Towards a Novel Nonlinear PID Controller Tuned with Particle Swarm Optimization with Improved Performance for First Order Plus Time Delay (FOPTD) Systems

Stefanos Charkoutsis[a] and Mohamed Kara-Mohamed[b]
Faculty of Engineering and Technology, Liverpool John Moores University, Byrom St, Liverpool, U.K.

Keywords: PID, NLPID, Nonlinear, Delay, PSO, MATLAB/Simulink, Uncertainty.

Abstract: The Proportional, Integral, and Derivative (PID) controller is ubiquitous in industry, facing nonlinear systems that it can struggle to compensate. The main limitation of PID is the trade-off between set-point tracking and disturbance rejection that causes control design issues affecting industrial outputs. This paper proposes a novel Nonlinear gains Proportional, Integral, and Derivative (NLPID) controller that shows improved results in the simultaneous set-point tracking and disturbance rejection, using time-varying gains, to control nonlinear systems. The paper also shows the performance of the proposed controller for the case of a First Order Plus Time Delay (FOPTD) system, which heavily exists in industry. The proposed NLPID controller is tuned using the Particle Swarm Optimization (PSO) algorithm. The proposed NLPID controller is simulated in MATLAB/Simulink and compared against PSO tuned PID controller (PSO_PID), Internal Model Control based PID (IMC_PID), and a PID controller with a nonlinear integral function gain (Son_NLPID), for the FOPTD system. This study shows that the proposed NLPID provides a faster response, with minimized overshoot, maintaining excellent disturbance rejection without compromising stability or speed. The study also shows that the proposed NLPID controller is robust against parametric uncertainty.

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 time-domain representation of the conventional PID controller is given as follows:

$$u_{\mathrm{PID}}(t) = k_{\mathrm{p}}\varepsilon(t) + k_{\mathrm{i}}\int_{0}^{t_{\mathrm{f}}} \varepsilon(t)\,\mathrm{d}t + k_{\mathrm{d}}\dot{\varepsilon}(t) \qquad (1)$$

where $k_{\mathrm{p}}, k_{\mathrm{i}},$, and k_{d} are the proportional, integral, and derivative gains, respectively, $\varepsilon(t)$ 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 (Åström and Hägglund, 1995). The linearity and simplicity of the controller make it useful in industrial applications, which is why it has received a lot of attention from researchers and engineers over the years (O'Dwyer, 2009). Industry faces nonlinear systems with process disturbances and modelling uncertainty and it is well known that the PID controller forms a single-degree-of-freedom (1DoF) control structure, that poses a trade-off in performance and robustness (Garpinger et al., 2014; Chen et al., 2019; Bernstein, 2022). Once the PID controller has been tuned for optimal disturbance rejection, an overshoot appears at the set-point response, and once it has been tuned to eliminate the overshoot, a slow disturbance rejection is observed (Garpinger et al., 2014; Chen et al., 2019). However, this imposes the appearance of an overshoot with large input costs in higher than first order, and nonlinear systems, needing complex tuning algorithms, gain scheduling and real-time algorithms have been proposed as a remedy to the issue of using a 1DoF PID Controller (Garpinger et al., 2014; Cetin and Iplikci, 2015; Chen et al., 2019; Shamseldin, 2023).

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 (Son et al., 2021; Shamseldin, 2023; Sivadasan et al., 2023). NLPID controllers have been under active and continuing research with many NLPID controllers that have been

[a] https://orcid.org/0000-0002-2598-9279
[b] https://orcid.org/0000-0001-6423-7275

Charkoutsis, S. and Kara-Mohamed, M.
Towards a Novel Nonlinear PID Controller Tuned with Particle Swarm Optimization with Improved Performance for First Order Plus Time Delay (FOPTD) Systems.
DOI: 10.5220/0012173600003543
In *Proceedings of the 20th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2023) - Volume 2*, pages 25-33
ISBN: 978-989-758-670-5; ISSN: 2184-2809

