

# Design of a Nonlinear PID Controller and Tuning Rules for First-Order Plus Time Delay Models

Gang-Gyoo JIN<sup>1</sup>, Yung-Deug SON<sup>2\*</sup>

<sup>1</sup> School of Electrical Engineering and Computing, Adama Science and Technology University, P.O. Box 1888, Adama, Ethiopia  
gjin30@gmail.com

<sup>2</sup> Department of Mechanical Facility Control Engineering, Korea University of Technology and Education, 1600 Chungjeol-ro, Dongnam-gu, Cheonan, Chungnam, 31253, Korea  
ydson@koreatech.ac.kr (\*Corresponding author)

**Abstract:** This paper introduces a simple but effective nonlinear proportional-integral-derivative (PID) controller and three model-based tuning rules for first-order plus time delay (FOPTD) models. The proposed controller is based on a conventional PID control architecture, wherein a nonlinear gain is coupled in series with the integral action to scale the error. The optimal parameter sets of the proposed PID controller for step setpoint tracking are obtained based on the FOPTD model, dimensional analysis and a genetic algorithm. As for gauging the performance of the controller, three performance indices (ISE, IAE and ITAE) are adopted. Then, tuning rules are derived using the tuned parameter sets, potential rule models and the least squares method. The simulation results carried out on three processes demonstrate that the proposed method exhibits better performance than the conventional linear PID controllers.

**Keywords:** Nonlinear PID, FOPTD model, Tuning rule, Dimensional analysis, Genetic algorithm, Least squares method.

## 1. Introduction

As is well known, proportional-integral-derivative (PID) controllers have been broadly used in industrial process loops for over 70 years. The popularity of PID controllers is due partly to their simplicity in structure and easiness in implementation. However, a PID controller has to balance three-gains for the overall feedback system to meet with the design specification. If it is not properly tuned, it will degrade the control performance and may lead to damage the whole system.

So far, PID controllers have been tuned in a large number of ways (O'Dwyer, A., 2006) including the two tunings of Ziegler and Nichols (1942). In the case of open-loop tuning based on the S-shaped reaction curve of processes, it will not be possible to obtain the exact parameters from the curve in the presence of noises, disturbances as well as the process parameter variations. In the case of closed-loop tuning based on the point of marginal stability, it is difficult to obtain sustained oscillations in the process variable in real processes. An initially setting PID controller can be expected to perform satisfactorily as long as the process is running in a range close to a nominal operating point; however, it may be insufficient to maintain the desired control performance under varying operating points or process uncertainties. Therefore, PID controllers need to be retuned on-site as operating conditions change.

In this circumstance, various approaches have been suggested in the literature by means of the concepts of self-tuning, adaptive or intelligent control such as relay feedback (Åström, K. J. & Hägglund, T., 1984), self-tuning using pattern recognition (Girirajkumar et al., 2010), real time adaptive tuning (Myron, T. J., 1986) and intelligent methods combining with fuzzy logic (Khan, A. A. & Rapal, N., 2006), neural networks (Kumar, R., Srivastava, S. & Gupta, J.R.P., 2016) and evolutionary algorithms (Civelek et al., 2016). In recent years another approach towards using nonlinear elements within the framework of the conventional PID controller has been found. Most of studies based on this approach can be roughly categorized into two types: nonlinear adjustment of controller gains and nonlinear scaling of the error. The former is the NPID controller, whose gains are gradually changed based on error and/or error rate. Relating this type of studies, Isayed & Hawwa (2007) presented a nonlinear PID control scheme for hard disk actuating systems, where controller gains of hyperbolic functions depend on the system error and/or its derivative. Korkmaz, Aydogdu & Dogan (2012) presented a nonlinear PID controller whose gains are adjusted by the error function. In a similar fashion, Zhang & Hu (2012) proposed a nonlinear PID controller whose gains are adjusted by three nonlinear functions for generator excitation system control. In (So, G. & Jin, G., 2018), a nonlinear PID controller for

temperature control of a continuous stirred-tank reactor process with tuning gains whose parameters are characterized by using a set of Tagaki-Sugeno fuzzy rules and are continuously changed while the process is in operation was presented.

The latter is the nonlinear PID controller, where a nonlinear gain in cascade with a linear PID controller produces the scaled error. As concerns this type of studies, Seraji (1997, 1998) introduced a class of nonlinear PID controllers which consist of a nonlinear gain in cascade with a linear fixed-gain PID controller. Three nonlinear gains such as the sigmoidal function, the hyperbolic function, and the piecewise-linear function were proposed and the stability of the closed-loop systems was investigated using the Popov stability criterion. In (Han, J., 2009), a nonlinear PID controller was designed, where the control input is decided by the nonlinearly weighted sum of the three PID controller gains. In this form, the nonlinear weights are designed by using a kind of special nonlinear functions named fal, a nonlinear combination of the error, its integration as well as the error rate. In a similar fashion to the Han's work, Kler, Rana & Kumar (2018) proposed a PID controller wherein its integral term gain is varied according to the tracking error to effectively attain Maximum Power Point Tracking (MPPT) in Photovoltaic (PV) systems. Zheng, Su & Mercorelli (2019) presented a simple nonlinear PD control design for positioning of servomechanisms with saturating actuator. So et al. (2018) presented two nonlinear PID controllers with a nonlinear function implemented as a Takagi-Sugeno fuzzy model for the glycol temperature control of a liquefied natural gas (LNG) regasification system.

