Design of an Intelligent NPID Controller Based on Genetic Algorithm for Disturbance Rejection in Single Integrating Process with Time Delay

GunBaek So

Division of Marine Engineering, Korea Maritime and Ocean University, 727 Taejongro, Yeongdo-gu, Busan 49112, Korea; sgb@kmou.ac.kr; Tel.: +82-10-9331-3608

Abstract: The integrating process with time delay (IPTD) is a fundamentally unstable open-loop system due to poles at the origin of the transfer function, and designing controllers with satisfactory control performance is very difficult because of the associated time delay, which is a nonlinear element. Therefore, this study focuses on the design of an intelligent proportional-integral-derivative (PID) controller to improve the regulatory response performance to disturbance in an IPTD, and addresses problems related to optimally tuning each parameter of the controller with a real coded genetic algorithm (RCGA). Each gain of the nonlinear PID (NPID) controller consists of a product of the gains of the linear PID controller and a simple nonlinear function. Each of these nonlinear functions changes the gains in the controller to on line by nonlinearly scaling the error signal. A lead-lag compensator or first-order filter is also added to the controller to mitigate noise, which is a disadvantage of ideal derivative action. The parameters in the controller are optimally tuned by minimizing the integral of time-weighted absolute error (ITAE) using a RCGA. The proposed method is compared with three other methods through simulation to verify its effectiveness.

Keywords: time delay model; nonlinear PID; first-order processes with time delay (FOPTD) model; real coded genetic algorithm (RCGA); disturbance rejection; regulatory performance

1. Introduction

Modeling the plant to be controlled is very important in model-based controller design. The reason is that satisfactory control performance cannot be expected if the proper model for the process is not selected, regardless of how well-tuned the parameters of the controller based on this model are. Most industrial processes are stable, overdamped systems with time delay. Therefore, many first-order processes with time delay (FOPTD) with one real pole in the transfer function have been used to design controllers for them [1–7].

However, the FOPTD model has limitations in approximating highly underdamped dynamics, so modeling with a high-order model plus time delay or integrating a model with time delay may be more appropriate in some cases. The integrating processes are plants with one or more poles at the origin. In particular, the integrating process with time delay (IPTD) is a fundamentally unstable open-loop system due to poles at the origin, and there are difficulties in designing controllers with satisfactory performance because of the associated time delay, which is a nonlinear element.

Many methods have been addressed in the literature to tune PID controllers for integrating systems. These are the direct synthesis method [1,2,8,9], internal model control method [10,11], equating coefficient method [12], stability analysis method [13,14], etc. Chen and Seborg [1] proposed a new direct synthesis approach for controller design based on disturbance rejection rather than set-point tracking. In this design method, the closed-loop time constant is the only tuning parameter, and it has a straightforward relation to the disturbance rejection characteristics. Skogestad [2] proposed direct analytic tuning rules for PID controllers in various integrating systems with time delay and modified the...
integral time to improve the disturbance rejection response. The controller design process in this paper consists of two steps. The first step is to obtain from the original model an approximate first-order plus time delay model or a second-order plus time delay model. The second step is to derive the model-based controller setting. The tuning rules for the PI controller result from a first-order model, whereas those for the PID controller result from a second-order model. Rao and Chidambaram [8] proposed a simple method for the design of a PID controller for IPTD using the direct synthesis approach. The controller had a standard PID controller in series with a first-order filter or lead-lag compensator. In this method, the user needs to adjust two parameters: the desired closed-loop time constant and the set-point weighting parameter. The set-point weighting parameter reduces the overshoot in the set-point tracking response, using a value between 0.3~0.4. Therefore, the user needs to tune only one tuning parameter, the desired closed-loop time constant value. Anil and Sree [9] proposed a parameter tuning method for PID controllers to control various types of IPTDs using the direct synthesis method. This tuning method compares the desired characteristic equation with the characteristic equation of the transfer function which consists of the product of a controller with a filter and integrating processes. The desired characteristic equation consists of multiple poles which are located at the same desired location.

Shamsuzzoha and Lee [10] proposed a simple internal model control (IMC)-based PID controller design method for two representative processes with time delay. The design method was based on disturbance rejection, and a set-point filter was suggested to eliminate the overshoot in the set-point tracking response. Rao and Sree [11] presented a PID controller design method based on internal model control principles for integrating systems with time delay. This controller also consisted of a standard PID controller in series with a lead-lag compensator. Therefore, the user only needs to tune the filter time constant.

Chidambaram and Sree [12] proposed a simple tuning method of PID controllers for a pure integrating process with time delay. This method was based on equating the coefficients of corresponding powers of \( s \) in the numerator to \( \alpha \) times of those in the denominator of the closed-loop transfer function for the set-point input.

In terms of tuning rules based on the stability analysis method, Luyben [13], expanded the method of Tyreus and Luyben [15], and Kookos et al. [14] proposed an online PI controller tuning method.

