tuning PID combined with fuzzy fusion [22]. Nonlinear PID has also been used for the improvement of stability in the hydraulic drive control of an excavator using nonlinear proportional gain [11]. In addition, in one of the most recent papers, a nonlinear PID controller has also been used in combination with the MIT adaptation rule for a set of nonlinear gains that achieved improved results when compared to the conventional and nonlinear PID methods [12]. An intelligent nonlinear PID controller has also been designed recently, combining neural networks as weights describing the gains and a particle swarm optimization algorithm to tune the weighting parameters to control the temperature of a CSTR system [17]. Finally, in another recent paper, a cross-coupled nonlinear PID controller has been implemented for a highly-coupled twin rotor MIMO system showing improved performance [13].

The main contribution of this paper is the proposal of a novel nonlinear PID controller that provides an effective control scheme with the following specifications:

1. Gains are described by a new set of nonlinear functions that follow a clear strategy for improving the simultaneous set-point tracking and disturbance rejection.
2. It has comparable input energy which is an advantage taken into consideration its strong robustness properties.
3. It achieves low rise-time with no overshoot for any step set-point function.
4. The control is robust against a range of uncertainties.

In this paper, the authors aim to answer the question of whether there is such a nonlinear PID controller that can work for a general class of systems, focusing on the most commonly seen type in industry, the FOPTD systems. In addition, nonlinear PID controllers that use Popov criterion indicate stability, however oscillatory type responses remain and overshoots persist. Many nonlinear PID controllers also have limitations of performance depending also on the set-point, where in this paper the proposed controller aims at maintaining its performance for any step-type set-point function. The proposed nonlinear PID controller is also addressing the limitations of the PID controller and establishes an improved response that can only be achieved by a two-degree-of-freedom system.

In the efforts to provide evidence of stability for the controller a Simulation-based Extensive Testing (SET) method has been conducted with input and output disturbances applied to the feedback system to show internal stability, using the L_2 norm. Finally, different types of uncertainty is conducted, such as parametric uncertainty, additive uncertainty, and multiplicative uncertainty to provide evidence that the controller will work for a broader class of FOPTD type systems without losing stability.

In the sections that follow, the novel NLPID controller proposed in this paper is presented in Section 2. Then, the tuning methodology and the particle swarm optimization algorithm used for the proposed controller is also shown in Section 3. Section 4 shows the results from the bench-marking of the controller against the PID, 2D_PID, and Smith Predictor PID controllers in a widely used industrial system. Section 5 shows the robustness of the proposed NLPID controller under a variety types of uncertainty. Finally, in Section 6 the conclusions and further work are presented to summarize the results found within this research and propose future directions.

2. Novel NLPID controller

In this paper, a novel nonlinear controller is proposed. The gains of the controller are time varying and their values depend on the magnitude of the error and the error rate. These gains are defined using a set of nonlinear functions that are defined to improve transient response and maintain robustness. The equation of the proposed NLPID controller follows a similar format to that of a parallel linear PID controller. The time-domain NLPID controller equation is given by:

$$\begin{aligned} u_{NLPID}(\epsilon(t),\dot{\epsilon}(t),r(t)) =& k_{\mathrm{p}}(\epsilon(t),r(t))\epsilon(t)+ \\ &+\ k_{\mathrm{i}}(\epsilon(t),r(t))\int_0^{t_{\mathrm{f}}}\epsilon(t)\,\mathrm{d}t + k_{\mathrm{d}}(\dot{\epsilon}(t),r(t))\dot{\epsilon}(t) \end{aligned} \tag{2}$$

The proposed NLPID controller is developed to generate fast set-point tracking, with no overshoot and a fast disturbance rejection. Under these requirements, the PID gains which most influence the overshoot negatively are the proportional and integral gains. When large proportional and integral gains are used, the controller generates an oscillatory response with a large overshoot. However, this also provides a fast response and fast disturbance rejection. As a result, in order to remove the overshoot, one can generate a large proportional signal at large error with a small integral signal at large error. This provides the fast tuning that is required, and then once the output reaches close to steady-state, the proportional gain must rapidly decrease and the integral gain must rapidly increase to correct for any steady-state errors. The derivative gain takes a similar form to the integral. However, in this case the derivative gain considers the error rate, so that once the error rate becomes rapid, the gain becomes zero to eliminate noise and derivative kicks. According to this knowledge of PID control behaviour, which is well known within the literature, the proposed NLPID controller is designed with nonlinear functions that must have this property. The nonlinear function that processes such a property is the mollifier function that originates from distribution theory and have not been used in the past within the NLPID control literature. The mollifier takes the mathematical form of:

$$M(x(t)) = \begin{cases} e^{\left\lfloor \frac{1}{|x(t)|^2-1} \right\rfloor} & \text{if } |x(t)| < 1 \\ 0 & \text{if } |x(t)| \geq 1 \end{cases} \tag{3}$$

Moreover, in this paper the mollifier is adopted such that the nonlinearity is applied at the transient response region, to maximize the effect of the nonlinearity for the minimization of overshoot. The adopted nonlinear gains for the proposed NLPID controller are hence described and shown as follows:

The proportional gain

The proportional nonlinear gain is represented by the following function:

$$k_p(\epsilon(t),r(t)) = \begin{cases} ak_0 - k_0 e^{\left\lfloor \frac{1}{\left|\frac{\epsilon(t)}{r(t)}\right|^2-1} \right\rfloor} & \text{if } \left|\frac{\epsilon(t)}{r(t)}\right| < 1,\ r(t)\neq 0 \\ ak_0 & \text{if } \left|\frac{\epsilon(t)}{r(t)}\right| \geq 1,\ r(t)\neq 0 \\ ak_0 - k_0 e^{\left\lfloor \frac{1}{|\epsilon(t)|^2-1} \right\rfloor} & \text{if } |\epsilon(t)| < 1,\ r(t)=0 \\ ak_0 & \text{if } |\epsilon(t)| \geq 1,\ r(t)=0 \end{cases} \tag{4}$$

where k_0 is the proportional constant gain, a is the mean or shift value of the nonlinear function that places the higher gain bounds at either higher or lower values directly related to a and k_0. The function is also dependent on the set-point function $r(t)$, which enlarges and shrinks the non-linearity so that the controller behaves non-linearly in the appropriate error range. Fig. 1 shows an example of a proportional gain k_{p} that is constructed by this function and tuned for certain values of a, k_0 and $r(t)$.

The proportional nonlinear gain is designed to produce the largest proportional signal at large errors, to compensate the error with fast transient response. The value then rapidly decreases smoothly to avoid the overshoot as much as possible. This proportional

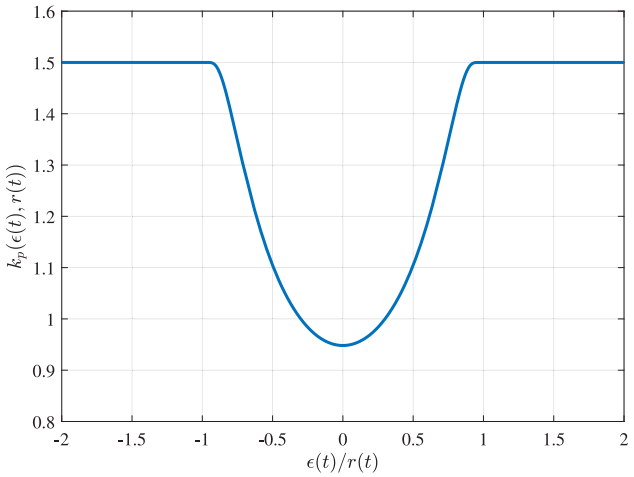


Fig. 1. An example of tuned nonlinear proportional gain shown for a step set-point function $r(t) = 1$, for values $a = 1$ and $k_0 = 1.5$.

gain then helps meet the criteria of fast transient response with no overshoot. This is then able to improve the limitation of the conventional PID control.

The integral gain

The integral nonlinear gain is represented by the following function:

$$k_i(\epsilon(t), r(t)) = \begin{cases} k_1 e^{\left[\frac{1}{\left|\frac{\epsilon(t)}{r(t)}\right|^2 - 1}\right]} & \text{if } \left|\frac{\epsilon(t)}{r(t)}\right| < 1,\ r(t) \neq 0 \\ 0 & \text{if } \left|\frac{\epsilon(t)}{r(t)}\right| \geq 1,\ r(t) \neq 0 \\ k_1 e^{\left[\frac{1}{|\epsilon(t)|^2 - 1}\right]} & \text{if } |\epsilon(t)| < 1,\ r(t) = 0 \\ 0 & \text{if } |\epsilon(t)| \geq 1,\ r(t) = 0 \end{cases} \tag{5}$$

where k_1 is the integral constant that determines the largest value of the integral nonlinear gain. In this case, the set-point function $r(t)$ also affects the gain where it enlarges and shrinks the integral non-linearity in order to adapt to the error range, so that the non-linearity is active throughout the transient response.