Despite that these methods achieve satisfactory results in different control environments, there is also a drawback regarding them. Every method is process dependent and, in most cases, has its own limitation due to the increased number of tuning parameters when compared to conventional PID controllers. This paper proposes a simple nonlinear PID controller which incorporates a nonlinear gain in cascade with the integral action of a conventional PID control architecture. It deals with the problem of obtaining new tuning rules for FOPTD models. The parameters of the controller that provides optimum tracking performance to the step change of the setpoint are derived on the basis of a process model and a genetic algorithm in terms of minimizing three performance indices. Three tuning rules are derived from a set of

tuned parameters, tuning rule models and the least squares method. To verify the usefulness of the proposed method, a set of simulation works regarding the three processes is carried out and the responses of the proposed controller are compared with those of the existing linear PID controller.

This paper is organized as follows: Section 2 gives a brief overview about the FOPDT model and two existing tuning rules. Section 3 describes the proposed nonlinear PID controller. Section 4 introduces three new tuning rules. Section 5 illustrates the simulation results and discussion. Finally, some concluding remarks and future work are presented in the last section.

## 2. FOPTD Model and PID Controller

### 2.1 First-order Plus Time Delay Model

Generally, processes with highly damped dynamics are represented as a first-order plus delay time (FOPDT) model. The FOPDT model is an approximation of many complex processes that is able to describe a broad range of process behaviours. It is widely used in controller tuning. Well-known tuning methods such as Ziegler-Nichols tuning, Cohen-Coon tuning, Tyreus-Luyben tuning, Chien-Hrones-Reswick tuning and so on are based on the parameters of the FOPDT model (O'Dwyer, A., 2006). Mathematically, the FOPDT model can be stated by the time domain expression:

$$\tau \frac{dy(t)}{dt} + y(t) = Ku(t-L) \quad , \quad (1)$$

where  $K$  denotes the process gain,  $\tau$  the process time constant, and  $L$  the process time delay.

### 2.2 Conventional PID Controller and Tuning Rules

In general, a parallel form of conventional PID controllers is given as follows:

$$u(t) = K_p [e(t) + \frac{1}{T_i} \int e(t)dt + T_d \frac{de(t)}{dt}] \quad , \quad (2)$$

where  $u$  is the control input,  $e$  is the error between the setpoint (SP) and the process variable (PV), and  $K_p$ ,  $T_i$  and  $T_d$  denote the proportional gain, the integral time and the derivative time, respectively.

For a given process, the tuning is represented by a proper selection of  $K_p$ ,  $T_i$  and  $T_d$  in such a way that the PID controller provides an acceptable performance in a certain sense and ensures

design requirements. Ziegler and Nichols (1942) proposed the two heuristic methods in the 1940s. They are based on the response obtained from experiments of a process. One of them, namely the open-loop method applies to a process whose unit-step response resembles an S-shaped reaction curve with no overshoot. The parameters of the PID controller are calculated using the parameter estimates ( $K$ ,  $\tau$ , and  $L$ ) obtained from the reaction curve and the Ziegler-Nichols tuning rules. If a process model is available, the PID controller can be directly tuned from the model without doing any experiment.

As another useful method, Tavakoli proposed an optimal method for tuning the PID controller for FOPDT models. Using the dimensional analysis and numerical optimization techniques, three formulas for integral of square error (ISE), integral of absolute error (IAE), and integral of time-weighted absolute error (ITAE) performance indices are given in Table 1 (Tavakoli, S. & Tavakoli M., 2003):

**Table 1.** Tuning formulas for step setpoint tracking proposed by Tavakoli

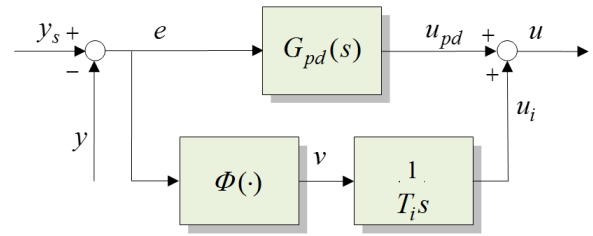
Performance Index	Dimensionless parameters		
	$KK_p$	$T_i/\tau$	$T_d/\tau$
ISE	$0.3\frac{L}{\tau} + 0.75$	$\frac{2.4}{L/\tau + 0.4}$	$\frac{1}{90\frac{L}{\tau}}$
	$\frac{L}{\tau} + 0.05$		
IAE	$\frac{1}{L/\tau + 0.2}$	$\frac{0.3\frac{L}{\tau} + 1.2}{L/\tau + 0.08}$	$\frac{1}{90\frac{L}{\tau}}$
ITAE	$\frac{0.8}{L/\tau + 0.1}$	$0.3 + \frac{1}{L/\tau}$	$\frac{0.06}{L/\tau + 0.04}$

### 3. Proposed Nonlinear PID Controller

#### 3.1 Structure of the Nonlinear PID Controller

Once a large change in setpoint occurs or a large disturbance applies, the proportional action and the derivative action operate in proportion to the error and the error rate, respectively. The integral action continuously accumulates the current and the past of the error to get rid of steady-state error. Excessive integral action introduces other problems such as overshoot, integral windup during transient response and reduction of stability

margin. Therefore, when the error is considerable, the integral of the error is preferred to take low values to improve the transient response and when the error is irrelevant, the integral of the error is preferred to take high values to remove offset quickly. In this way, the integral action needs to be continuously adjusted so as to meet both conditions. Consequently, a nonlinear PID controller whose architecture is similar to the conventional PID controller is proposed, but only the integral action contains a nonlinear function while the other actions are linear. Hence, the error input to the integral action is scaled by a nonlinear function in the form of the product of the error and a nonlinear gain. The block diagram of the proposed nonlinear PID (NPID) controller is shown in Figure 1. Hereafter, this is called the NPID controller.