Generally, even if the desired value in the starting/warming-up mode is determined when operating the process, the set-point is gradually increased in a piecewise step instead of changing the set-point to the target value at one time to reach the target value while observing the operating status of the process.

Once the starting and warming-up process finishes, an operator no longer changes the set-point and sets the controller to running mode. After that, the role of the controller is to make the output return to the steady-state as quickly as possible whenever there is a disturbance.

Therefore, this study focuses on the design of a PID controller to improve the regulatory response performance to disturbance. For this purpose, this study designs an intelligent nonlinear PID (NPID) controller that introduces a simple nonlinear function into the structure of a conventional parallel linear PID (LPID) controller and addresses how to optimally tune each parameter of the controller. Each gain of the NPID controller consists of the product of the gains of the LPID controller and a simple nonlinear function. Each of these nonlinear functions intelligently changes the gains in the controller to online by nonlinearly scaling the error signal. A lead-lag compensator or first-order filter is also added to the controller to mitigate noise, which is a disadvantage of ideal derivative action. The parameters in the controller and filter are optimally tuned by minimizing the integral of the time-weighted absolute error (ITAE) using a real coded genetic algorithm (RCGA). GA can be applied to various optimization problems such as fuzzy control and neural networks, etc. Optimization is essentially the process of finding a more satisfactory solution to a problem. GA creates competing candidates and evolves them through the process.
of natural selection. Poor quality solutions are prone to extinction, while better quality solutions survive and reproduce. GA reproduces the optimal solution by repeating this process several times.

The proposed method is applied to the IPTD model and compared with three other methods through simulation to verify its effectiveness.

This paper is organized as follows: In Section 2, a brief overview about the controlled processes and three existing tuning rules is given. Section 3 describes the design method of the intelligent nonlinear PID controller and discusses how to optimize the parameters. Section 4 applies the proposed NPID controller to control the two processes, and its performance is compared to the existing PID controller. Section 5 highlights the conclusion of this paper.

2. SIPTD Model and PID Controller

2.1. SIPTD Model

First, this section provides a brief overview of the two types of single integrating processes with time delay (SIPTD) in this study.

The first controlled object is a pure integrating process with time delay (PIPTD) as shown in Equation (1) its parameters are \(k = 1\) and \(L = 5\).

- **PIPTD:**
  \[
  P(s) = \frac{ke^{-Ls}}{s}
  \]
  where \(k\) and \(L\) denote a gain and a time delay of process, respectively.

The second controlled object is an integrating and first-order process with time delay (IFOPTD) as shown in Equation (2) its parameters are \(k = 1\), \(L = 0.2\), and \(\tau = 1\), respectively.

- **IFOPTD:**
  \[
  P(s) = \frac{ke^{-Ls}}{s(1 + \tau s)}
  \]
  where \(\tau\) denotes a time constant of process.

The two types of processes above are difficult to control because they have one pole at the origin and time delay.

2.2. PID Controller

This section provides an overview of the controllers to be compared.

2.2.1. Luyben’s PID Controller

Luyben [13] proposed a method to tune each parameter in a parallel form of a PID controller based on the stability analysis method (hereafter referred to as Luyben-SA). This method uses the ultimate gain and the ultimate period to improve Ziegler-Nichols’ frequency response method. The results are summarized as follows in Equation (3).

- **PID controller** \(C(s)\):
  \[
  C(s) = k_p(1 + \frac{1}{\tau_i s} + \tau_d s),
  k_p = \frac{k_u}{2.2},
  \tau_i = 2.2p_u,
  \tau_d = \frac{p_u}{6.3}
  \]
  where \(k_u\), \(\tau_i\), and \(\tau_d\) denote proportional gain, integral time, and derivative time in the parallel PID controller, respectively. \(k_u\) and \(p_u\) are the critical gain and the critical period of the closed-loop system, respectively.
2.2.2. Skogestad’s PID Controller

Skogestad [2] proposed a method to tune each parameter in a series form of a PID controller based on the direct synthesis method (hereafter referred to as Sk-DS). The results are summarized as follows in Equation (4).

\[ C(s) = \bar{k}_p \left( 1 + \frac{1}{\tau_i s} \right) \left( 1 + \tau_d s \right) \]  \[ (4) \]

where \( \bar{k}_p, \tau_i, \) and \( \tau_d \) denote proportional, integral, and derivative gains in a series form of a PID controller, respectively.

- PI controller for a PIPTD:

\[ \bar{k}_p = \frac{1}{k} \left( \frac{1}{\lambda + L} \right), \quad \tau_i = \frac{1}{w(\lambda + L)} \] \[ (5) \]

- PID controller for an IFOPTD:

\[ \bar{k}_p = \frac{1}{k} \left( \frac{1}{\lambda + L} \right), \quad \tau_i = \frac{w(\lambda + L)}{\tau_d}, \quad \tau_d = \tau \] \[ (6) \]

where \( \lambda \) is the desired closed-loop time constant and is the sole tuning parameter for the controller.