The integral gain, shown in Fig. 2, is built according to Eq. (5) and it is designed so that it starts from a value of zero and increases as the error approaches steady-state, approaching its maximal bounded value. This allows for the integral to error-correct the system during steady-state while keeping a low integral value during the transient response, which helps maintain low overshoot.

The derivative gain

Finally, the derivative nonlinear gain is represented by the following function:

$$k_d(\dot{\epsilon}(t), r(t)) = \begin{cases} k_2 e^{\left[\frac{1}{\left|\frac{\dot{\epsilon}(t)}{r(t)}\right|^2 - k_3^2}\right]} & \text{if } \left|\frac{\dot{\epsilon}(t)}{r(t)}\right| < k_3,\ r(t) \neq 0 \\ 0 & \text{if } \left|\frac{\dot{\epsilon}(t)}{r(t)}\right| \geq k_3,\ r(t) \neq 0 \\ k_2 e^{\left[\frac{1}{|\dot{\epsilon}(t)|^2 - k_3^2}\right]} & \text{if } |\dot{\epsilon}(t)| < k_3,\ r(t) = 0 \\ 0 & \text{if } |\dot{\epsilon}(t)| \geq k_3,\ r(t) = 0 \end{cases} \tag{6}$$

where k_2 is the derivative constant that increases the maximum derivative value, $r(t)$ is the set-point function which can enlarge and shrink the nonlinearity accordingly in a similar behaviour to the previous nonlinear gains. For the derivative gain, as it can be noticed, the input to the nonlinear derivative function is the error rate instead of the error. This helps the controller to easily identify

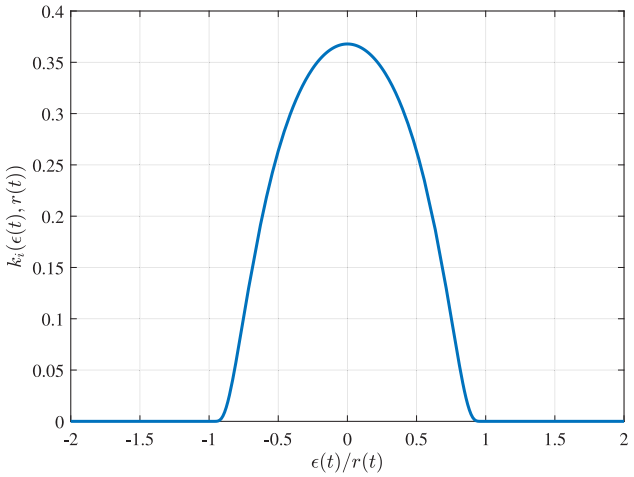


Fig. 2. An example of tuned nonlinear integral gain shown for a step set-point function $r(t) = 1$, for value $k_1 = 1$.

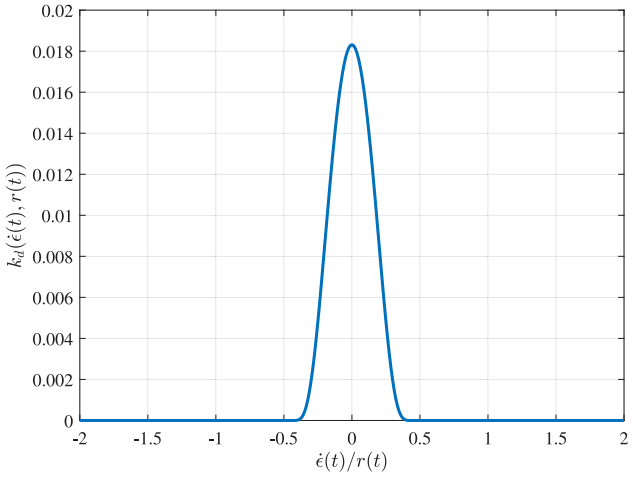


Fig. 3. An example of tuned nonlinear derivative gain shown for a step set-point function $r(t) = 1$, for values $k_2 = 1$ and $k_3 = 0.5$.

the point of steady-state, where the derivative gain is maximized for increased damping, minimizing overshoot, while becoming zero at error rate values higher than the filter constant k_3.

The constant k_3 is the filtering constant which is a design value determined by the designer according to the amount of derivative needed to be included in the controller. This constant is useful to overcome some of the well-known PID limitations. It reduces the derivative kick and reduces the impact of high-frequency noise that might affect the controller input. It changes the range at which the nonlinearity operates and defines the points of noise and derivative kick cancellation. This means that the control designer has the ability to freely adjust the noise signals that one wants to eliminate.

An example of tuned derivative gain is shown in Fig. 3. It has similar shape to the integral gain, where the difference between the two gains is controlled by the design filtering constant k_3.

The nonlinear derivative gain is designed to minimize the effects of noise and derivative kicks in the feedback response of the system. This improves on the common limitations of the conventional PID controller and improves the system input signal. Finally, the derivative gain is maximized near steady-state to eliminate overshoot and improve the speed of the response, which enhances transient performance.

The effect of a changing set-point to the proposed nonlinear proportional gain is shown in Fig. 4 where the larger the set-point becomes, the wider the nonlinearities are, preserving the design constants, such as the maximum value of the gains, the minimum value of the gains, and so that the nonlinearities are active within the range $-\epsilon_{max} \leq \epsilon \leq \epsilon_{max}$. Similar impact also occurs on the integral and derivative gains as discussed above. Having $r(t)$ in the definition of the three gains, makes them all work in synchronization, according to the reference function. This activates the nonlinearities during the transient response, minimizing overshoot, and allows for the gains to settle to a robust value at steady-state. This design produces the same transient response for

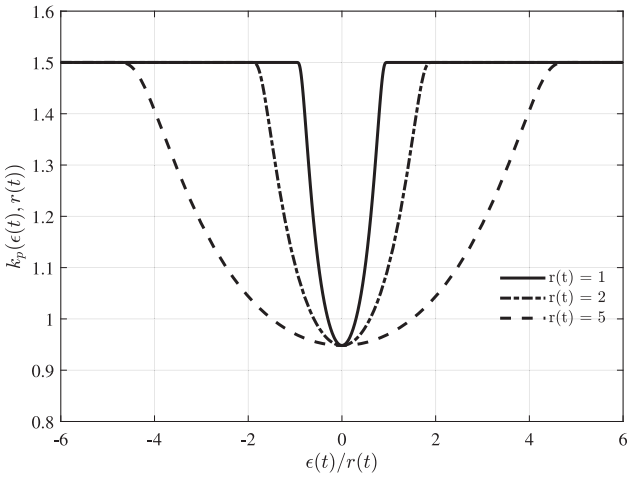


Fig. 4. The tuned nonlinear proportional gain as it adapts to new step set-point values of $r(t) = 1, 2$, and 5.

any given set-point function, improves disturbance rejection, and improves the rejection of any unmodelled uncertainties, as shown later.

In the following section, the tuning methodology of the controllers is shown to ensure fair comparisons with established controllers in literature so that the improvements are shown and are isolated to be due to the proposed nonlinear function gains.

3. Tuning methodology

In this paper, the proposed NLPID controller is aimed at the practitioner to improve the response to commonly seen systems. PID control research has the difficulty and unfortunate disadvantage that many control comparisons are unfair and improved results can be achieved by spending more effort on tuning [1,23]. In addition, control algorithms can possess different number of adjustable parameters, making it difficult to make fair comparisons. Due to these issues, fair comparison of each controller is assured by using a common set of control criteria. The control criteria that are considered in this paper for judging the control performance include both fast set-point tracking with no overshoot, and robustness against disturbance and uncertainties. The controllers are compared based on their ability to meet these criteria using the tools and techniques available to practitioners. MATLAB software is frequently used in industry for the tuning and design of PID and 2D_PID controllers. As a result, this paper compares the proposed NLPID controller against the conventional methods used by the practitioner, by tuning the conventional controllers using MATLAB. This is built on the scope of improving upon the PID and 2D_PID control limitations. The difficulty being when a fair comparison is made, it must be ensured that the right tuning approach is taken and is made transparent.

3.1. Particle Swarm Optimization Algorithm

The Particle Swarm Optimization (PSO) algorithm is one of many evolutionary and stochastic optimization approaches that is simple and effective in solving complex optimization problems. Although many optimization algorithms exist in literature, which have been used in specific applications, see for example [24,25], most engineers and control systems researchers use the particle swarm optimization algorithm for control tuning due to its effectiveness, simplicity, and fast convergence without the use of derivatives [17,23,26,27]. For this reason, the PSO is used in this paper to tune the proposed NLPID controller. However, the downside is that it is easy for PSO to fall to a local minimum [17,28]. To overcome this challenge, two steps have been considered to lower the possibility of such an occurrence. Firstly, the particles are limited within a range of specified values. Secondly, the personal best value of each particle, also known as cognition, is not considered in this case. The global, also known as social intelligence, is used, taking the social best objective value to make it more difficult to fall at a local optimum.