**Figure 1.** The proposed NPID controller topology

In Figure 1,  $y_s$  and  $y$  denote the SP and PV, respectively.  $u_{pd}$  and  $u_i$  denote the outputs of the PD action and the I action, respectively. The equation of the NPID controller is of the following form:

$$v(t) = \Phi(e) = k(e)e(t), \quad (3)$$

$$u(t) = K_p \left[ e(t) + \frac{1}{T_i} \int v(t) dt + T_d \frac{de(t)}{dt} \right] \quad (4)$$

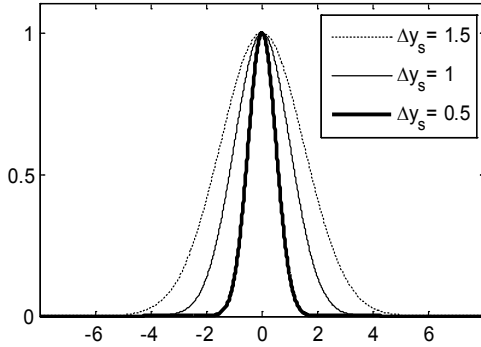
where  $v(t)$  is a scaled error,  $\Phi(e)$  a nonlinear function which provides a nonlinearly-scaled error according to  $e$ , and  $k(e)$  a nonlinear gain.  $k(e)$  is described as:

$$k(e) = \exp\left(-\frac{e^2}{2\Delta y_s^2}\right), \quad (5)$$

where  $\Delta y_s (\neq 0)$  is the difference between the current SP,  $y_s(n)$ , and the previous SP,  $y_s(n-1)$  and  $n$  denotes the discrete instant of time.

Figure 2 shows typical shapes of  $k(e)$  versus  $e$  according to the change of  $\Delta y_s$ . Note that  $k(e)$  is an even function. Given the input  $e$ , the shape of the nonlinear gain  $k(e)$  depends on  $e$  and  $\Delta y_s$ . The lower the magnitude of  $\Delta y_s$ , the narrower the width becomes, and the larger the magnitude of  $\Delta y_s$ , the wider the width. Moreover, it can also

be noticed that  $k(e)$  converges to the lower limit 0 when  $e$  goes to infinity, and converges to the upper limit 1 when  $e$  approaches 0. It is clearly seen from Figure 2 that, for large  $e$  which is caused by the SP change or/and disturbances,  $k(e)$  scales down the error in order to avoid excessive control effort. This favourable property of  $k(e)$  produces good transient responses.



**Figure 2.** Shapes of  $k(e)$  versus  $e$  according to the change of  $y_s$

### 3.2 Dimensional analysis

As seen in Equation (1), the FOPTD model has three parameters. The unit of  $\tau$  and  $L$  is time and the unit of  $K$  depends on the input and output of the process. It is quite difficult to derive any optimum controller setting in terms of these three parameters. Therefore, the dimensionless approach which can make the optimum setting problem by simply reducing some of the parameters used and which can analyse the process behaviour without physical dimension is applied.

Defining dimensionless variable  $t' = t/\tau$  and letting  $y(\tau t') = \bar{y}(t')$  in order to simplify notation, then the derivative term  $dy(t)/dt$  can be written in the dimensionless form by

$$\frac{dy(t)}{dt} = \frac{dy(\tau t')}{dt} = \frac{1}{\tau} \frac{d\bar{y}(t')}{dt'} \quad (6)$$

Also a delay of  $L$  can be expressed by a delay of  $L/\tau$  in the new time variable  $t'$  by defining  $u(\tau t') = \bar{u}(t')$ :

$$u(t-L) = u\left[\tau\left(t' - \frac{L}{\tau}\right)\right] = \bar{u}\left(t' - \frac{L}{\tau}\right) \quad (7)$$

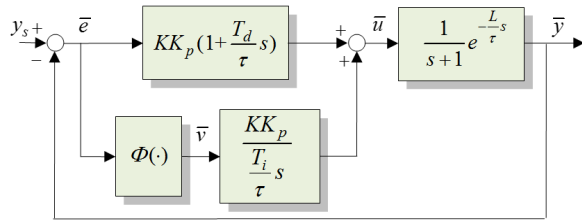
Then, Equation (1) can be written in the dimensionless form

$$\frac{d\bar{y}(t')}{dt'} + \bar{y}(t') = K\bar{u}\left(t' - \frac{L}{\tau}\right) \quad (8)$$

Similarly, by defining variables and applying the dimensional analysis to the controller equation, Equation (4) can be also rewritten in terms of  $t'$  as

$$\bar{u}(t') = K_p \left[ \bar{e}(t') + \frac{\tau}{T_i} \int \bar{v}(t') dt' + \frac{T_d}{\tau} \frac{d\bar{e}(t')}{dt'} \right] \quad (9)$$

Taking the Laplace transform for Equations (8) and (9), it yields the overall block diagram with unity feedback, as depicted in Figure 3.