2.2.3. Rao’s PID Controller

Rao et al. [8] proposed a method to tune each parameter in a parallel form of a PID controller in series with a first-order filter or a lead-lag compensator based on the direct synthesis method (hereafter referred to as Rao-DS).

- PI controller for a PIPTD:

In a PIPTD model, a first-order filter is in series with a parallel PID controller, and the results are as follows in Equation (7).

\[ C(s) = k_p \left( 1 + \frac{1}{\tau_i s} + \tau_d s \right) \frac{1}{(\beta s + 1)} \] \[ (7) \]

where \( k_p = \frac{2\lambda + 1.5L}{(\lambda^2 + 2\lambda L + 0.5L^2) \beta}, \quad \tau_i = 2\lambda + 1.5L, \quad \tau_d = \frac{4\lambda + 0.5L^2}{\lambda^2 + 2\lambda L + 0.5L^2}, \quad \beta = \frac{0.5L^2}{\lambda^2 + 2\lambda L + 0.5L^2}, \quad \lambda = (0.8 \sim 3)L, \) where \( \lambda \) is the desired closed-loop time constant and is a tuning parameter for the controller, and \( \beta \) is a constant related to the filter.

- PID controller for an IFOPTD:

In an IFOPTD model, a lead-lag compensator is in series with a parallel PID controller, and the results are summarized as follows in Equation (8).

\[ C(s) = k_p \left( 1 + \frac{1}{\tau_i s} + \tau_d s \right) \frac{a s + 1}{(\beta s + 1)} \] \[ (8) \]

where \( k_p = \frac{\eta_1}{(\lambda^2 + 1.5\lambda L + 0.5L^2) \eta_2 - \eta_3}, \quad \tau_i = \eta_1, \quad \tau_d = \frac{\eta_2}{\eta_1}, \quad \eta_1 = 3\lambda + L, \quad \eta_2 = \frac{(0.5L - \tau) \lambda^3 + (3\tau^2 - 1.5\tau) \lambda^2 + 3\tau \lambda^2 + 0.5\lambda^2 \tau^2}{\tau (\tau + 0.5L)}, \quad \lambda = (0.8 \sim 3)L, \quad \alpha = 0.5L, \quad \beta = 0.1\beta_1, \quad \beta_1 = \frac{0.5\lambda^3}{\tau (3\lambda^2 + 1.5\lambda L + 0.5L^2) \eta_2}, \quad a \) and \( \beta \) are constants related to the compensator.

3. Design of Intelligent NPID Controller

3.1. Proposed Intelligent NPID Controller

Consider the closed-loop control system depicted in Figure 1. It consists of a PID controller \( C(s) \) and a controlled object \( P(s) \).
where $1_{2}(3_{2}0_{.5})$ denotes time-varying gains, which are nonlinear functions of error $e$. $\alpha$ and $\beta$ are constants related to the filter.

If the standard form is used as the PID controller in Figure 1, the control input and the transfer function yield:

$$u(t) = k_pe(t) + k_i \int_0^t e(\tau) d\tau + k_d \frac{d e(t)}{d t}$$

(9)

$$C(s) = k_p + \frac{k_i}{s} + k_d s$$

(10)

where $k_p$, $k_i$, and $k_d$ denote proportional, integral, and derivative gains in the parallel PID controller, respectively.

The linear controller of Equations (9) and (10) with fixed gains can be expected to perform the satisfactory control as long as the process is running in a range close to a nominal operating point. However, it is necessary to readjust the parameters of the controller because the parameters of the controlled object may change during operation due to changes in the environment. Moreover, the controller may become unstable due to deteriorated performance by severe disturbance.

In such cases, better performance can be maintained even if the operating environment changes if the gains of the PID controller can be continuously adjusted online according to the operating conditions.

The ideal derivative term in Equations (9) and (10) may cause a derivative kick because of the amplified noise when high-frequency noise is mixed in the signal. Therefore, this study proposes an NPID controller as shown in Equation (11), which adds a lead-lag filter to the controller.

$$C(s) = [K_p(e) + \frac{K_i(e)}{s}] \frac{1 + \alpha s}{1 + \beta s}$$

(11)

where $K_p(e)$, $K_i(e)$, and $K_d(e)$ are time-varying gains, which are nonlinear functions of error $e$. $\alpha$ and $\beta$ are constants related to the filter.

- Nonlinear proportional gain

The proportional action is proportional to the product of the proportional gain and error. If the proportional action is too large, overshoot or oscillation occurs due to excessive control. To improve the control performance, the proportional action must, therefore, be increased to enhance the response speed when the error is large and reduce the proportional action when the response reaches the target value. In this regard, the control performance can be intelligently improved by scaling the error by Equations (12) and (13) and adjusting magnitude of the proportional gain according to the error.

$$K_p(e) = k_p g_p(e)$$

(12)

$$g_p(e) = a_0 + a_1 \left\{ \frac{2}{1 + \exp(-a_2 |e|)} - 1 \right\}$$

(13)
where \( k_p \) is a positive constant, and \( g_p(e) \) is a nonlinear function with three user-defined positive parameters, \( a_0 \), \( a_1 \), and \( a_2 \). 