proposed, are shown to be an efficient alternative control method only for specific nonlinear systems (So, 2019; Jin and Son, 2019; Pathak et al., 2020; Son et al., 2021; Shamseldin, 2023; Sivadasan et al., 2023). Hence, there is need for the development of an NLPID controller for industry that is effective for a larger class of systems (So, 2019; Pathak et al., 2020; Son et al., 2021; Pugazhenthi P et al., 2021; Shamseldin, 2023; Sivadasan et al., 2023). 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. A nonlinear PID controller that utilises only a scaled integral nonlinearity has a stability proof and provides adequate responses to linear and delay type systems (Son et al., 2021). 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 main contributions of this paper is a novel nonlinear function gains PID controller, designed to show improvements to the limitations of the PID controller. The proposed NLPID controller has gains described by a new set of nonlinear functions as a strategy for improving the simultaneous set-point tracking and disturbance rejection. These gains reduce the rise-time with no overshoot for any step set-point function. It provides low input energy and offers a nonlinear PID control scheme that is effective in its performance specifications and is robust against parametric uncertainty.

In this paper a novel nonlinear function gains PID controller is proposed that addresses on 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.

In the sections that follow, the novel nonlinear PID controller proposed in this paper is presented in Section 2. Then, the tuning methodology used across all controllers for benchmarking and the particle swarm optimization algorithm used for the proposed controller is also shown in Section 3. Section 4 shows the results from the benchmarking of the controller against the PSO tuned PID, IMC_PID and the Son_NLPID controllers in a widely used industrial system. Section 5 shows the robustness test under parametric uncertainty of the proposed NLPID controller. Finally, in Section 6 the conclusions and further work are presented to summarise the results found within this research and propose future directions.

2 NOVEL NONLINEAR PID CONTROLLER

The structure of the proposed nonlinear PID controller is similar to that of the parallel linear PID controller. However, in this case the gains are described using nonlinear functions that change the value of the gains depending on the feedback error and feedback error-rate.

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

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 the steady-state error. 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 has such a property is the mollifier function that originate from distribution theory and has not been used in the past within the NLPID control literature. The mollifier takes the mathematical form of:

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

Moreover, in this paper the mollifier is adopted such that the nonlinearity is applied at the transient response region, to maximise 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 nonlinear proportional gain is described by:

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

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. Figure 1 below 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 $r(t)$.

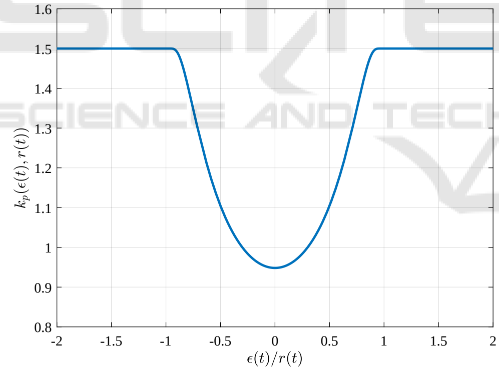

Figure 1: The proposed nonlinear function derivative gain for a constant set-point $r(t) = 1$ and tuning parameters $k_0 = 1.5, a = 1$.

The nonlinear integral gain is described by:

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

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 nonlinearity in order to adapt to the error range, so that the non-linearity is active throughout the transient response.

The integral gain, shown in Figure 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.

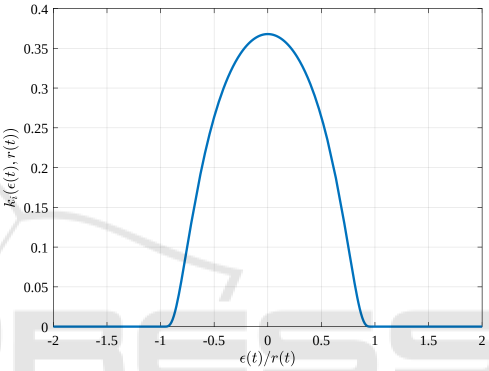

Figure 2: The proposed nonlinear function integral gain for a constant set-point $r(t) = 1$ and tuning parameters $k_1 = 1$.

The nonlinear derivative gain is described by:

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

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 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 system input. It changes the range at which the nonlinearity operates and defines the zero points of the nonlinear gains. 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 Figure 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.

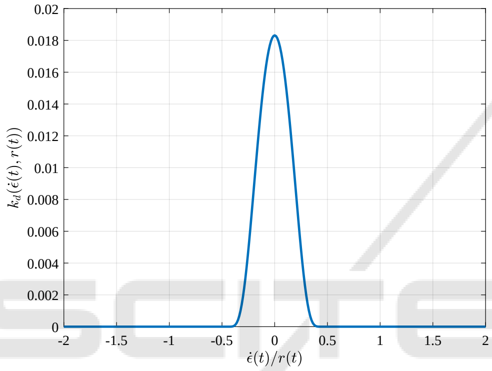

Figure 3: The proposed nonlinear function derivative gain for a constant set-point $r(t) = 1$ and tuning parameters $k_2 = 1, k_3 = 0.5$.