**Figure 3.** The dimensionless NPID control system

As it can be seen in Figure 3, the controlled object is described only as a function of the dimensionless parameter  $L/\tau$  by combining the process gain with the proportional gain of the controller. A major advantage of the proposed NPID controller is that it is characterized by the three parameters  $KK_p$ ,  $T_i/\tau$ , and  $T_d/\tau$  as the conventional PID controller.

### 4. Tuning of the NPID Controller

The NPID controller designed in the previous section can be applied to various types of systems. However, as mentioned earlier, many industrial processes can be modelled as FOPTD, so this section deals with the tuning problem of the NPID controller for the FOPTD model.

The NPID controller parameters are determined using the dimensionless model shown in Figure 3 and a genetic algorithm (GA). This method is based on minimizing three performance indices such as ISE, IAE, and ITAE. Their formulas in the dimensionless form are as follows:

$$ISE = \int_0^{\infty} \bar{e}^2(t') dt' \quad (10)$$

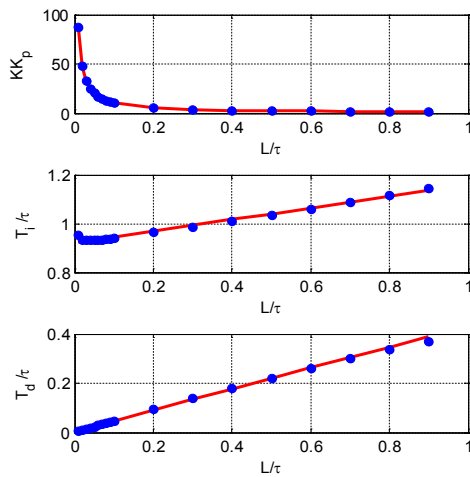
$$IAE = \int_0^{\infty} |\bar{e}(t')| dt' \quad (11)$$

$$ITAE = \int_0^{\infty} t' |\bar{e}(t')| dt' \quad (12)$$

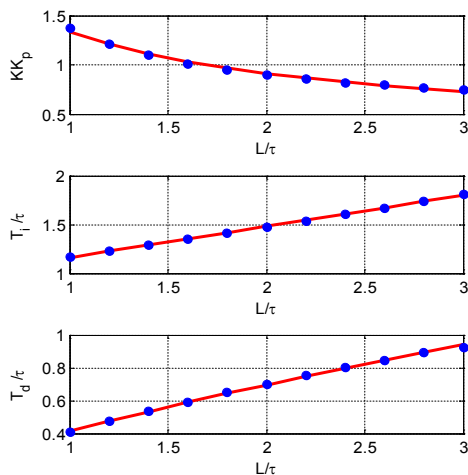
ISE is often used for optimal control because it is easy to analyse. However, it penalizes the error larger than one larger, and conversely smaller than one smaller, which makes it insensitive to the change of parameters near the optimal solution. IAE exhibits better sensitivity than ISE by using

the absolute magnitude of the error, while ITAE gives a time penalty for the long transient, thus having greater discrimination than IAE or ISE.

In this study, only cases in which  $L/\tau$  of the controlled objects is in the range between 0.01 to 3 are considered as in the previous studies (O'Dwyer, A., 2006). Therefore, the proposed settings are valid only for processes that can be modelled as FOPTD and for which of  $L/\tau$  is between 0.01 and 3. For each performance index, the program runs 10 times with different random seeds to obtain the solution sets ( $KK_p$ ,  $T_i/\tau$ , and  $T_d/\tau$ ) which minimize the performance index to the setpoint change while changing the value of  $L/\tau$  from 0.01 to 3. Then, the results are averaged. The calculated parameters of the NPID controller for the changes of  $L/\tau$  for each index are shown as data at different points in Figures 4-6.