\( g_p(e) \) is upper-bounded by \( a_0 + a_1 \) when \( e \to \pm \infty \), lower-bounded by \( a_0 \) when \( e = 0 \). That is, \( g_p(e) \) has a value between \( a_0 \) and \( a_0 + a_1 \). Thus \( a_1 \) determines the range of variation of \( g_p(e) \), while \( a_2 \) specifies the rate of variation of \( g_p(e) \).

**Nonlinear integral gain**

The magnitude of the integral action is proportional to the product of the absolute value of the cumulative error and the integral gain. When the absolute value of the error is small, the integral gain must be increased to reduce the steady-state error. When the absolute value of the error is large, the integral gain must be reduced to prepare for overshoot occurrence. Based on these, Equations (14)–(16) must be used as an integral action.

\[
\begin{align*}
K_i(e) &= k_i v(t) \quad \text{(14)} \\
v(t) &= g_i(e) e(t) \quad \text{(15)} \\
g_i(e) &= \exp(-b_0 e^2) \quad \text{(16)}
\end{align*}
\]

where \( k_i \) is the positive gain, \( v(t) \) the scaled error, and \( g_i(e) \) a nonlinear function with parameter \( b_0 (>0) \). The larger the \( b_0 \) value, the narrower the width.

**Nonlinear derivative gain**

The derivative action is proportional to the product of the derivative gain and the rate of error change. And, as the proportional and integral actions increase, damping is applied because the output is predicted to increase accordingly. The nonlinear function of Equation (13) for realizing the proportional gain is used as the nonlinear function for scaling the error. This reduces the number of parameters that need to be tuned. In terms of using a derivative controller, noise should be blocked because the derivative term amplifies signals such as high-frequency noise.

\[
K_d(e) = k_d g_p(e) \quad \text{(17)}
\]

### 3.2. Real coded Genetic Algorithm

Artificial intelligence refers to a research field in which a machine imitates or metaphorically applies human intelligence. When a problem cannot be solved deterministically, the methods taken can often be included in the category of artificial intelligence. Artificial intelligence techniques include rule-based expert system, fuzzy expert system, neural network, evolutionary computation, hybrid artificial intelligence system, and data mining.

The evolutionary methodology for machine learning is based on natural selection and genetic computational models, which is called evolutionary computation. Evolutionary computation is a broad concept and includes genetic algorithms, evolution strategies, and genetic programming. All of these techniques mimic evolution using operators such as reproduction, crossover, and mutation.

RCGA is a statistical search algorithm based on biological evolution to create a population of individuals, evaluate the fitness of the individuals, and create a new individuals using genetic operators. RCGAs can be applied to various optimization problems such as fuzzy control and neural network control, etc.

The search process of the genetic algorithm is divided into five stages: population initialization, fitness evaluation, reproduction, crossover, and mutation. Figure 2 shows a basic flowchart of RCGA.
The evolutionary methodology for machine learning is based on natural selection and genetic computational models, which is called evolutionary computation. Evolutionary computation is a broad concept and includes genetic algorithms, evolution strategies, and genetic programming. All of these techniques mimic evolution using operators such as reproduction, crossover, and mutation.

RCGA is a statistical search algorithm based on biological evolution to create a population of individuals, evaluate the fitness of the individuals, and create a new individual using genetic operators. RCGAs can be applied to various optimization problems such as fuzzy control and neural network control, etc.

The search process of the genetic algorithm is divided into five stages: population initialization, fitness evaluation, reproduction, crossover, and mutation. Figure 2 shows a basic flowchart of RCGA.

- **Stage 1:** An initial population is formed that consists of individuals likely to be a solution to a given matter.
- **Stage 2:** All individuals in the population are evaluated for fitness for each generation. The fitness is calculated by the objective function selected to solve the given matter.
- **Stage 3:** The individuals in the current population are selected for reproduction depending on the relative fitness values of the individuals. Generally, the concept of survival of the fittest is applied to the case where individuals with high-performance are more likely to be selected, and those with poor performance disappear from the population.
- **Stage 4:** Selected individuals exchange and combine their genetic information with each other through crossover to generate offspring.
- **Stage 5:** Finally, they provide a means of introducing new information into the population by arbitrarily changing the genetic information of the selected individual through mutation.

In this way, the newly formed population repeats the fitness evaluation, reproduction, crossover, and mutation operations previously performed until the optimal solution is found.

### 3.3. Tuning of NPID Controller

As described above, the proposed NPID controller has three time-variable gains and a filter, with eight or nine parameters that need tuning. As shown in Figure 3, these parameters need proper tuning to improve the regulatory response performance to disturbance.