The effect of a changing set-point to the proposed nonlinear proportional gain is shown in Figure 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 $-\varepsilon_{max} \leq \varepsilon \leq \varepsilon_{max}$. 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 in order to maximize the speed during the transient response and maximise robustness during the steady-state. This will improve disturbance rejection and will accommodate any unmodeled uncertainties as it will be shown.

3 TUNING METHODOLOGY

PID control research has the difficulty and unfortunate disadvantage that many control comparisons are

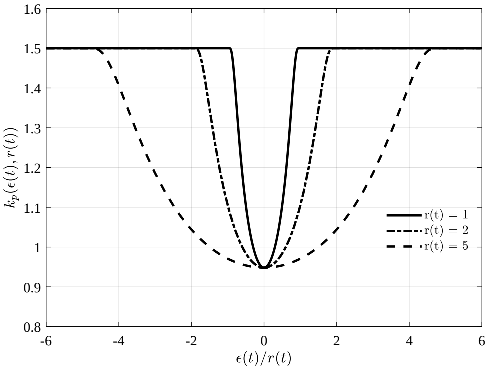

Figure 4: The adaptation of the proposed nonlinear function proportional gain to changing set-point values $r(t) = 1, 2$, and 5.

unfair and improved results can be achieved by spending more effort on tuning (Åström and Hägglund, 1995; Valluru and Singh, 2018; Chen et al., 2019; Joseph et al., 2022). 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, which is the scope of improving upon the PID limitations. The difficulty being when a fair comparison is needed it must be ensured that the right tuning approach is taken and is made transparent.

The Particle Swarm Optimization (PSO) algorithm is a stochastic optimization algorithm that is simple, effective, and able to find optimal values to optimization problems, without the use of derivatives (Shaikh and Yadav, 2022). Due to these features of PSO, it is frequently used with high success in the tuning of nonlinear PID control problems (Shaikh and Yadav, 2022; Joseph et al., 2022). As a result, it is used to tune the proposed NLPID controller and a linear PID controller for the process of benchmarking for the case of an FOPTD system.

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 as:

$$\begin{aligned} \underset{k_0, k_1, k_2, a}{\text{minimize}} \quad & f(t, \varepsilon(t), t_s) = \int_0^{t_f} t|\varepsilon(t)|\,\mathrm{d}t + t_s \\ \text{subject to} \quad & 0 \leq k_0, k_1, k_2 \leq 3, \\ & 0 \leq a, \\ & \quad a \leq 3 \end{aligned} \tag{7}$$

where t_s is the settling time, $\varepsilon(t)$ is the feedback error, t_f is the final time. The optimization problem is defined with the parameter constraints within the specified region to ensure stability and lower the chances of trapping inside local optima. The objective function minimises the feedback error as fast as possible, meeting the requirements of minimising overshoot and rise time. The settling time is added to the objective function to enhance the speed of the response for the fast elimination of the steady-state error, improving the transient response as per the proposed specifications. The parameter k_3 is a filtering parameter determined by the designer without the necessity of a tuning algorithm. For the case of FOPTD systems, PI control can provide excellent response and derivative action is not necessary. As a result, the filtering parameter $k_3 = 0.5$ is designed and justified for the control of FOPTD systems. The PSO algorithm iterates until the final iteration has been reached, using the following steps (Wang et al., 2018; Shaikh and Yadav, 2022):

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 $f(t, \varepsilon(t), t_s) = \int_0^{t_f} t|\varepsilon(t)|\,\mathrm{d}t + t_s$.
5. If values surpass the defined range, re-initialise a random number in range $[k_{min}, k_{max}]$ for k_0, k_1, k_2 and re-initialise a random number in range $[a_{min}, a_{max}]$ for a.
6. The new velocity and position of each particle is computed using a slightly modified version of the PSO algorithm 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 ${}_kV_i^j$ 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 $P_i^j[k_0, k_1, k_2, a]_i^j$. The variable k represents a natural number taking values 1 to 4, iterating between the 4 parameters in the parameter set. If the new position X_{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 tuning (Wang et al., 2018).