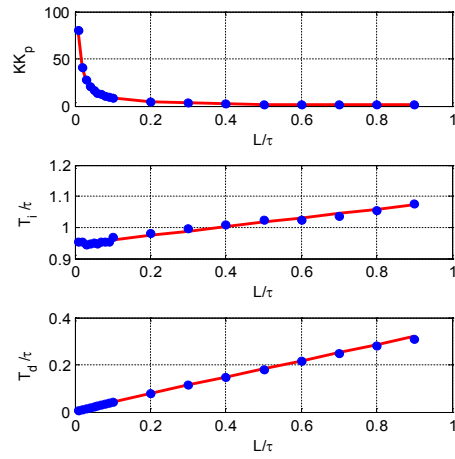


(a)  $0.01 \leq L/\tau \leq 0.9$

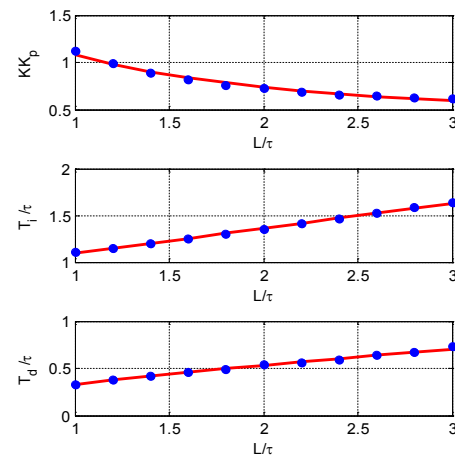


(b)  $1 \leq L/\tau \leq 3$

**Figure 4.** The tuned parameters for the changes of  $L/\tau$  for ISE

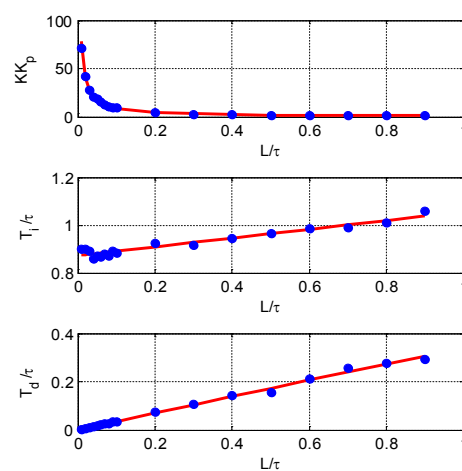


$0.01 \leq L/\tau \leq 0.9$

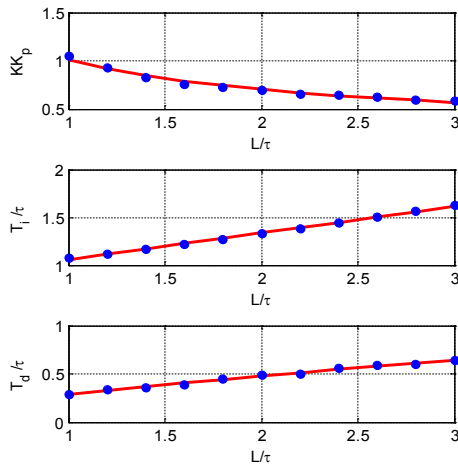


(b)  $1 \leq L/\tau \leq 3$

**Figure 5.** The tuned parameters for the changes of  $L/\tau$  for IAE



(a)  $0.01 \leq L/\tau \leq 0.9$



(b)  $1 \leq L/\tau \leq 3$

**Figure 6.** The tuned parameters for the changes of  $L/\tau$  for ITAE

The following tuning rule models to do curve fitting for both linear regression and nonlinear regression are considered:

$$KK_p = a_1 \left(\frac{L}{\tau}\right)^{b_1}, \tag{13a}$$

$$\frac{T_i}{\tau} = a_2 + b_2 \left(\frac{L}{\tau}\right), \tag{13b}$$

$$\frac{T_d}{\tau} = a_3 \left(\frac{L}{\tau}\right)^{b_3}, \tag{13c}$$

where  $a_1, b_1, a_2, b_2, a_3, b_3$  are unknown parameters. Formula (13) can be thought of a form of existing rules (O'Dwyer, A., 2006). Given the data sets and each model, a specialized software can be used for curve fits. Nonlinear regression models of formulas (13a) and (13c) are converted to a linear regression model by taking the logarithm of both sides. The tuning rules obtained for each performance index are listed in Tables 2-3 and their curve fittings are shown in Figures 4-6. The curve fitting plots shown in (b) are the continuation of the plots shown in (a).

**Table 2.** ISE, IAE and ITAE tuning rules for step setpoint tracking ( $0.01 \leq L/\tau \leq 0.9$ )

Performance Index	Dimensionless parameters		
	$KK_p$	$T_i/\tau$	$T_d/\tau$
ISE	$1.2886\left(\frac{L}{\tau}\right)^{-0.9182}$	$0.9217 + 0.2375\frac{L}{\tau}$	$0.4302\left(\frac{L}{\tau}\right)^{0.9645}$
IAE	$1.0350\left(\frac{L}{\tau}\right)^{-0.9327}$	$0.9465 + 0.1398\frac{L}{\tau}$	$0.3527\left(\frac{L}{\tau}\right)^{0.9406}$
ITAE	$1.0019\left(\frac{L}{\tau}\right)^{-0.9457}$	$0.8723 + 0.1848\frac{L}{\tau}$	$0.3401\left(\frac{L}{\tau}\right)^{0.9799}$

**Table 3.** ISE, IAE and ITAE tuning rules for step setpoint tracking ( $1 \leq L/\tau \leq 3$ )

Performance Index	Dimensionless parameters		
	$KK_p$	$T_i/\tau$	$T_d/\tau$
ISE	$1.3362\left(\frac{L}{\tau}\right)^{-0.5488}$	$0.8419 + 0.3190\frac{L}{\tau}$	$0.4137\left(\frac{L}{\tau}\right)^{0.75}$
IAE	$1.0822\left(\frac{L}{\tau}\right)^{-0.5495}$	$0.8237 + 0.2692\frac{L}{\tau}$	$0.3331\left(\frac{L}{\tau}\right)^{0.6831}$
ITAE	$1.0093\left(\frac{L}{\tau}\right)^{-0.5198}$	$0.7813 + 0.2781\frac{L}{\tau}$	$0.2878\left(\frac{L}{\tau}\right)^{0.7317}$

## 5. Simulation Results

This section illustrates the effectiveness of the proposed NPID controller and three tuning rules through a set of simulation works. The responses of the NPID controller are compared with those of the linear PID controller tuned by the Ziegler-Nichols open-loop method (hereafter referred to as LPID-ZN) and the Tavakoli's method (hereafter referred to as LPID-ISE, LPID-IAE, and LPID-ITAE).