The tuning of the proposed NLPID parameters k_0, k_1, k_2, and a are conducted using the objective function and optimization problem designed with the Integral Time Absolute Error (ITAE) performance measure and the settling time of the system, which can be mathematically expressed as:

$$\begin{aligned} \underset{k_0,k_1,k_2,a}{\text{minimize}} \quad & f(t,\epsilon(t),t_s) = \int_0^{t_f} t|\epsilon(t)|\,\mathrm{d}t + t_s \\ \text{subject to} \quad & k_{min} \leq k_0, k_1, k_2 \leq k_{max}, \\ & a_{min} \leq a \leq a_{max} \end{aligned} \tag{7}$$

where t_s is the settling time, $\epsilon(t)$ is the feedback error, and t_f is the final time. The optimization problem is defined with the parameter constraints within a specified range to ensure stability and to lower the chances of trapping inside local optima. The

objective function is defined to reduce the feedback error in the shortest time possible, meaning minimization of rise-time and overshoot. In addition, adding the settling time means approaching steady-state faster, which helps meet the proposed design criteria. The parameter k_3 is a filtering parameter and is determined by the designer according to the system and the amount of derivative that is necessary. This means that it is not necessary to use the PSO algorithm for tuning this parameter as it can be adapted and changed by the designer after the controller has been tuned.

The PSO algorithm iterates until the final iteration has been reached, with the following steps [28]:

1. Generate n number of random position particle vector X_0^n in the range $[k_{min}, k_{max}]$ for k_0, k_1, k_2 and $[a_{min}, a_{max}]$ for a.
2. Assume initial velocity vector $V_0^n = 0$.
3. Simulate the control system in Simulink.
4. Compute $f(t, \epsilon(t), t_s) = \int_0^{t_f} t|\epsilon(t)|\,\mathrm{d}t + t_s$.
5. If values surpass the defined range, re-initialize a random number in range $[k_{min}, k_{max}]$ for k_0, k_1, k_2 and re-initialize a random number in range $[a_{min}, a_{max}]$ for a.
6. then compute the new velocity and position values, which are modified to be as:

$$ {}_kV_{i+1}^j = |{}_kV_i^j +_k r_i^j c_1({}_kGbest_i -_k P_i^j)| \tag{8} $$

$$ {}_kX_{i+1}^j = |{}_kX_i^j +_k V_{i+1}^j| \tag{9} $$

7. Re-iterate.

where k represents a natural number taking values 1 to 4, iterating between the 4 parameters in the parameter set, ${}_kV_i^j$ is the velocity vector for each iteration i, particle j, and tuning parameter k, ${}_kX_i^j$ is the position vector for each iteration i particle j, and tuning parameter k, ${}_kr_i^j$ is the stochastic variable that changes for every iteration and lies in the range $[0, 1]$, $Gbest$ is the minimum value of the objective function of all particles across iterations, each particle representing a specific tuning parameter set $P_i^j[k_0, k_1, k_2, a]_i^j$. If the new position ${}_kX_{i+1}^j$ is outside the specified range of values, then these specific new particles are re-initialized within the pre-specified range. The parameter $c_1 = 1.3$ is a tuning parameter taken from research surveys on PSO [28].

This process is a modification of the particle swarm optimization, which included the history of the minimum objective value for each particle, in this case only the social best values are considered. The modified PSO algorithm searched for the nonlinear gain parameters k_0, k_1, k_2, and a that minimize settling time, overshoot, and transient response as per the design constraints.

In the following section an FOPTD simulation example is established. The benchmarking between PID, 2D_PID, and SP_PID against the proposed NLPID controller is established for the set-point tracking and disturbance rejection criteria.

4. Simulation example

A commonly seen class of industrial nonlinear systems is the First Order Plus Time Delay (FOPTD) systems. For instance, FOPTD models are used widely to represent plants in process and chemical engineering systems. For that reason, the FOPTD is commonly used as a benchmark process for PID controllers and it can be described by the following transfer function [29]:

$$ P(s) = T(s)e^{-s},\ T(s) = \frac{1}{s+1} \tag{10} $$

The nominal plant represented by Eq. (10) is used for the control benchmarking and all controllers are designed with the following control criteria:

- Minimization of overshoot $\leq 2\%$.
- Minimization of rise time and settling time.
- Fast Disturbance rejection to input and output disturbances.

Using these control criteria, all controllers have been tuned appropriately and are benchmarked against the proposed NLPID controller. The conventional controllers are shown and designed using MATLAB tuning algorithm, which is also explicitly shown. The simulation results from the benchmark tests is then shown for the response of each controller to set-point tracking and disturbance rejection. Then, the L_2 norm is computed at the system input signal, so that the energy and internal stability of the controllers is explicitly shown. This process reassures the effectiveness of the results and the fair benchmark comparison against the conventional controllers.

4.1. Controller benchmarking to set-point tracking and disturbance rejection

The proposed NLPID controller is focused on simultaneous set-point tracking and disturbance rejection. Hence, the benchmarking is based on the claim that the proposed NLPID controller is providing an improvement when compared to the conventional controllers in both set-point tracking and disturbance rejection. The proposed NLPID controller is benchmarked against the conventional and state-of-the-art methods of controlling an FOPTD system as an example case, to create a comparison of the advantages provided by the different methods in FOPTD systems. This claim is tested with both input and output disturbances, as shown by the schematic block diagram in Fig. 5.

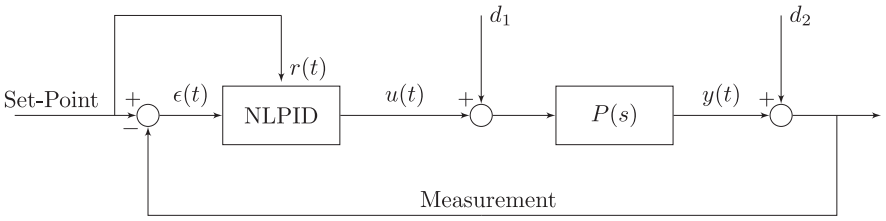


Fig. 5. The schematic block diagram of the control system with both the input and output disturbances.

Table 1
Tuned values of the control parameters used for the benchmarking simulations.

Controller	Tuning parameters
NLPID	$k_0 = 1.9344, k_1 = 1.7142, k_2 = 1.2373, k_3 = 0.5, a = 0.6965$
PID	$k_{p_1} = 0.4458, k_{i_1} = 0.4422, k_{d_1} = 0, N_1 = 0$
2D_PID	$k_{p_2} = 0.5308, k_{i_2} = 0.4743, k_{d_2} = 0, N_2 = 0, b = 0.9400, c = 0$
SP_PID	$k_{p_3} = 1.4089, k_{i_3} = 2.1239, k_{d_3} = 0.4227, N_3 = 2.4471$
T2_PID	$k_{p_4} = 0.6357, k_{i_4} = 0.6306, k_{d_4} = 0.0227, N_4 = 0.5000$

The tuning of the proposed NLPID controller is established using the proposed control criteria and the FOPTD process, represented by Eq. (10). To find the appropriate tuning, the PSO optimization problem is first developed by identifying the appropriate parameter constraints. According to the FOPTD dynamics, the parameter constraints are determined based on the instability regions of the PID controller and based on a series of PSO tuning trials to refine the constraints. The optimization problem is then defined as:

$$\begin{aligned} \underset{k_0,k_1,k_2,a}{\text{minimize}} \quad & f(t,\epsilon(t),t_s) = \int_0^{t_f} t|\epsilon(t)|\,\mathrm{d}t + t_s \\ \text{subject to} \quad & 0 \le k_0, k_1, k_2 \le 2, \\ & 0.5 \le a \le 2 \end{aligned} \tag{11}$$

where the range of values of $[0, 2]$ for k_0, k_1, and k_2 and the range $[0.5, 2]$ for a are specified to guarantee stability of the FOPTD system Eq. (10) and to minimize the chances of falling into local optima. The parameter $k_3 = 0.5$ is used, since FOPTD systems are well-known to be adequately controlled by a PI controller. Hence, there is less need for a derivative action, which means a low value of k_3 can be used. The PSO algorithm determined the parameters of the nonlinear controller gains as shown in Table 1. The tuned parameter values can be used to plot the nonlinear function gains and the same results with slightly shifted values will appear as in Figs. 1, 2, and 3 from Section 2.