This process is a modification of the original 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.

4 SIMULATION EXAMPLE

The proposed NLPID controller is benchmarked against the conventional and state-of-the-art methods of controlling FOPTD system, represented as follows:

$$P(s) = \frac{e^{-0.5s}}{s+1}; \tag{10}$$

The FOPTD process model is to be controlled under 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 then benchmarked against the proposed NLPID controller.

To create a comparison between the different methods, the proposed NLPID controller is benchmarked against conventional and nonlinear control methods. The proposed NLPID controller is benchmarked on the basis of the claim that it is providing an improvement to the fast set-point tracking and disturbance rejection. The system is simulated and compared against conventional and nonlinear controllers in the FOPTD system in both set-point tracking and in disturbance rejection.

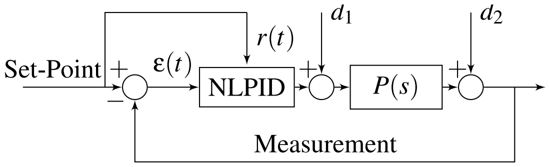

Figure 5: The schematic block diagram of the control system with both the input and output disturbances.

A linear PID controller with derivative filtering has been tuned using the PSO optimisation algorithm. The format of the linear PID controller follows the MATLAB filtered PID controller parallel form as follows:

$$K_{\mathrm{PID}}(s) = k_{\mathrm{p}} + k_{\mathrm{i}}\frac{1}{s} + \frac{k_{\mathrm{d}}N}{1 - \frac{N}{s}} \tag{11}$$

The Son_NLPID controller has a nonlinearity in the integral gain of the PID controller while the proportional and derivative gains are constants. The Son_NLPID controller takes the following form (Son et al., 2021):

$$u(t) = k_p\varepsilon(t) + k_i \int e^{\left[-\frac{\varepsilon(t)}{2\Delta r(t)^2}\right]} \varepsilon(t)\,\mathrm{d}t + k_d \frac{d\varepsilon(t)}{dt} \tag{12}$$

where $\Delta r(t)$ is the set-point change and follows the condition that $\Delta r(t) \neq 0$.

The PSO algorithm determines the NLPID and PSO_PID gains shown in Table 1. The tuning for the Son_NLPID and IMC_PID controllers are from literature for the equivalent FOPTD system shown in Table 1 (Son et al., 2021).

Table 1: Tuned values of the control parameters for each controller.

	Tuning Parameters
Proposed	$k_0 = 1.54, k_1 = 2.95, k_2 = 2.32, k_3 = 0.50, a = 1.13$
IMC_PID	$k_\mathrm{p} = 1.11, k_\mathrm{i} = 0.88, k_\mathrm{d} = 0.11$
PSO_PID	$k_\mathrm{p} = 0.77, k_\mathrm{i} = 1.13, k_\mathrm{d} = 0.74, N = 0.44$
Son_NLPID	$k_\mathrm{p} = 1.98, k_\mathrm{i} = 1.94, k_\mathrm{d} = 0.36$

Figure 6 illustrates that the proposed NLPID controller outperforms the IMC_PID and PSO_PID controllers, showing faster rise and settling time. Although the Son_NLPID controller indicates faster rise time, it also presents an overshoot of approximately 4.8%, while the proposed NLPID controller shows an overshoot of 0.35%.

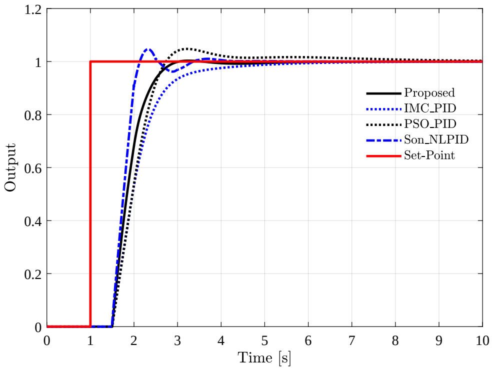

Figure 6: Benchmarking controllers to a step set-point function response.