### 5.1 Process I

Firstly, the process is given by the following second-order model

$$G_p(s) = \frac{e^{-2s}}{(1+s)(1+10s)}. \tag{14}$$

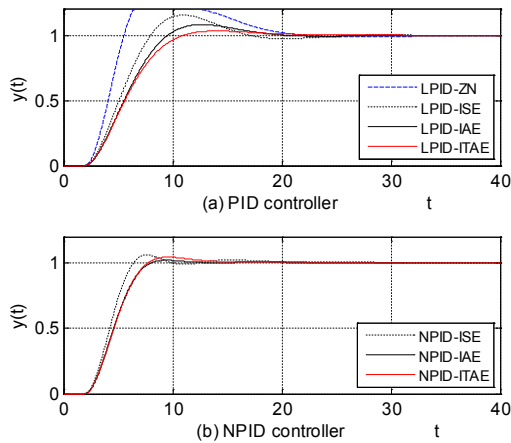
The FOPTD model parameters of the given process were determined by using the GA-based model reduction technique. An input is simultaneously applied to both the process and the model which are connected in parallel.

The adjustable parameters  $K, \tau,$  and  $L$  of the model are changed by a genetic algorithm that minimizes a performance index of the difference between the process output and the model output (Lee, H. & Jin, G., 1998). This technique gives the values  $K= 1, \tau = 10.002,$  and  $L= 3.06$ . It can be seen that in this case  $L/\tau \approx 0.31 (< 1)$ . Table 4 lists the parameters of the PID controller and the NPID controller obtained from the estimated parameters.

In order to evaluate the SP tracking performance of the NPID controller tuned by the proposed tuning rules, the responses for a unit step SP change were obtained and the results were compared with those of the two methods. Figure 7 shows the closed-loop responses resulted from applying the three methods to the model (14). Figure 7(a) and 7(b) are the responses of the linear PID controller and the proposed NPID controller, respectively.

**Table 4.** PID and NPID controller tunings for Process I

Controller Type	Tuning Method	Parameters		
		$K_p$	$T_i$	$T_d$
PID	LPID-ZN	3.9224	6.1200	1.5300
	LPID-ISE	2.3650	10.4032	0.1111
	LPID-IAE	1.9765	10.2422	0.1111
	LPID-ITAE	1.9707	10.9200	0.5307
NPID	NPID-ISE	3.8230	9.9456	1.3729
	NPID-IAE	3.1238	9.8947	1.1579
	NPID-ITAE	3.0709	9.2902	1.0658



**Figure 7.** Responses for the unit step input of Process I

As can be seen, Figure 7 shows that the proposed method produces good transient responses in terms of both swiftness and closeness of the response. The Z-N method has the poorest response, where it shows a large overshoot. For a quantitative comparison of the responses, the performance indices such as rise time  $t_r = t_{95} - t_s$ , overshoot  $M_p$ , 2% settling time  $t_s$ , and integral of absolute error (IAE) were obtained and summarized in Table 5.

**Table 5.** Quantitative comparison of the SP tracking responses for Process I

Tuning Method	SP tracking performance			
	$t_r$	$M_p$	$t_s$	IAE
LPID-ZN	2.8144	32.4842	20.2026	6.6986
LPID-ISE	4.8337	15.7218	22.1004	6.1206
LPID-IAE	6.0636	8.4150	18.1562	6.2059
LPID-ITAE	6.9033	3.5027	18.1018	6.1417
NPID-ISE	3.4739	6.0375	16.5593	4.6084
NPID-IAE	4.5600	2.0088	9.5668	4.8222
NPID-ITAE	4.4751	4.4037	12.5920	4.9683

As it can be noticed in Table 5, it becomes clear that the proposed method offers better overall performance than the Z-N method and

the Tavakoli’s method. Especially, NPID-IAE and NPID-ITAE tunings offer a significant improvement. In the case of IAE tuning,  $t_r$ ,  $M_p$ ,  $t_s$  and IAE of NPID-IAE tuning are approximately 4.6 seconds, 2%, 9.6 seconds and 4.8, respectively while for LPID-IAE tuning are 1.3, 4.2, 1.9, 1.3 times bigger or longer, respectively. Among the proposed tunings, as expected, NPID-ISE tuning provides shorter  $t_r$  and smaller IAE than both NPID-IAE and NPID-ITAE tunings at the expense of bigger  $M_p$  and longer  $t_s$ .