In this paper, MATLAB PID and 2D_PID control parallel structures are used, containing derivative filtering for reducing derivative kick effects, improving control stability, and performance. The transfer function representations are described as:

$$K_{\text{PID}}(s) = k_{p_1} + k_{i_1}\frac{1}{s} + \frac{k_{d_1}N_1}{1 - \frac{N_1}{s}} \tag{12}$$

$$K_{\text{2D\_PID}}(s) = k_{p_2}(br - y) + k_{i_2}\frac{1}{s} + \frac{k_{d_2}N_2}{1 - \frac{N_2}{s}}(cr - y) \tag{13}$$

where k_{p_1}, k_{i_1}, and k_{d_1} are the proportional, integral, and derivative gains of the PID controller respectively, $N_{1,2}$ are the filtering parameters, which represent the inverse of the time constant of the filter, and k_{p_2}, k_{i_2}, and k_{d_2} are the proportional, integral, and derivative gains of the two degree of freedom PID controller with b and c the set-point weightings, as a percentage of the set-point that is contained within the error and error rate, respectively. These are the respective parameters to be tuned by MATLAB algorithm, according to the performance criteria. As can be seen, the PID controller has 4 parameters to be tuned, including the filter, while the 2D_PID controller has 6 parameters, which include the set-point weighting.

An alternative method for controlling FOPTD systems is the SP_PID control. SP_PID controllers provide an improvement of the PID controller specific to delay systems, where the delay dynamics are predicted and then a PID controller is also used and tuned. Research indicates that this method provides improved results as compared to PID control in delay systems [30,31]. It is used in this paper as a fair comparison and extending the simulations to more complex and industry-used control systems. With the use of the smith predictor one can achieve faster response with a minimal overshoot that can improve the PID controller with a time-delay prediction step.

The SP_PID controller design is formulated using MATLAB PID controller described by Eq. (12) with the FOPTD plant model described by Eq. (10). The Pade approximations are used to approximate the delay $e^{-\tau s}$ as a realizable transfer function $G_{\text{p}}(s)$ used

to design the SP_PID controller, as follows:

$$G_{\mathrm{p}}(s) = \frac{\tau^2 s^2 - 6\tau s + 12}{\tau^2 s^2 + 6\tau s + 12} \tag{14}$$

with the SP_PID control transfer function being:

$$K_{\mathrm{SP\_PID}}(s) = \frac{K_{\mathrm{PID}}(s)}{1 + K_{\mathrm{PID}}(s)T(s)(1 - G_p(s))} \tag{15}$$

The SP_PID controller is tuned in a separate attempt to control the FOPTD system using a state-of-the-art control system. There is a plethora of tuning algorithms that exist within the literature to tune PID controllers for FOPTD systems, which can become complex [2,6]. However, in this paper, the PID, 2D_PID, and SP_PID control algorithms are tuned using MATLAB tuning algorithm as a simple and effective method of tuning, available to the practitioner.

MATLAB tuning algorithm works by parameterizing the controller based on the designer's pre-specified value of the cross-over frequency and the phase margin of the controller [32]. The cross-over frequency is directly related to the open-loop system bandwidth, which is directly related to the speed of the response and uses the phase margin to design the robustness of the controller [32]. In the case of a nonlinear plant, MATLAB PID tuning algorithm initially linearizes the system at the operating point. Then it tunes the PID controller according to the linearized model, at that operating point.

The parameterization of the controller allows the designer to directly visualize the response according to the set design criteria, which can be changed in real-time. Making it a simple, effective, and easy-to-learn method of tuning. The parameterization of the controller used by the algorithm, can be written as [32]:

$$C(s) = \frac{\omega_c}{s}\left(\frac{sin(\phi_z)s + \omega_c cos(\phi_z)}{\omega_c}\right)\left(\frac{sin(\beta)s + \omega_c cos(\beta)}{sin(\alpha)s + \omega_c cos(\alpha)}\right) \tag{16}$$

where ω_c is the frequency at which the magnitude of the open-loop response $Y(s) = K_{\mathrm{PID}}(s)P(s)$ first crosses the 0 dB line, and angles ϕ_z, α, and β vary between 0 and 90 degrees, with a total phase shift provided by the PID controller at frequency ω_c given by [32]:

$$\Delta\phi = \phi_z + \beta - \alpha \tag{17}$$

In addition, this tuning tool allows for prioritization in robustness, set-point tracking, or a balance of both, which is adopted as the designer requires, and it is also applicable for tuning the 2D_PID control algorithm [32,33]. MATLAB tuner has also been reported in research as an effective tuning method that can provide results for diverse problems, including nonlinear systems, systems with delays, systems with non-minimum phase dynamics, and any linear models [32–35]. This makes MATLAB tuning an easy and available tuning tool for the practitioner, showing improved results when compared to Ziegler–Nichols or other classical tuning rules used in industry [32–35]. As a result, the tuning of the PID, 2D_PID, and SP_PID has been conducted in MATLAB where the tuning is focused on providing a balance between tracking and robustness.

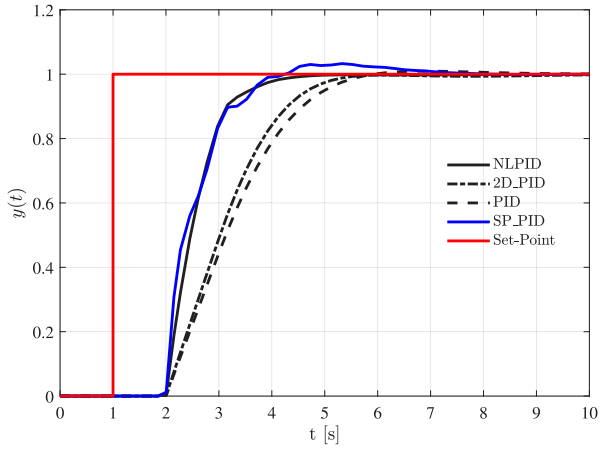
After extensive tuning trials using MATLAB tuning tool in the efforts to increase controller speed and minimize overshoot, the tuned parameters are selected for the benchmarked controllers and are shown in Table 1.

The conventional control methods are benchmarked against the proposed NLPID controller on set-point tracking. The computer that is used to conduct the research has a quadcore intel i7-6700HQ processor with 16 GB RAM memory and a 250 GB SSD memory card. The operating system of the computer is a 64-bit Windows 10. MATLAB/Simulink R2021a software version is installed under the academic license, and is used to conduct the simulations. A variable step-size and solver are used so that they are automatically selected by the software as is best fit for the problem. In the cases of the simulations the automatic solver selected the (Runge–Kutta) ode45 solver with a relative tolerance of 10^{-3}.

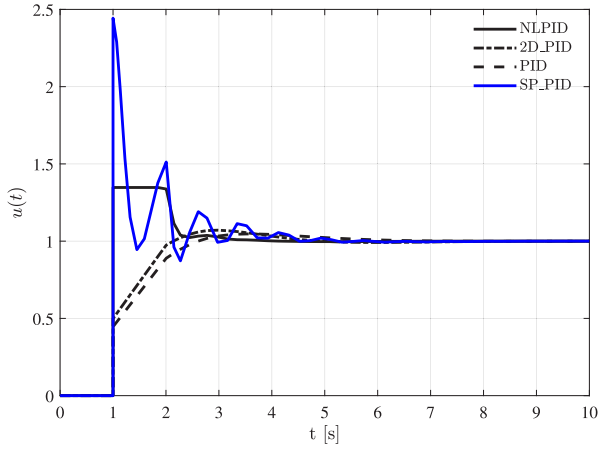
It can be seen by Fig. 6(a) that the proposed NLPID controller outperforms all the conventional control methods and produces no overshoot with the fastest settling time. It can be seen by the figure that the conventional controllers can produce a response with no overshoot, however, they have a significantly slower settling and rise time of approximately 2 s difference. Fig. 6(b) shows the system inputs to the plant model for each of the benchmarked controllers. The NLPID controller shows low system input energy with bounded signals and cannot exceed that limiting value, which eliminates the derivative kicks. The SP_PID controller shows a large sharp input with large system input energy that deteriorates the controller performance and can degrade actuators. In contrast the PID and 2D_PID controllers show a slow response with a smooth input to the process model.

Fig. 7 shows the time variation of the nonlinear gains. It is clear that the gains start from their steady-state values, since the step-function produces no signal until the 1 s time-mark. Then the proportional gain produces its maximum value where it then rapidly drops as it approaches the steady-state value again, converging to a specific gain value. The integral gain starts from steady-state, then goes to zero and increases rapidly during the transient response, settling to a converged value, as the system reaches a steady-state, providing error correction. It can be seen that the derivative also starts from its steady-state value. The derivative gain then increases at the start, when the signal from the step function is produced at the 1 s time mark, but due to the delay the controller receives no response for another second. It then reaches zero, eliminating any derivative kicks and noisy signals at transient response. As the system approaches steady-state, the error and error rate are zero and the derivative gain is increasing to its maximum value. Once steady-state is reached it then increases providing the necessary speed of the system to eliminate overshoot.