Optimal tuning is performed by using RCGA [16–18] and minimizing the ITAE index of Equation (18).

\[
J(\phi) = \int_0^{t_f} t |y_r - y| dt
\]  

(18)

where \( \phi \) is a vector composed of adjusting parameters related to the gains of the NPID controller, \( y_r(t) \) is the set-point, \( y(t) \) the output, and \( t_f \) the integral time.
3.3. Tuning of NPID Controller

As described above, the proposed NPID controller has three time-variable gains and a filter, with eight or nine parameters that need tuning. As shown in Figure 3, these parameters need proper tuning to improve the regulatory response performance to disturbance.

Figure 3. Tuning of the NPID controller with lead-lag compensator.

For the control parameters of RCGA, \( P_{size} = 40 \) for the population size, \( P_c = 0.9 \) for the crossover probability, and \( P_m = 0.05 \) and \( b = 5 \) for the dynamic mutation are considered. Constraints used in the optimization problem are \( 0 \leq k_p, k_d \leq 1, 0 \leq k_i \leq 0.1, 0 \leq a_0, a_1, a_2 \leq 0.5, 0 \leq b_0 \leq 1.0, \) and \( 0 \leq \beta \leq 0.5 \) for PIPTD, and \( 0 \leq k_p, k_i, k_d \leq 5.0, 0 \leq a_0, a_1, a_2 \leq 2.0, 0 \leq b_0 \leq 2.0, \) and \( 0 \leq \alpha, \beta \leq 0.5 \) for IFOPTD. Since the accuracy and convergence of the solution obtained from RCGA depend on the initial population, all of the tuned values are tuned five times as independent seeds and averaged.

Figures 4 and 5 show the evolutionary tuning processes for PIPTD and IFOPTD. As shown in the Figures 4 and 5, the solution is found at approximately 60 and 50 generations, respectively.

Figure 4. RCGA-based evolutionary tuning for PIPTD.

Figure 5. RCGA-based evolutionary tuning for IFOPTD.
The tuning results of the NPID controllers for PIPTD and IFOPTD are shown in Tables 1 and 2, respectively. Tables 1 and 2 also show the parameter values of the three other conventional controllers obtained by using Equations (3)–(7) above.

Table 1. Controller tuning for PIPTD.

| Tuning Methods | $k_p$ | $k_d$ | $k_i$ | Parameters |
|----------------|-------|-------|-------|------------|
| Proposed       | 0.425 | 0.900 | 0.018 | $\beta = 0.40$ |
| Rao-DS         | 0.200 | 0.429 | 0.011 | $\beta = 0.714$ |
| Sk-DS          | 0.100 | 0    | 0.004 | $w = 2.50, \lambda = L$ |
| Luyben-SA      | 0.116 | 0.367 | 0.003 | $k_u = 0.2546, p_u = 20$ |

Table 2. Controller tuning for IFOPTD.

| Tuning Methods | $k_p$ | $k_d$ | $k_i$ | Parameters |
|----------------|-------|-------|-------|------------|
| Proposed       | 3.703 | 1.581 | 4.999 | $\alpha = 0.062, \beta = 0.098$ |
| Rao-DS         | 4.906 | 2.582 | 2.453 | $\alpha = 0.100, \beta = 0.005$ |
| Sk-DS          | 4.063 | 2.500 | 1.563 | $w = 4.0, \lambda = L$ |
| Luyben-SA      | 2.135 | 1.044 | 0.315 | $k_u = 4.696, p_u = 3.08$ |

4. Simulation and Review

To verify the effectiveness of the proposed NPID controller, a set of simulation works on two virtual processes is performed.

The regulatory performance of the proposed NPID controllers is compared with those of the PID controllers by Luyben-SA, Sk-DS, and Rao-DS. For quantitative comparison in each simulation work, the performance measures such as peak time ($t_{peak}$), maximum peak error ($M_{peak}$), recovery time ($t_{rcy}$), and integral of absolute error ($IAE$) are calculated. $M_{peak}$ indicates $|y_r - y_{max}|$ or $|y_r - y_{min}|$, and $t_{rcy}$ is the time required for recovering the output $y$ to less than 2% of the set-point $y_r$.

The overall performance evaluation is performed based on $IAE$ considering $t_{peak}$, $M_{peak}$, and $t_{rcy}$, and the smaller the $IAE$ value, the better the overall performance.

4.1. PIPTD Model

Firstly, the PIPTD model in Equation (1) is considered.

4.1.1. Regulatory Response to Disturbance

Simulation work is carried out to demonstrate the disturbance rejection capability of the proposed method. A step load disturbance of a magnitude of 0.1 is introduced into the perfect PIPTD model at $t = 200$ [s] while the system is operating at output magnitude of 1 and the regulatory responses are shown in Figure 4. The performance measures for quantitative comparison are calculated and listed in Table 3. As shown in Figure 6 and Table 3, although all of the disturbance responses approach the set-point over time, the proposed method and Rao-DS are able to reject the disturbance quickly and bring the output back to the steady-state value. In particular, Luyben-SA shows a very long $t_{rcy}$ and Sk-DS has a very large $M_{peak}$.