Figure 7 shows the system input of each benchmarked control system. The proposed NLPID controller indicates the cheapest control strategy with practical and effective system input that does not generate large derivative and proportional kick signals. It is clear from the figure that the IMC_PID and Son_NLPID controllers show large derivative and proportional kicks, making them energy expensive control strategies. This also indicates the internal stability of the proposed NLPID controller under step set-point tracking. The derivative kicks are presented in the input during benchmarking and the authors of the paper have been contacted to inquire about how they dealt with the derivative kicks, but no response has been received. In order to reproduce the most accurate form of the controllers, the derivative kicks have not been filtered.

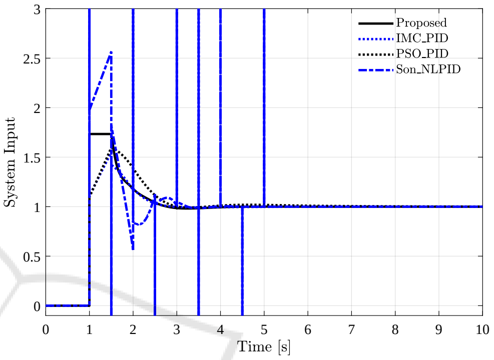

Figure 7: System input due to step set-point function response.

Table 2 shows the control criteria of a step set-point response of all the controllers. The L_2 energy has also been computed to show the system input energy generated by each controller. The L_2 computation has the following form:

$$L_2(u(t)) = \sqrt{\sum_{t_0}^{t_f}(u(t)^2)} \tag{13}$$

It is clearly shown that the proposed NLPID controller produces the fastest response with minimal overshoot and lowest system input energy.

Table 2: The benchmark results of the set-point response control criteria.

	%Os	$t_s(s)$	$t_r(s)$	L_2
Proposed	0.35	2.76	0.82	43.94
IMC_PID	0	4.26	1.16	1.55×10^{13}
PSO_PID	4.48	3.98	0.88	44.34
Son_NLPID	4.8	3.15	0.45	5.11×10^{13}

For the benchmark testing, the disturbance rejection has been taken to be 10% of the set-point value, applied to the system input at 6 seconds time mark and a 10% disturbance has also been taken in the output at 12 seconds time mark, which represents a sensor bias and deviation from the true value as shown by Figure 5. The proposed NLPID controller has the fastest output disturbance rejection response, while the Son_NLPID controller has the fastest input disturbance rejection response. The proposed NLPID controller and IMC_PID show a faster response with no overshoot when compared to the PSO_PID controller. The proposed NLPID controller outperforms the IMC_PID and PSO_PID in both input and output disturbance rejections.

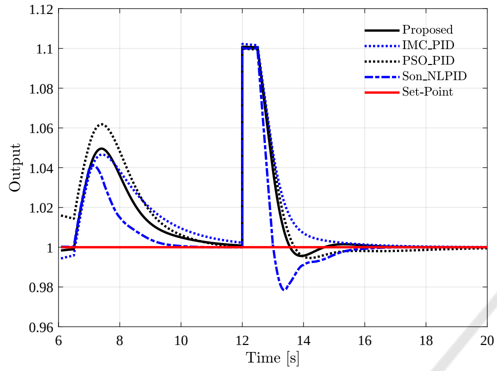

Figure 8: Output response of both input and output disturbances.

The input and output disturbance rejection of control systems can at times render a system internally unstable. As a result, the system input energy has been shown in order to show internal stability and practicality of the controllers. It is clearly shown from Figure 9 that the proposed NLPID controller and all benchmarked controllers are internally stable with bounded system input signals. However, the Son_NLPID and IMC_PID controllers indicated derivative kick signals due to the step function disturbances, which also have a negative effect on system input energy.

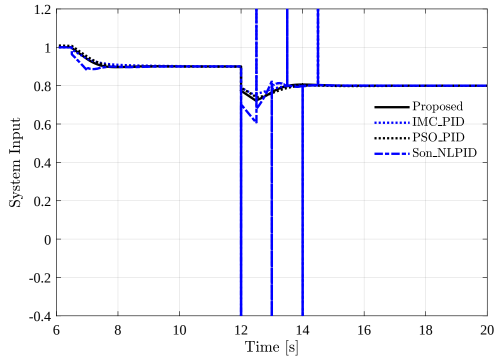

Figure 9: System input due to the disturbance rejection response in both input and output disturbances.