To ensure a fair comparison of the controllers, once the proposed NLPID control gains reach the steady-state values, shown by Fig. 7 they are then instead used to tune a separate, second tuning trial PID controller (T2_PID), to ensure that even when the PID controller is tuned with the proposed NLPID steady-state gains, the proposed NLPID still outperforms the PID controller. The filtering



(a) Benchmark comparisons of NLPID controller against conventional PID, 2D_PID, and SP_PID controllers.



(b) System Input benchmark of NLPID against PID, 2D_PID, and SP_PID.

Fig. 6. Controller performance comparison of the set-point tracking of a step-function.

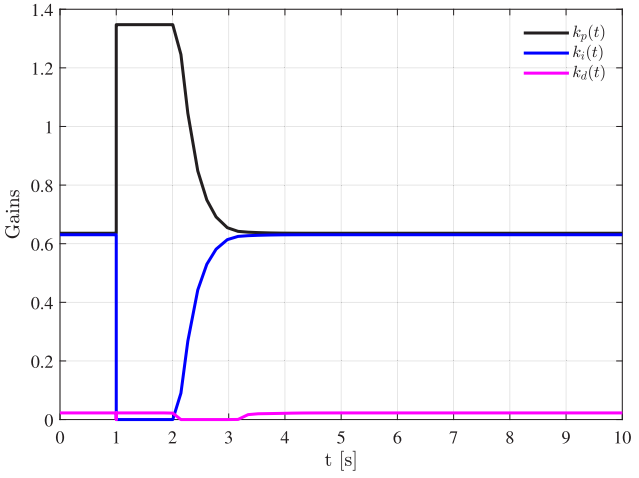
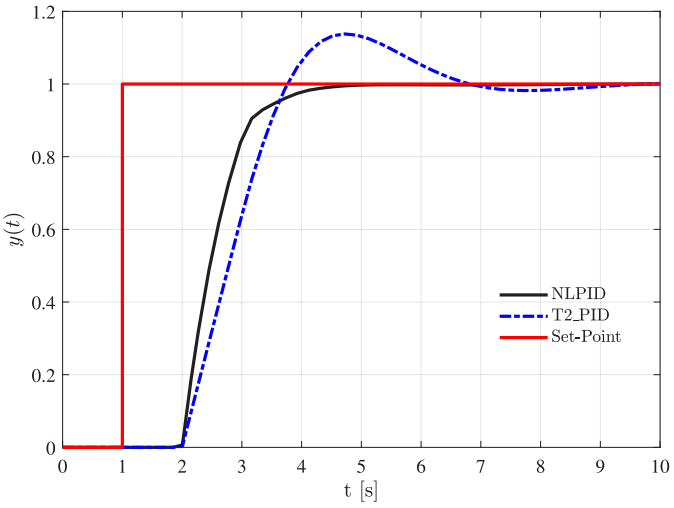
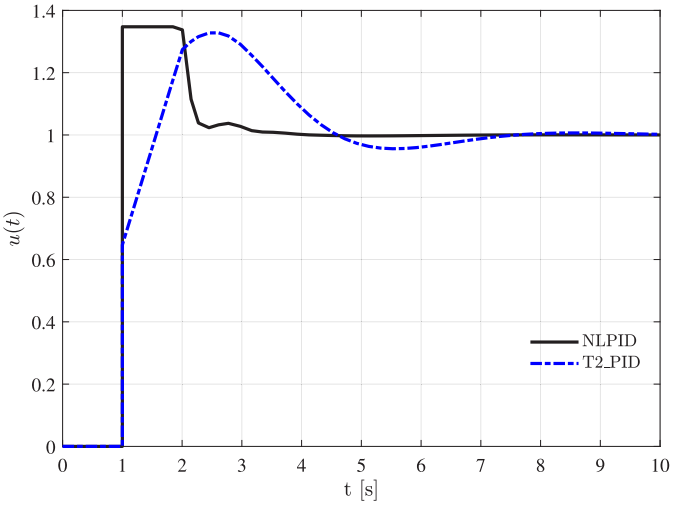


Fig. 7. Nonlinear gain values across the time-span of the simulation.



(a) Benchmarking of NLPID controller against state-of-the-art control methods.



(b) System Input benchmark of NLPID against T2_PID controller due to the set-point step function.

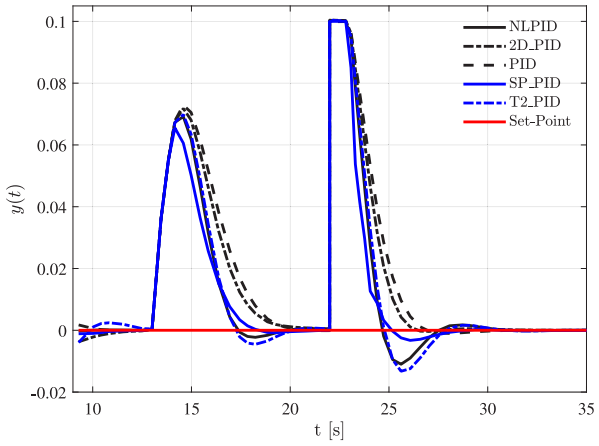
Fig. 8. NLPID controller performance comparison against a second tuning case of a liner PID controller.

value N_4 is given the value of 0.5, in a similar fashion to the NLPID controller parameter k_3, since there is no significant effect from the derivative, making filtering unnecessary. This can provide additional evidence that the suggested nonlinear functions are providing performance improvement, excluding the possibility that the nonlinearities can be replaced by a linear controller using the gain values at steady-state as tuning. The tuning of a separate PID controller now becomes as shown in Table 1.

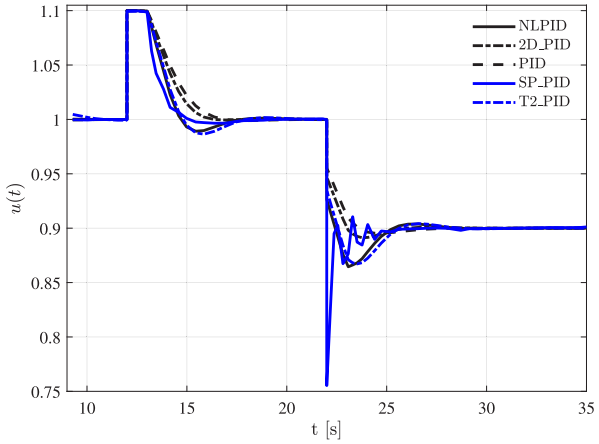
Fig. 8(a) shows that the T2_PID controller generates a large overshoot of approximately 15%. This also indicates the limitations of PID control that when tuned for fast input disturbance rejection, an overshoot appears with a fast transient response. Fig. 8(b) shows the system inputs to the plant for the proposed NLPID and T2_PID controllers. It can be seen that the T2_PID shows a slower input response with higher input energy than PID and 2D_PID.

Following from the set-point tracking response, the controllers are also benchmarked for disturbance rejection. The benchmark testing is conducted using a step-function of final value 10% of the set-point, applied to the system input at 12 s time mark. In addition, a step-function 10% of the set-point has also been taken as the output disturbance at 22 s time mark, which represents a sensor bias and deviation from the true value.

The response to disturbance rejection by the benchmarked controllers is shown in Fig. 9. The output shown in Fig. 9(a) indicates the deviation of the steady-state value produced by the disturbances. The proposed NLPID controller outperforms the PID and 2D_PID controllers in input disturbance rejection, apart from the SP_PID controller, which shows the fastest input disturbance



(a) Controllers comparison to both input and output disturbance rejection response.



(b) NLPID System Input due to both input and output disturbance rejection

Fig. 9. Controller performance comparison of the disturbance rejection showing both the output and system input responses.

rejection. The T2_PID controller shows similar input and output disturbance rejection response to the proposed NLPID controller. The figure also indicates that the PID and 2D_PID controllers outperform the proposed NLPID, SP_PID, and T2_PID controllers in output disturbance rejection, producing a faster settling time. Fig. 9(b) indicates that the SP_PID disturbance rejection comes with a cost of overcompensating signal, as compared to the proposed NLPID control, while the conventional PID, T2_PID, and 2D_PID controllers have a similar system input to the proposed NLPID controller.

The L_2 norm of the system input is computed. This provides satisfactory evidence of internal stability, as well as energy usage. The values below are the computed L_2 norms using the following equation:

$$L_2(u(t)) = \sqrt{\sum_{t_0}^{t_f}(u(t)^2)} \tag{18}$$

Using Eq. (18) the L_2 norm of the system input signals, with the applied input and output disturbances, are computed with their values shown in Table 2.