**5.2 Process II**

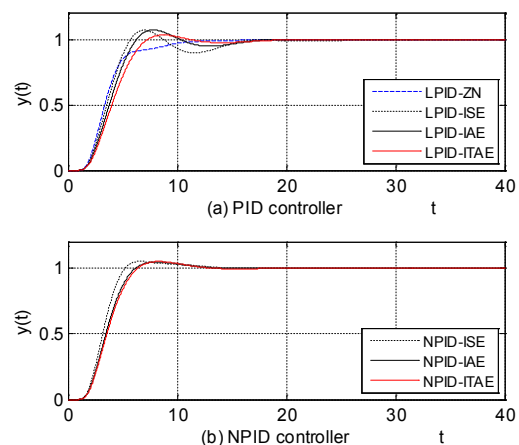
Secondly, the third-order process model was considered:

$$G_p(s) = \frac{e^{-s}}{(1 + 0.5s)(1 + s)(1 + 2s)}, \quad (15)$$

where,  $K= 1.001$ ,  $T= 2.306$  and  $L= 2.32$  were obtained as the FOPDT model parameters. It can be seen that  $L/T \approx 1$  in this process. The PID controller and the NPID controller give the following tunings as listed in Table 6. Figure 8 depicts the unit step responses.

**Table 6.** PID and NPID controller tunings for Process II

Controller Type	Tuning Method	Parameters		
		$K_p$	$T_i$	$T_d$
PID	LPID-ZN	1.1916	4.6400	1.1600
	LPID-ISE	0.9950	3.9600	0.0256
	LPID-IAE	0.8283	3.2081	0.0256
	LPID-ITAE	0.7226	3.0020	0.1331
NPID	NPID-ISE	1.3304	2.6815	0.9583
	NPID-IAE	1.0775	2.5240	0.7713
	NPID-ITAE	1.0051	2.4469	0.6666



**Figure 8.** Responses for the unit step input of Process II

The improvement obtained using the proposed method is again clear. The comparison between the LPID-IAE and the NPID-IAE tunings from Table 7 shows that the proposed method provides more improved quantitative performance.

**Table 7.** Quantitative comparison of the SP tracking responses for Process II

Tuning Method	SP tracking performance			
	$t_r$	$M_p$	$t_s$	IAE
LPID-ZN	7.0113	-	10.6807	3.8839
LPID-ISE	3.5471	7.2059	15.5931	4.2545
LPID-IAE	3.9637	7.4227	16.0999	4.2553
LPID-ITAE	4.6614	3.5598	15.4307	4.3244
NPID-ISE	3.1020	5.1367	11.3887	3.4821
NPID-IAE	3.8304	4.6129	11.1456	3.7697
NPID-ITAE	4.0093	5.0046	11.1250	3.8845

### 5.3 Process III

Finally, the following third-order process model was considered:

$$G_p(s) = \frac{e^{-5s}}{(1+s)^2(1+2s)}, \quad (16)$$

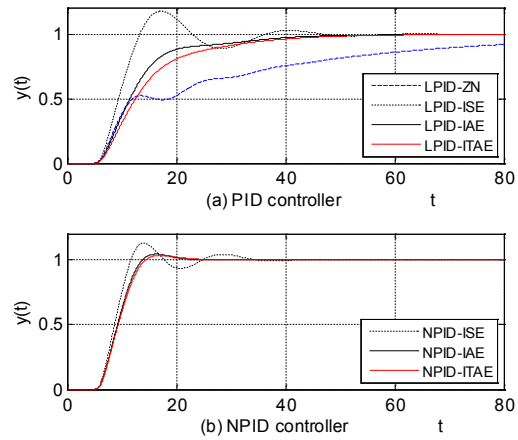
where,  $K=1.002$ ,  $\tau=2.497$ ,  $L=6.64$  were obtained as the FOPDT model parameters, and it can be seen that  $L/\tau \approx 2.7 > 1$  in this process. The three settings are summarized in Table 8.

**Table 8.** PID and NPID controller tunings for Process III

Controller Type	Tuning Method	Parameters		
		$K_p$	$T_i$	$T_d$
PID	LPID-ZN	0.4504	13.2800	3.3200
	LPID-ISE	0.5702	5.2092	0.0277
	LPID-IAE	0.3491	4.8427	0.0277
	LPID-ITAE	0.2894	4.4890	0.1476
NPID	NPID-ISE	0.7797	4.2204	2.1511
	NPID-IAE	0.6310	3.8443	1.6223
	NPID-ITAE	0.6059	3.7975	1.4699

The unit step responses obtained using the three settings are compared in Figure 9. A clear enhancement in the performance is observed from Figure 9 and Table 9 when the proposed method is employed. Its responses exhibit smaller overshoot and remarkably reduced rise time and settling time than the other methods. The tuning methods based on the linear PID controller suffer from poor performance in the case of Process III with  $L/\tau >$

1. Among the proposed tunings, NPID-IAE and NPID-ITAE tunings result in relatively smaller overshoot, shorter settling time and smaller IAE than those using NPID-ISE.