Figure 10 shows the time variation of the nonlinear function gains. The initial value of the set-point is zero, as a result, the functions take the steady-state value and as the set-point is produced the functions then increase and settle again as the system approaches steady-state value. The nonlinear proportional gain produces a large signal with a fast response initially, which then reduces rapidly near steady state to reduce the overshoot. The nonlinear integral gain increases as steady state error approaches to eliminate any steady state error. It can also be seen that when the step function is applied in both set-point and disturbances, the derivative gain rapidly reduces to zero, eliminating any unwanted derivative kick effects.

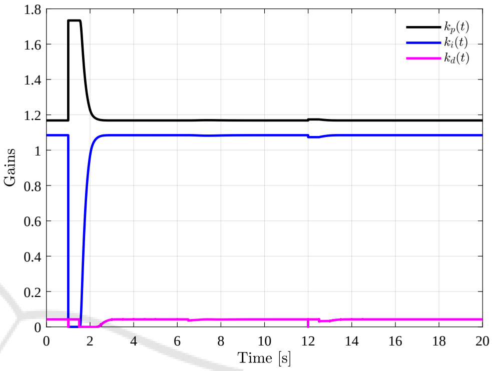

Figure 10: The time variation of the proposed nonlinear gains to the step response and disturbance rejection.

The effectiveness of the proposed NLPID controller comes from the properties of the proposed nonlinear gain functions. The nonlinear proportional gain produces a large signal at large errors, quickly reducing the feedback error, then rapidly decreasing to avoid overcompensation. The nonlinear integral gain is negligible at large errors and increases rapidly near steady-state for the purpose of eliminating left-over steady-state errors. Finally, the derivative gain is produced near steady-state in order to minimise overshoot and oscillatory responses. These properties of the proposed NLPID controller provide the fast transient response with minimal overshoot, while maintaining fast disturbance rejection.

5 ROBUSTNESS OF PROPOSED NLPID CONTROLLER AGAINST PARAMETRIC 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 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 parametric uncertainty of the nominal FOPTD plant is modelled using Eq. (14) in the following transfer function format:

$$\overline{P}(s) = \frac{\overline{k}e^{\overline{\tau}s}}{\overline{t}_p s + 1} \tag{14}$$

where $0.9 \leq \overline{k} \leq 1.1$, $0.9 \leq \overline{t}_p \leq 1.1$, and $0.45 \leq \overline{\tau} \leq 0.55$, which models a $\pm 10\%$ variable change.

Figure 11 shows how the proposed NLPID controller responds to $\pm 10\%$ 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 and lag parameters are underestimated, the response shows a small overshoot and a larger settling time. The system shows no effect on stability, providing evidence for robust performance and robust stability of the proposed NLPID to gain, lag, and delay variations.

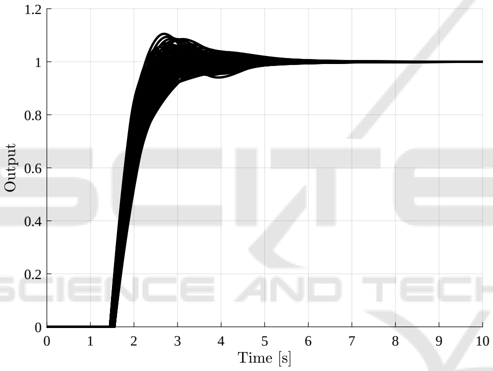

Figure 11: Output response to step set-point function under parametric uncertainty.

Figure 12 shows the system input of the proposed NLPID controller under parametric uncertainty. This indicates that the proposed NLPID controller generates minimal to none derivative and proportional kicks, making it an effective and practical controller. In addition it indicates that the proposed controller has no hidden internal instabilities produced due to parametric uncertainty, hence indicating that the proposed NLPID controller is an effective and robust control method.

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, showing robust stability. The uncertainty tests indicate that stability is maintained across different types of parameter variations with some changes in performance of approximately 3 seconds slower settling time, when compared to the nominal plant, and approximately 10% maximum overshoot.

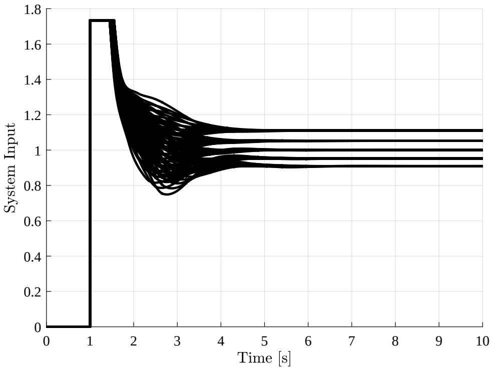

Figure 12: System input due to parametric uncertainty.