The proposed NLPID controller shows improvements in performance and maintains fast disturbance rejection, while having a comparable L_2 energy to that of the PID, 2D_PID, and SP_PID controllers. In addition, according to the Simulation-based Extensive Testing (SET) the NLPID controller shows a finite L_2 norm that indicates internal stability of the NLPID controller (see Table 2.) In the following section the robustness of the NLPID controller is shown as part of the Simulation-based Extensive Testing (SET) method. The stability testing that is conducted, is based on the ability of the proposed NLPID controller to maintain a stable feedback system against different types of uncertainties in the system modelling of FOPTD plants.

Table 2
L_2 gain values for each controller.

Controller	L_2
NLPID	10.8960
2D_PID	9.3657
PID	9.3036
SP_PID	10.5629
T2_PID	9.7813

5. Robustness of the proposed NLPID controller to uncertainty

Nonlinear controllers introduce additional nonlinearity into the system, and as a result robustness tests of the proposed controller must be conducted. Robustness tests are used as part of the SET method to show that the proposed NLPID controller has the properties of robust stability and performance, in varying plant dynamics, in the case of inaccurate FOPTD models. The uncertainty is modelled using the general feedback system, without considering input or output disturbances.

The initial test shown is the parametric uncertainty of ±10% parameter variations, which are expected structured model uncertainty. Then, general unstructured uncertainties have been considered, such as additive and multiplicative uncertainties.

5.1. Parametric uncertainty

The parametric uncertainty of the nominal FOPTD plant is modelled using Eq. (19) in the following transfer function format:

$$\overline{P}(s) = \frac{\overline{k}e^{\overline{\tau}s}}{\overline{t}_p s + 1} \tag{19}$$

where $0.9 \leq \overline{k} \leq 1.1$, $0.9 \leq \overline{t}_p \leq 1.1$, and $0.9 \leq \overline{\tau} \leq 1.1$, which models a parameter change of ±10%.

Fig. 10, shows the gain, lag, and delay parametric uncertainty output and system input plots. Fig. 10(a) shows how the proposed NLPID controller responds to a large set of ±10% variations in gain k, lag t_p, and delay parameter τ. It can be seen that there are no large variations of overshoot and no instabilities. In the case where the gain, lag, and delay parameters are underestimated, the response shows a maximum overshoot of approximately 10% and a larger settling time. The figure also shows no effect on stability, providing evidence of robust performance and robust stability for the proposed NLPID controller against gain, lag, and delay variations.

According to the parametric uncertainty study, it can be seen that the proposed NLPID controller shows resilience to parameter variations in a structured model uncertainty. The uncertainty tests indicate that internal stability is maintained across different types of parameter variations with some changes in performance, showing slower settling time, extending from 4 s of the nominal plant, up to a maximum of 10 s for the extreme variations.

5.2. Additive uncertainty

After conducting tests of the proposed controller against structured uncertainties the next step is to move into the unstructured uncertainties to see whether the proposed controller maintains stability. The tests begin with additive uncertainty, which is modelled using Eq. (20). The uncertainty plant $\Delta(s)$ is designed to be additional lag dynamics into the plant model that may not be considered in the modelling process.

$$\overline{P}(s) = P(s) + \Delta(s) \tag{20}$$

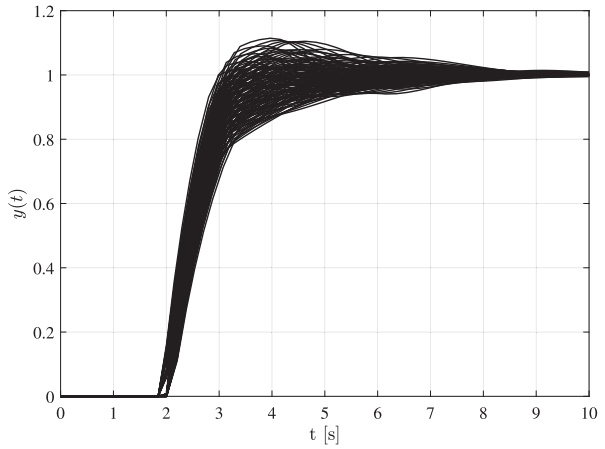
The additive uncertainty has been considered to be an additional 50% of lag dynamics into the system in an unstructured form. Assuming that $\Delta(s)$ is any arbitrary transfer function, satisfying the condition $||\Delta(s)||_\infty \leq 1$ then an arbitrarily large variation of uncertainty is chosen to be [36]:

$$\Delta(s) = \frac{1}{t_p s + 1} = \frac{1}{1.5s + 1} \tag{21}$$

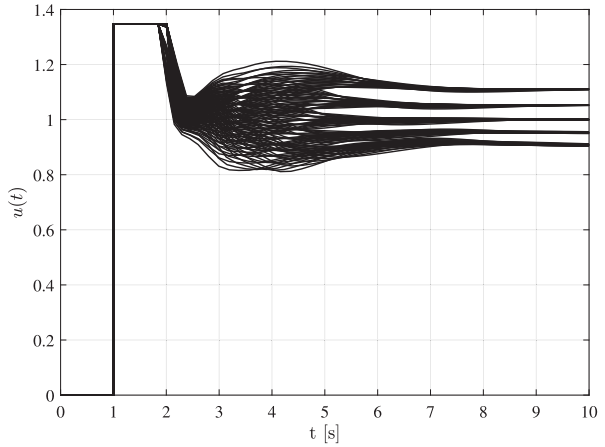
The simulation of the additive uncertainty is conducted under the developed uncertainty. The system is represented as a plant with a minimal realization that can be expressed in the additive form. The worst-case scenario is taken as the primary example for the simulation, depicting the 50% unmodelled lag dynamics.

Fig. 11(a) shows that the proposed NLPID controller shows robust stability to the additional lag dynamics, with a fast settling time of approximately 6 s. Performance deterioration is observed as the plant damping is reduced. The performance of the proposed NLPID indicates that even after a large addition of lag into the system, the controller maintains stability with a slightly reduced performance, observing an overshoot of approximately 10%. Fig. 11(b) shows the system input due to the additive uncertainty. According to the figure, it can be seen that the signal is bounded and hence internally stable.

The proposed NLPID controller can maintain its stability under large deviations in dynamics, which are often seen due to inaccurate modelling. Nonlinear controllers can sometimes have unpredictable outcomes and such interrogation of the controller to large uncertainty provides confidence in the capabilities of the proposed NLPID controller to perform consistently.



(a) NLPID Output response due to gain, lag, and delay parametric uncertainty.



(b) NLPID System Input due to gain, lag, and delay parametric uncertainty

Fig. 10. NLPID Controller response to gain, lag, and delay parametric uncertainty in FOPDT system, showing both output and system input responses.

5.3. Multiplicative uncertainty

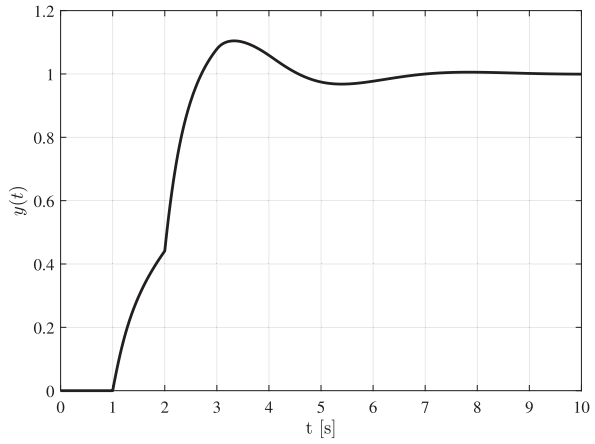
In the effort to do extensive robustness tests, following from the parametric and additive uncertainty, is the multiplicative uncertainty in the delay dynamics and observe the corresponding changes to the proposed NLPID controller response. The multiplicative uncertainty is modelled using Eq. (22) as follows:

$$\overline{P}(s) = P(s)[1 + W(s)\Delta(s)] \tag{22}$$

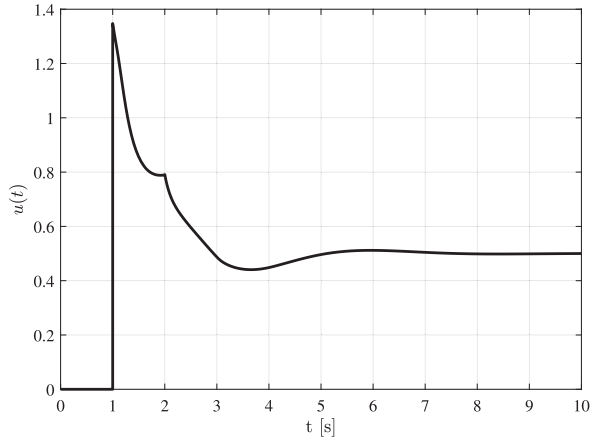
where $\Delta(s)$ being any arbitrary transfer function, satisfying the condition $||\Delta(s)||_\infty \leq 1$.