**Figure 9.** Responses for the unit step input of Process III

**Table 9.** Quantitative comparison of the SP tracking responses for Process III

Tuning Method	SP tracking performance			
	$t_r$	$M_p$	$t_s$	IAE
LPID-ZN	-	-	-	26.7754
LPID-ISE	6.5109	17.8481	43.8427	11.8062
LPID-IAE	27.1644	-	42.0165	13.8580
LPID-ITAE	29.6095	-	46.2831	15.4925
NPID-ISE	5.1644	12.7912	31.9624	9.8009
NPID-IAE	6.6818	4.2582	19.4872	9.6143
NPID-ITAE	7.1121	3.4249	20.1375	9.7327

## 6. Conclusion

In this paper, a NPID controller with a nonlinear gain coupled in series with the integral action of a conventional PID controller and three model-based tuning rules was proposed. This new approach incorporated a dimensional analysis and genetic algorithms. For step setpoint tracking, the nonlinear PID controller parameters are tuned optimally based on the first-order plus time delay model and a genetic algorithm. Three performance indices (ISE, IAE and ITAE) were used to measure the performance of the proposed controller. Then, three tuning rules were derived with a set of tuned parameters, potential rule models and the least squares method. The effectiveness of the proposed controller and its rules was verified through a set of simulation works performed on



three experimental processes. On the basis of the simulation results, it has been demonstrated that the proposed NPID controller and tuning approach lead to satisfactory closed-loop responses and provide more improved performance than the linear PID controllers tuned by the two tuning methods. Future studies will focus on proving the absolute stability of the closed-loop system by using the Fade approximation of the FOPTD model and the circle criterion.

## REFERENCES

1. Åström, K. J. & Hägglund, T. (1984). Automatic Tuning of Simple Regulators with Specification of Phase and Amplitude Margins, *Automatica*, 20(5), 645-651.
2. Civelek, Z., Çam, E., Lüy, M. & Mamur, H. (2016). Proportional–integral–derivative parameter optimisation of blade pitch controller in wind turbines by a new intelligent genetic algorithm, *IET Renewable Power Generation*, 10(8), 1220-1228.
3. Girirajkumar, S. M., Kumar, A. A. & Anantharaman, N. (2010). Tuning of a PID Controller for a Real Time Industrial Process using Particle Swarm Optimization, *International Journal of Computer Applications*, 1(7), 35-40.
4. Han, J. (2009). From PID to Active Disturbance Rejection Control, *IEEE Transactions on Industrial Electronics*, 56(3), 900-906.
5. Isayed, B. M. & Hawwa, M. A. (2007). A nonlinear PID control scheme for hard disk drive servosystems. In *Proceedings of 2007 Mediterranean Conference on Control & Automation* (pp. 1-6).
6. Khan, A. A. & Rapal, N. (2006). Fuzzy PID Controller Design, Tuning and Comparison with Conventional PID Controller. In *Proceedings of 2006 IEEE International Conference on Engineering of Intelligent Systems* (pp. 1-6).
7. Kler, D., Rana, K. P. S. & Kumar, V. (2018). A nonlinear PID controller based novel maximum power point tracker for PV systems, *Journal of the Franklin Institute*, 355(16), 7828-7864.
8. Korkmaz, M., Aydoğdu, Ö. & Doğan, H. (2012). Design and performance comparison of variable parameter nonlinear PID controller and genetic algorithm based PID controller. In *Proceedings of 2012 IEEE International Symposium on Innovations in Intelligent Systems and Applications* (pp. 1-5).
9. Kumar, R., Srivastava, S. & Gupta, J. R. P. (2016). Artificial Neural Network based PID Controller for Online Control of Dynamical Systems. In *Proceedings of the 1st IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)* (pp. 1-6).
10. Lee, H. & Jin, G. (1998). On-line Parameter Estimation of Continuous Systems Using Genetic Algorithms, *Journal of Institute of Control, Robotics, and Systems*, 4(1), 76-81.
11. Myron, T. J. (1986). Self-Tuning PID Control – An Expert System Approach. In *Proceedings of IFAC Workshop on Automatic Control in Petroleum* (pp. 77-80).
12. O'Dwyer, A. (2006). *Handbook of PI and PID Controller Tuning Rules*, second ed., 154-180. Imperial College Press.
13. Seraji, H. (1997). A new class of nonlinear PID controllers. In *Proceedings of the 5th IFAC Symposium on Robot Control 1997 (SYROCO '97)*, 30(20) (pp. 65-71).
14. Seraji, H. (1998). A new class of nonlinear PID controllers with robotic applications, *Journal of Robotic Systems*, 15(3), 161–181.
15. So, G. & Jin, G. (2018). Fuzzy-based nonlinear PID controller and its application to CSTR, *Korean Journal of Chemical Engineering*, 35(4), 819-825.

## Acknowledgements

The research for this paper was financially supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF- 2018R1D1A1 B07048954).

16. So, G., Yi, H., Son, Y. & Jin, G. (2018). Temperature Control of a Regasification System for LNG-fuelled Marine Engines Using Nonlinear Control Techniques, *International Journal of Control, Automation and Systems*, 16(6), 3047-3054.
17. Tavakoli, S. & Tavakoli, M. (2003). Optimal Tuning of PID Controllers for First Order Plus Time Delay Models Using Dimensional Analysis. In *Proceedings of the Fourth International Conference on Control and Automation* (pp. 942-946).
18. Zhang, H. & Hu, B. (2012). The Application of Nonlinear PID Controller in Generator Excitation System, *Energy Procedia*, 17(Part A), 202-207.
19. Zheng, C., Su, Y. & Mercorelli, P. (2019). A simple nonlinear PD control for faster and high-precision positioning of servomechanisms with actuator saturation, *Mechanical Systems and Signal Processing*, 121(15), 215-226.
20. Ziegler, J. G. & Nichols, N. B. (1942). Optimum Setting for Automatic Controllers, *Transactions of the A.S.M.E.*, 64(11), 759-768.