6 CONCLUSIONS

The tuning of the proposed NLPID controller is conducted using the PSO algorithm to find the optimal values. This is then simulated and compared against IMC_PID, PSO_PID, and Son_NLPID for a FOPTD system.

The proposed NLPID controller outperforms the conventional control systems in simultaneous transient response and disturbance rejection, which is a critical response specification in control systems. The proposed NLPID controller also outperforms the Son_NLPID controller in set-point tracking response and output disturbance rejection. However, the Son_NLPID controller outperforms the proposed NLPID controller in input disturbance rejection. The proposed controller also shows robust stability and robust performance to parametric uncertainty with no large variations.

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.

REFERENCES

Åström, K. J. and Hägglund, T. (1995). *PID Controllers: Theory, Design and Tuning*. Instrument Society of America, 2nd edition.

Bernstein, D. S. (2022). Facing Future Challenges in Feedback Control of Aerospace Systems Through Scientific Experimentation. *Journal of Guidance, Control, and Dynamics*, 45(12):2202–2210.

Cetin, M. and Iplikci, S. (2015). A novel auto-tuning PID control mechanism for nonlinear systems. *ISA Transactions*, 58:292–308.

Chen, J., Fang, S., and Ishii, H. (2019). Fundamental limitations and intrinsic limits of feedback: An overview in an information age. *Annual Reviews in Control*, 47:155–177.

Garpinger, O., Hägglund, T., and Åström, K. J. (2014). Performance and robustness trade-offs in PID control. *Journal of Process Control*, 24(5):568–577.

Jin, G.-G. and Son, Y.-D. (2019). Design of a Nonlinear PID Controller and Tuning Rules for First-Order Plus Time Delay Models. *STUD INFORM CONTROL*, 28(2).

Joseph, S. B., Dada, E. G., Abidemi, A., Oyewola, D. O., and Khammas, B. M. (2022). Metaheuristic algorithms for PID controller parameters tuning: Review, approaches and open problems. *Heliyon*, 8(5):e09399.

O'Dwyer, A. (2009). *Handbook of PI and PID Controller Tuning Rules*. Imperial College Press ; Distributed by World Scientific Pub, London : Hackensack, NJ, 3rd ed edition.

Pathak, D., Bhati, S., and Gaur, P. (2020). Fractional-order nonlinear PID controller based maximum power extraction method for a direct-driven wind energy system. *International Transactions on Electrical Energy Systems*, 30(12):e12641.

Pugazhenthi P, N., Selvaperumal, S., and Vijayakumar, K. (2021). Nonlinear PID controller parameter optimization using modified hybrid artificial bee colony algorithm for continuous stirred tank reactor. *Polska Akademia Nauk. Bulletin of the Polish Academy of Sciences: Technical Sciences*, 69(3).

Shaikh, M. and Yadav, D. (2022). A Review of Particle Swarm Optimization (PSO) Algorithm.

Shamseldin, M. A. (2023). Design of Auto-Tuning Nonlinear PID Tracking Speed Control for Electric Vehicle with Uncertainty Consideration. *World Electric Vehicle Journal*, 14(4):78.

Sivadasan, J., Iruthayarajan, M. W., Stonier, A. A., and Raymon, A. (2023). Design of Cross-Coupled Nonlinear PID Controller Using Single-Objective Evolutionary Algorithms. *Mathematical Problems in Engineering*, 2023:e7820047.

So, G.-B. (2019). EA-Based Design of a Nonlinear PID Controller Using an Error Scaling Technique. *SIC*, 28(3):279–288.

Son, Y.-D., Bin, S.-D., and Jin, G.-G. (2021). Stability Analysis of a Nonlinear PID Controller. *Int. J. Control Autom. Syst.*, 19(10):3400–3408.

Valluru, S. K. and Singh, M. (2018). Performance investigations of APSO tuned linear and nonlinear PID controllers for a nonlinear dynamical system. *Journal of Electrical Systems and Information Technology*, 5(3):442–452.

Wang, D., Tan, D., and Liu, L. (2018). Particle swarm optimization algorithm: An overview. *Soft Comput*, 22(2):387–408.