The $IAE$ values are also smaller in the order of the proposed method, Rao-DS, Sk-DS, and Luyben-SA. Therefore, the proposed method has the best performance and Luyben-SA has the worst.
recovery times. In particular, Sk-DS has a very large sensitive to parameter changes, while the other two methods are unsatisfactory due to longer the worst.

Luyben-SA. Therefore, the proposed method has the best response and Luyben-SA has the proposed method and Rao-DS method are less sen-

Quantitative comparison of responses to disturbance in PIPTD with

Table 5. Regulatory responses to disturbance in PIPTD with +10% parameter mismatch.

| Tuning Methods | $t_{peak}$ | $M_{peak}$ | $t_{rcy}$ | IAE   |
|----------------|------------|------------|-----------|-------|
| Proposed       | 213.37     | 1.643      | 41.74     | 8.752 |
| Rao-DS         | 219.31     | 1.959      | 61.93     | 25.00 |
| Sk-DS          | 222.74     | 1.726      | 145.3     | 37.96 |
| Luyben-SA      | 212.64     | 1.601      | 43.35     | 6.215 |

Figure 6. Regulatory responses to disturbance in perfect model of PIPTD.

4.1.2. Regulatory Response to Parameter Change

The parameters in the process may change during operation. The robust performance is checked by assuming a perturbation of −10% or +10% in nominal process parameter values.

Figure 7 shows the regulatory responses when a stepwise load disturbance of magnitude of $d = 0.1$ is applied in case the process gain $k$ and time delay $L$ are simultaneously reduced by 10% of the nominal value. Figure 8 shows the regulatory responses when simultaneously increased by 10%.

Figure 7. Regulatory responses to disturbance in PIPTD with −10% parameter mismatch.

Figure 8. Regulatory responses to disturbance in PIPTD with +10% parameter mismatch.
As shown in Figures 7 and 8, the proposed method and Rao-DS method are less sensitive to parameter changes, while the other two methods are unsatisfactory due to longer recovery times. In particular, Sk-DS has a very large $M_{\text{peak}}$ and Luyben-SA has a very long $t_{\text{rcy}}$.

This can also be confirmed by the quantitative results shown in Tables 4 and 5, where the $IAE$ values are smaller in the order of the proposed method, Rao-DS, Sk-DS, and Luyben-SA. Therefore, the proposed method has the best response and Luyben-SA has the worst.

**Table 4.** Quantitative comparison of responses to disturbance in PIPTD with $-10\%$ parameter mismatch.

| Tuning Methods | $t_{\text{peak}}$ | $M_{\text{peak}}$ | $t_{\text{rcy}}$ | $IAE$  |
|---------------|-------------------|-------------------|------------------|-------|
| Proposed      | 212.05            | 1.513             | 48.85            | 6.837 |
| Rao-DS        | 212.84            | 1.554             | 41.36            | 8.926 |
| Sk-DS         | 219.89            | 1.871             | 41.36            | 25.91 |
| Luyben-SA     | 223.40            | 1.701             | 141.3            | 37.99 |

**Table 5.** Quantitative comparison of responses to disturbance in PIPTD with $+10\%$ parameter mismatch.

| Tuning Methods | $t_{\text{peak}}$ | $M_{\text{peak}}$ | $t_{\text{rcy}}$ | $IAE$  |
|---------------|-------------------|-------------------|------------------|-------|
| Proposed      | 213.28            | 1.700             | 58.81            | 8.930 |
| Rao-DS        | 213.98            | 1.700             | 60.45            | 9.479 |
| Sk-DS         | 219.18            | 2.061             | 74.12            | 24.96 |
| Luyben-SA     | 215.46            | 1.754             | 147.9            | 37.94 |

**4.1.3. Regulatory Response to Noise**

A simulation is carried out to verify the performance of the proposed method in the presence of measurement noise. The output is perturbed by an additive white Gaussian noise component $N(0, 0.05^2)$, as shown in Figure 9b. Figure 9a and Table 6 show that the proposed method and Rao-DS have little change in the responses, but the response of the Luyben-SA is severely distorted due to the ideal derivative action. Sk-DS does not use derivative action, so the effect of noise is insignificant.

**Table 6.** Quantitative comparison of the responses to disturbance in PIPTD under Gaussian noise $N(0, 0.05^2)$.

| Tuning Methods | $t_{\text{peak}}$ | $M_{\text{peak}}$ | $t_{\text{rcy}}$ | $IAE$  |
|---------------|-------------------|-------------------|------------------|-------|
| Proposed      | 212.53            | 1.604             | 44.18            | 9.261 |
| Rao-DS        | 213.98            | 1.640             | 50.62            | 11.34 |
| Sk-DS         | 213.40            | 1.959             | 61.51            | 26.75 |
| Luyben-SA     | 225.03            | 1.789             | -                | 36.19 |
noise component $N(0, 0.05^2)$, as shown in Figure 9b. Figure 9a and Table 6 show that the proposed method and Rao-DS have little change in the responses, but the response of the Luyben-SA is severely distorted due to the ideal derivative action. Sk-DS does not use derivative action, so the effect of noise is insignificant.