The following inequality must hold true for any multiplicative uncertainty, indicating a circle of radius equal to the magnitude of $W(s)$ that the system uncertainty must lie away from the $-1 + 0j$ point of the Nyquist plot. The multiplicative uncertainty has been considered to be an unstructured uncertainty of delay dynamics equivalent to 20% time delay from the nominal plant. From the above condition, we can determine the weighting dynamics of the uncertainty as follows:

$$\begin{aligned}\left|\frac{\overline{P}(s)}{P(s)} - 1\right| &= \left|\frac{\frac{1}{s+1}e^{-(1+\lambda)s}}{\frac{1}{s+1}e^{-s}} - 1\right| \\ &= \left|\frac{e^{-(1+\lambda)s}}{e^{-s}} - 1\right| \\ &= |e^{-\lambda s} - 1| \leq |W(s)|\end{aligned}$$



(a) NLPID Output response due to unstructured additive uncertainty.



(b) NLPID System Input due to unstructured additive uncertainty

Fig. 11. NLPID Controller response to unstructured additive uncertainty in FOPTD system, showing both output and system input responses.

for which when the maximum delay uncertainty of 20% occurs at the value of $\left[\lambda_{min}, \lambda_{max}\right] = [0, 0.2]$ that makes the equation into:

$$|e^{-0.2s} - 1| \leq |W(s)| \tag{23}$$

The weighting function $W(s)$ is the transfer function that has been modelled to contain the worst case magnitude of the delay uncertainty magnitude. The weighting function that is recommended to fit the uncertain lag dynamics to be modelled as [36]:

$$W(s) = \frac{\lambda_{max} s}{\frac{\lambda_{max}}{2} s + 1} = \frac{0.2s}{0.1s + 1} \tag{24}$$

The magnitude plot can be shown by the red and blue plots, respectively, in Fig. 12, which shows $W(s)$ estimating the distribution of the worst-case delay uncertainty transfer function magnitude. This forms a more generic unstructured delay uncertainty that can be implemented in more complex controllers to show extensive robustness to a larger set of uncertain dynamics. When compared to parametric uncertainty, which only includes a certain range of values, the degree and structure of the plant dynamics are assumed to be unknown. In this case, unstructured uncertainty allows some flexibility for ignorance in the degree and structure of the dynamics.

The uncertainty $\Delta(s)$ is chosen to be a transfer function that contains the same poles as the nominal plant, the appropriate transfer function selection is modelled as [36]:

$$\Delta(s) = \frac{1}{t_p s + 1} \tag{25}$$

As a result, the total uncertainty dynamics in the multiplicative form can be represented as:

$$W(s) = \frac{0.2s}{0.1s + 1}, \Delta(s) = \frac{1}{s + 1} \tag{26}$$

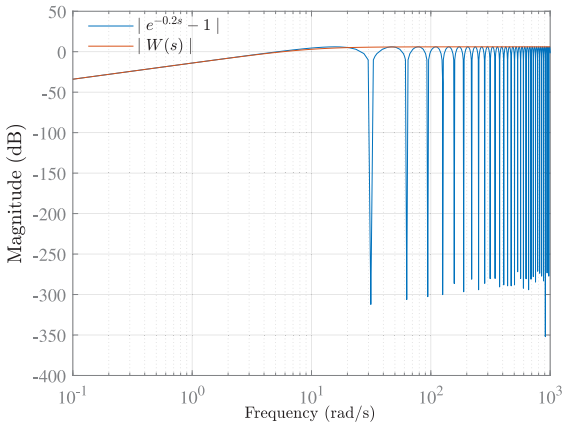


Fig. 12. The bode magnitude response of the uncertainty weighting transfer function and the maximum delay deviation transfer function.

The uncertainty function $\Delta(s)$ represents the uncertainty in the magnitude and phase dynamics and is implemented according to the weighting function $W(s)$, where $||\Delta(s)||_\infty \leq 1$, satisfying the H_∞ condition.

The simulation of the multiplicative uncertainty is conducted under the developed uncertainty. The system is represented as a plant with a minimal realization that can be expressed in the multiplicative form. The worst-case scenario is taken as the primary example for the simulation, depicting the 20% unmodelled delay dynamics.

Fig. 13(a) shows the output response of the proposed NLPID controller, indicating robust stability within a large set of unstructured dynamics of the plant model. It also indicates that the controller suffers from deteriorated performance with an overshoot of less than 10% and a settling time of approximately 8 s. Fig. 13(b) shows the system input from the proposed NLPID controller into the uncertain plant, indicating internal stability to the uncertainty.

The proposed NLPID controller shows robust stability to unstructured multiplicative uncertainty. This shows that with the addition of time-delay uncertainty into the system the proposed controller maintains stability with a performance degradation, as expected. This expands on the results and shows extensive robustness.

6. Conclusions and future work

The PID controller suffers from the limitations of a single-degree-of-freedom algorithm, where the tuning can only focus on one issue, either set-point tracking or disturbance rejection. This requires design compromise and effective tuning methods, which are hard to find, and it depends on the system dynamics. In this paper, this limitation is resolved by the proposed NLPID controller that manages to improve the transient response speed while maintaining fast disturbance rejection of the system. In the benchmarking results, the proposed NLPID controller outperforms the conventional control systems in simultaneous transient response and disturbance rejection for the case of FOPTD systems. The controller also shows robust stability and robust performance to parametric uncertainty with no large variations. However, the main limitation of the proposed NLPID controller is that there are yet no tools of analysis for its stability to allow the designer to compute its margins of stability and robustness. The authors are working on this limitation with promising results that will be published in a separate article. As a result, this depends on extensive simulation trials to different model uncertainties to reassure stability. To show its robustness, extensive tests have been conducted for various plant uncertain models in additive and multiplicative forms. The proposed controller maintains robust stability to large lag dynamics in additive uncertainty form and indicates fast rise time with 10% overshoot and large settling time, indicating deterioration in performance. In the case of multiplicative uncertainty, the proposed controller also shows robust stability to a large variety of delay uncertainty models of the unstructured multiplicative form, providing an extensive stability and robustness study, through simulation.

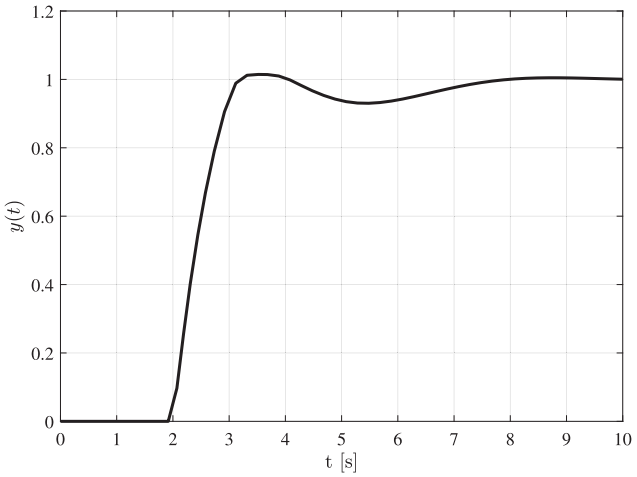
As part of future work, the authors will work on applying the proposed NLPID controller to different plants with nonlinearities and non-minimum phase characteristics. The authors will also work on a novel controller that will contain the proposed NLPID control structure with an extended state observer, with the effect of improving robustness, disturbance rejection, and transient response characteristics.

Declaration of competing interest

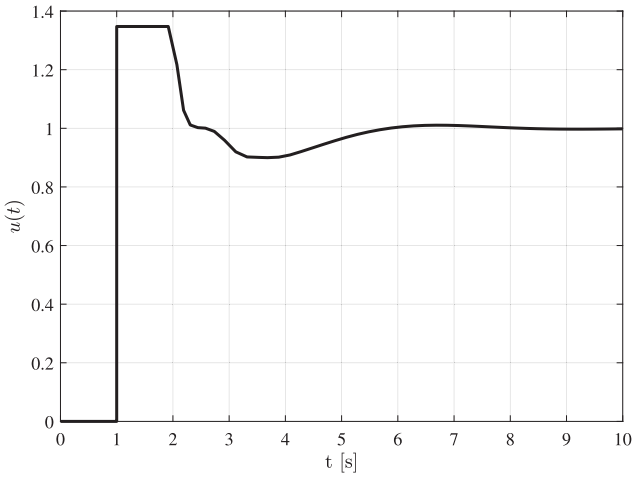
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.