![Figure 9a: Regulatory responses](image1)

**Figure 9.** Regulatory responses to disturbance in PIPTD under Gaussian noise $N(0, 0.05^2)$.

### 4.2. IFOPTD Model

Secondly, the IFOPTD model in Equation (2) was considered.

#### 4.2.1. Regulatory Response to Disturbance

A unit step change in load disturbance is introduced to perfect IFOPTD model at $t = 20$ [s] while the system is operating at output magnitude of 1; the regulatory responses are shown in Figure 10. The performance measures for quantitative comparison are calculated and listed in Table 7.

![Figure 10: Regulatory responses](image2)

**Figure 10.** Regulatory responses to disturbance in perfect model of IFOPTD.

| Tuning Methods | $t_{\text{peak}}$ | $M_{\text{peak}}$ | $t_{\text{rcy}}$ | $\text{IAE}$ |
|----------------|------------------|------------------|----------------|-----------|
| Proposed       | 20.17            | 1.181            | 2.543          | 0.204     |
| Rao-DS         | 21.48            | 1.179            | 2.543          | 0.408     |
| Sk-DS          | 21.55            | 1.222            | 7.742          | 0.639     |
| Luyben-SA      | 22.33            | 1.484            | 24.26          | 3.154     |

**Table 7.** Quantitative comparison of responses to disturbance in perfect model of IFOPTD.

As shown in Figure 10 and Table 7, the proposed method is able to reject the disturbance quickly and bring the output back to the steady-state value. In contrast, the other
methods require more time to recover from disturbances. In particular, Luyben-SA shows a larger $M_{\text{peak}}$ and a longer $t_{\text{rcy}}$ than the others. In case of Rao-DS, the $M_{\text{peak}}$ value is similar to that of the proposed method, but the $t_{\text{rcy}}$ is relatively long and the IAE is large.

In addition, the IAE values are smaller in the order of the proposed method, Rao-DS, Sk-DS, and Luyben-SA.

4.2.2. Regulatory Response to Parameter Change

As in the previous case, the robust performance is checked by assuming a perturbation of −10% or +10% in nominal process parameter values.

Figure 11 shows the regulatory responses when a stepwise load disturbance of magnitude of $d = 1$ is applied in case the process gain $k$, time delay $L$, and time constant $\tau$ are simultaneously reduced by 10% of the nominal value. Figure 12 shows the regulatory responses when simultaneously increased by 10%. As shown in Figures 11 and 12, the proposed method shows satisfactory responses even if the parameters change. In particular, Luyben-SA has a significantly larger $M_{\text{peak}}$ and a longer $t_{\text{rcy}}$ compared to the other methods.

This can also be confirmed by the quantitative results shown in Tables 8 and 9, where the IAE values are smaller in the order of the proposed method, Rao-DS, Sk-DS, and Luyben-SA.

**Table 8.** Quantitative comparison of responses to disturbance in IFOPTD with −10% parameter mismatch.

| Tuning Methods   | $t_{\text{peak}}$ | $M_{\text{peak}}$ | $t_{\text{rcy}}$ | IAE  |
|------------------|-------------------|-------------------|------------------|------|
| Proposed         | 20.17             | 1.181             | 2.543            | 0.204|
| Rao-DS           | 21.48             | 1.179             | 5.614            | 0.408|
| Sk-DS            | 21.55             | 1.222             | 7.742            | 0.639|
| Luyben-SA        | 22.33             | 1.484             | 24.26            | 3.154|
Table 9. Quantitative comparison of responses to disturbance in IFOPTD with +10% parameter mismatch.

| Tuning Methods | \( t_{\text{peak}} \) | \( M_{\text{peak}} \) | \( t_{\text{recy}} \) | IAE |
|----------------|----------------|----------------|--------------------|-----|
| Proposed       | 21.20          | 1.194          | 4.686              | 0.215|
| Rao-DS         | 21.45          | 1.185          | 5.740              | 0.408|
| Sk-DS          | 21.51          | 1.231          | 7.829              | 0.639|
| Luyben-SA      | 22.30          | 1.502          | 24.16              | 3.153|

4.2.3. Regulatory Response to Noise

The output is perturbed by an additive white Gaussian noise component \( N(0, 0.01^2) \), as shown in Figure 13b. As shown in Figure 13a, the proposed method shows satisfactory responses even if there is a measurement noise. This can also be confirmed by the quantitative results shown in Table 10, where the IAE values are smaller in the order of the proposed method, Rao-DS, Sk-DS, and Luyben-SA.