(a) NLPID Output response due to multiplicative uncertainty.



(b) NLPID System Input due to the multiplicative uncertainty

Fig. 13. NLPID Controller response to multiplicative unstructured uncertainty in FOPTD system.

References

[1] Åström KJ, Hägglund T. PID controllers: Theory, design and tuning. 2nd ed. Instrument Society of America; 1995.
[2] O'Dwyer A. Handbook of PI and PID controller tuning rules. 3rd ed. London, Hackensack, NJ: Imperial College Press; Distributed by World Scientific Pub; 2009.
[3] Garpinger O, Hägglund T, Åström KJ. Performance and robustness trade-offs in PID control. J Process Control 2014;24(5):568–77. http://dx.doi.org/10.1016/j.jprocont.2014.02.020.
[4] Bernstein DS. Facing future challenges in feedback control of aerospace systems through scientific experimentation. J Guid Control Dyn 2022;45(12):2202–10. http://dx.doi.org/10.2514/1.G006785.
[5] Chen J, Fang S, Ishii H. Fundamental limitations and intrinsic limits of feedback: An overview in an information age. Annu Rev Control 2019;47:155–77. http://dx.doi.org/10.1016/j.arcontrol.2019.03.011.
[6] Alfaro VM, Vilanova R, Arrieta O. Two-degree-of-freedom PI/PID tuning approach for smooth control on cascade control systems. In: 2008 47th IEEE conference on decision and control. 2008, p. 5680–5. http://dx.doi.org/10.1109/CDC.2008.4738796.
[7] Cetin M, Iplikci S. A novel auto-tuning PID control mechanism for nonlinear systems. ISA Trans 2015;58:292–308. http://dx.doi.org/10.1016/j.isatra.2015.05.017.
[8] Abut T, Soyguder S. Real-time control and application with self-tuning PID-type fuzzy adaptive controller of an inverted pendulum. Ind Robot 2019;46(1):159–70. http://dx.doi.org/10.1108/IR-10-2018-0206.
[9] Qin Y, Zhao G, Hua Q, Sun L, Nag S. Multiobjective genetic algorithm-based optimization of PID controller parameters for fuel cell voltage and fuel utilization. Sustainability 2019;11(12):20. http://dx.doi.org/10.3390/su11123290.
[10] Psonis T, Mitronikas E, Nikolakopoulos P. Comparison of PID and fuzzy PID controller for a linearised magnetic bearing. Tribol Ind 2017;39(3):349–56. http://dx.doi.org/10.24874/ti.2017.39.03.10,.
[11] Liu T. Research on stability of hydraulic system based on nonlinear PID control. Nonlinear Eng 2022;11:494–9. http://dx.doi.org/10.1515/nleng-2022-0222.

[12] Shamseldin MA. Design of auto-tuning nonlinear PID tracking speed control for electric vehicle with uncertainty consideration. World Electr Veh J 2023;14(4):78. http://dx.doi.org/10.3390/wevj14040078.
[13] Sivadasan J, Iruthayarajan MW, Stonier AA, Raymon A. Design of cross-coupled nonlinear PID controller using single-objective evolutionary algorithms. Math Probl Eng 2023;2023:e7820047. http://dx.doi.org/10.1155/2023/7820047.
[14] Feng Y, Wu M, Chen X, Chen L, Du S. A fuzzy PID controller with nonlinear compensation term for mold level of continuous casting process. Inform Sci 2020;539:487–503. http://dx.doi.org/10.1016/j.ins.2020.06.024.
[15] Son Y-D, Bin S-D, Jin G-G. Stability analysis of a nonlinear PID controller. Int J Control Autom Syst 2021;19(10):3400–8. http://dx.doi.org/10.1007/s12555-020-0599-y.
[16] Chong S-H, Chan R-SA, Hasim N, Chong S-H, Chan R-SA, Hasim N. Enhanced nonlinear PID controller for positioning control of maglev system. In: Control based on PID framework - the mutual promotion of control and identification for complex systems. IntechOpen; 2021, http://dx.doi.org/10.5772/intechopen.96769.
[17] Chaturvedi S, Kumar N, Kumar R. A PSO-optimized novel PID neural network model for temperature control of jacketed CSTR: design, simulation, and a comparative study. Soft Comput 2023. http://dx.doi.org/10.1007/s00500-023-09138-0.
[18] Korkmaz M, Aydoğdu Ö, Doğan H. Design and performance comparison of variable parameter nonlinear PID controller and genetic algorithm based PID controller. In: 2012 international symposium on innovations in intelligent systems and applications. 2012, p. 1–5. http://dx.doi.org/10.1109/INISTA.2012.6246935.
[19] Maddi A, Guessoum A, Berkani D. Design of nonlinear PID controllers based on hyper-stability criteria. In: 2014 15th international conference on sciences and techniques of automatic control and computer engineering. 2014, p. 736–41. http://dx.doi.org/10.1109/STA.2014.7086743.
[20] Kasim I, Riyadh W. On the improved nonlinear tracking differentiator based nonlinear PID controller design. IJACSA 2016;7(10). http://dx.doi.org/10.14569/IJACSA.2016.071032,.
[21] So G-B. EA-based design of a nonlinear PID controller using an error scaling technique. SIC 2019;28(3):279–88. http://dx.doi.org/10.24846/v28i3y201904,.
[22] Pugazhenthi P N, Selvaperumal S, Vijayakumar K. Nonlinear PID controller parameter optimization using modified hybrid artificial bee colony algorithm for continuous stirred tank reactor. Polska Akad Nauk Bull Pol Acad Sci Tech Sci 2021;69(3). http://dx.doi.org/10.24425/bpasts.2021.137348.
[23] Joseph SB, Dada EG, Abidemi A, Oyewola DO, Khammas BM. Metaheuristic algorithms for PID controller parameters tuning: Review, approaches and open problems. Heliyon 2022;8(5):e09399. http://dx.doi.org/10.1016/j.heliyon.2022.e09399.
[24] Ahmed I, Rehan M, Basit A, Tufail M, Hong K-S. A dynamic optimal scheduling strategy for multi-charging scenarios of plug-in-electric vehicles over a smart grid. IEEE Access 2023;11:28992–9008. http://dx.doi.org/10.1109/ACCESS.2023.3258859.
[25] Ahmed I, Alvi U-E-H, Basit A, Rehan M, Hong K-S. Multi-objective whale optimization approach for cost and emissions scheduling of thermal plants in energy hubs. Energy Rep 2022;8:9158–74. http://dx.doi.org/10.1016/j.egyr.2022.07.015.
[26] Abushawish A, Hamadeh M, Nassif AB. PID Controller Gains Tuning Using Metaheuristic Optimization Methods: A survey. Int J Comput 2020;14:87–95. http://dx.doi.org/10.46300/9108.2020.14.14.
[27] Shaikh M, Yadav D. A review of particle swarm optimization (PSO) algorithm. Open Science Framework; 2022, http://dx.doi.org/10.17605/OSF.IO/KQ34H.
[28] Wang D, Tan D, Liu L. Particle swarm optimization algorithm: An overview. Soft Comput 2018;22(2):387–408. http://dx.doi.org/10.1007/s00500-016-2474-6,.
[29] Åström KJ, Hägglund T. Benchmark systems for PID control. IFAC Proc Vol 2000;33(4):165–6. http://dx.doi.org/10.1016/S1474-6670(17)38238-1.
[30] Zerong Y, Zhigang W. Smith predictive PID control in vapor temperature control of ground source heat pump system. GRC Trans 2017;41.
[31] Gnanamurugan M, Senthilkumar A. Smith predictor for control of the Process with Long Dead Time. IJERT 2018;2(6)..
[32] Gahinet P, Chen R, Eryilmaz B. Automated PID controller design. 2013, US8467888B2.
[33] Wang L. PID control system design and autotuning using MATLAB/Simulink. Wiley-IEEE Press; 2020.
[34] Gomes M, Bássora L, Morandin O, Vivaldini K. PID control applied on a line-follower AGV using a RGB camera. In: 2016 IEEE 19th international conference on intelligent transportation systems. 2016, p. 194–8. http://dx.doi.org/10.1109/ITSC.2016.7795553.
[35] Scherlozer A, Orsini M, Patole S. Simulation and numerical analysis and comparative study of a PID controller based on Ziegler–Nichols and auto turning method. In: 12th IEEE international conference on control and automation. 2016.
[36] Skogestad S, Postlethwaite I. Multivariable feedback control: Analysis and design. 2nd ed. Chichester, New York, Brisbane, Toronto, Singapore: John Wiley & Sons, Incorporated; 2001.