Table 10. Quantitative comparison of responses to disturbance in IFOPTD under Gaussian noise \( N(0, 0.01^2) \).

| Tuning Methods | \( t_{\text{peak}} \) | \( M_{\text{peak}} \) | \( t_{\text{recy}} \) | IAE |
|----------------|----------------|----------------|--------------------|-----|
| Proposed       | 21.17          | 1.182          | 4.751              | 0.267|
| Rao-DS         | 21.48          | 1.180          | 6.106              | 0.445|
| Sk-DS          | 21.55          | 1.222          | 7.536              | 0.662|
| Luyben-SA      | 22.36          | 1.484          | 24.81              | 3.181|

5. Conclusions

This study focused on the design of an intelligent PID controller to improve the regulatory response performance to disturbance. Therefore, this study proposed a nonlinear PID controller with time-varying gains and a filter in a parallel form of linear PID frame...
and discussed how to optimally tune each parameter of the controller. The unique part is that the nonlinear function for realizing the P and D actions uses the same function and parameters to reduce the number of parameters that need tuning. The parameters in the controller were optimally tuned by minimizing the ITAE using RCGA. The search process of the RCGA is divided into five stages: population initialization, fitness evaluation, reproduction, crossover, and mutation. The proposed method was applied to control the PIPTD process and IFOPTD process, and its performance was compared with those of Rao-DS, Sk-DS, and Luyben-SA. The results showed that the performance of the proposed controller was better than that of the other controllers.

**Funding:** This article received no external funding.

**Data Availability Statement:** Data available in a publicly accessible repository.

**Conflicts of Interest:** The authors declare that there is no conflict of interest regarding the publication of this paper.

**References**

1. Chen, D.; Seborg, D.E. PI/PID controller design based on direct synthesis and disturbance rejection. *Ind. Eng. Chem. Res.* **2002**, *41*, 4807–4822. [CrossRef]
2. Skogestad, S. Simple analytic rules for model reduction and PID controller tuning. *J. Process Control* **2003**, *13*, 291–309. [CrossRef]
3. Tavakoli, S.; Tavakoli, M. Optimal tuning of PID controller for first order plus time delay models using dimensional analysis. In Proceedings of the Fourth International Conference on Control and Automation, Montreal, QC, Canada, 12 June 2003; pp. 942–946.
4. O’Dwyer, A. *Handbook of PI and PID Controller Tuning Rules*; Imperial College Press: London, UK, 2009.
5. Cvejn, J. Sub-optimal PID Controller Settings for FOPDT Systems with Long Dead Time. *J. Process Control* **2009**, *19*, 1486–1495. [CrossRef]
6. Jin, G.G.; Son, Y.D. Design of a nonlinear controller and tuning rules for first-order plus time delay models. *Stud. Inform. Control* **2019**, *28*, 157–166. [CrossRef]
7. So, G.B. EA-based design of a nonlinear PID controller using an error scaling technique. *Stud. Inform. Control* **2019**, *28*, 279–287. [CrossRef]
8. Rao, A.S.; Rao, V.S.R.; Chidambaram, M. Direct synthesis-based controller design for integrating processes with time delay. *J. Franklin. Inst.* **2009**, *346*, 38–59.
9. Anil, C.; Sree, R.P. Tuning of PID controllers for integrating systems using direct synthesis method. *Isa Trans.* **2015**, *57*, 211–219. [CrossRef] [PubMed]
10. Shamsuzzoha, M.; Lee, M. PID controller design for integrating processes with time delay. *Korean J. Chem. Eng.* **2008**, *25*, 637–645. [CrossRef]
11. Rao, C.V.N.; Sree, R.P. IMC based controller design for integrating systems with time delay. *Indian Chem. Eng.* **2010**, *52*, 194–218. [CrossRef]
12. Chidambaram, M.; Sree, R.P. A simple method of tuning PID controllers for integrator/dead-time processes. *Comput. Chem. Eng.* **2003**, *27*, 211–215. [CrossRef]
13. Luyben, W.L. Design of proportional integral and derivative controllers for integrating dead-time processes. *Ind. Eng. Chem. Res.* **1996**, *35*, 3480–3483. [CrossRef]
14. Kookos, I.K.; Lygeros, A.I.; Arvanitis, K.G. On-line PI controller tuning for integrator/dead time processes. *Eur. J. Control* **1999**, *5*, 19–31. [CrossRef]
15. Tyreus, B.D.; Luyben, W.L. Tuning PI controllers for integrator/dead-time processes. *Ind. Eng. Chem. Res.* **1992**, *31*, 2625–2628. [CrossRef]
16. Jin, G.G. *Genetic Algorithm and Their Application*; Kyo-Woo Publishing Co.: Seoul, Korea, 2000.
17. Aleksandra, A.S.; Radiša, Ž.J.; Vojislav, M.N.; Nataša, M.N.; Branislav, Ž. Support vector machine for the prediction of heating energy use. *Therm. Sci.* **2018**, *22*, 1171–1181.
18. Jovanović, R.Z.; Sretenović, A.A. Various multistage ensembles for prediction of heating energy consumption. *Modeling Identif. Control* **2015**, *36*, 119–132. [CrossRef]