



## Article

# A Tuning Method via Borges Derivative of a Neural Network-Based Discrete-Time Fractional-Order PID Controller with Hausdorff Difference and Hausdorff Sum

Zhe Gao

Department of Electrical Engineering and Automation, College of Light Industry, Liaoning University, Shenyang 110036, China; gaozhe@lnu.edu.cn

**Abstract:** In this paper, the fractal derivative is introduced into a neural network-based discrete-time fractional-order PID controller in two areas, namely, in the controller's structure and in the parameter optimization algorithm. The first use of the fractal derivative is to reconstruct the fractional-order PID controller by using the Hausdorff difference and Hausdorff sum derived from the Hausdorff derivative and Hausdorff integral. It can avoid the derivation of the Gamma function for the order updating to realize the parameter and order tuning based on neural networks. The other use is the optimization of order and parameters by using Borges derivative. Borges derivative is a kind of fractal derivative as a local fractional-order derivative. The chain rule of composite function is consistent with the integral-order derivative. It is suitable for updating the parameters and the order of the fractional-order PID controller based on neural networks. This paper improves the neural network-based PID controller in two aspects, which accelerates the response speed and improves the control accuracy. Two illustrative examples are given to verify the effectiveness of the proposed neural network-based discrete-time fractional-order PID control scheme with fractal derivatives.



**Citation:** Gao, Z. A Tuning Method via Borges Derivative of a Neural Network-Based Discrete-Time Fractional-Order PID Controller with Hausdorff Difference and Hausdorff Sum. *Fractal Fract.* **2021**, *5*, 23. <https://doi.org/10.3390/fractalfract5010023>

Received: 10 February 2021  
Accepted: 9 March 2021  
Published: 14 March 2021

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



**Copyright:** © 2021 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** fractional-order PID control; neural network control; tuning methods; Hausdorff derivative; Borges derivative

## 1. Introduction

Due to the introductions of integral-order and differential-order, the design of fractional-order proportion-integral-derivative (PID) controllers is more flexible compared with that of integral-order PID controllers, allowing better control, which has attracted attention from many scholars [1]. Due to the simple structure of fractional-order PID controllers, as effective fractional-order controllers, they have been widely used in many control systems and achieved satisfactory control performances [2]. Take the examples of power systems [3], TCP networks [4], DC motors [5], supercapacitor energy storage systems [6], and robotic manipulator systems [7]. In addition, a variety of mixed fractional-order PID controllers have been derived from other algorithms, such as the fuzzy PID controller [8] and fractional-order PID sliding mode controllers [9].

The tuning methods of parameters in fractional-order PID controllers mainly include the analytic method and intelligent optimization algorithms. The analytic method can give the optimal parameters of fractional-order PID controllers with the given performance indexes. Generally, the method of analytic method is to design the parameters of the fractional-order controller in the frequency domain [10]. The dominant pole placement was used to tune the parameters of fractional-order PID controller to obtain a satisfactory load disturbance response in [11]. The phase margin and gain crossover frequency were concerned to present a tuning method of fractional-order PID controllers for minimum phase plants [12]. The robustness on the loop gain variations was guaranteed to tune the parameters of fractional-order PID controller in [13]. Although the analytic method can provide the mathematical expression of the parameter tuning, this method depends on the

mathematical model of the controlled plant. It is difficult to use for the parameter tuning of fractional-order PID controllers of complex nonlinear controlled plants. Therefore, many scholars have studied the parameter tuning methods of fractional-order PID controllers based on various intelligent optimization methods, such as the genetic algorithm [14], the chaotic atom search optimization algorithm [15], the quantum bacterial foraging algorithm [16], the extremum seeking algorithm [17], and the particle swarm optimization algorithm [18]. Although the above intelligent optimization algorithms can obtain the effective parameter tuning results for the nonlinear controlled plant, the parameter tuning methods are offline calculation methods.

As an intelligent optimization and decision algorithm, the neural network algorithm is also applied in the design of fractional-order PID controllers. In [19], the RBF neural network algorithm was used to tune the parameters of fractional-order PID controllers and applied for a wind turbine. However, it is also an offline parameter tuning method and requires a lot of offline data. The difficulty of online parameter tuning methods of fractional-order PID controllers is mainly the adaptive tuning of fractional-orders, due to the global operators of fractional-order derivatives in the Caputo sense or Riemann–Liouville sense, and the fractional-order difference in the Grünwald–Letnikov (G–L) sense. The fractional-order operators also play a role in neural networks. Many scholars have proposed various fractional-order neural networks defined by fractional-order derivatives [20–24]. There are many research results in the cross research of neural networks and fractional-order calculus. Hence, it is worth studying the application of neural network optimization algorithms to the parameter tuning of fractional-order PID controllers.

In order to emphasize the adaptive adjustments of fractional-orders, the parameters of fractional-order PID controller in this paper include the parameters the proportional coefficient, the integral coefficient, and the differential coefficient; and orders integral order and differential order. The neural network algorithm is also used to adjust the parameters and orders of fractional-order PID controllers. It is necessary to study the discretized method of fractional-order operators to design a discrete-time fractional-order PID controller. There are many discretization methods of fractional-order derivatives [25,26]; the simplest discretization form of a fractional-order derivative is the G–L difference [27]. The discretization method via the G–L difference is simple, but the mathematical expression of the discretization result contains the Gamma function related to the order. If the orders in fractional-order PID controllers are adjusted adaptively by a neural network, the derivatives of the Gamma function with respect to the orders need to be obtained. Unfortunately, the derivative of the Gamma function is not a closed-form mathematical expression [28], and it not easy to calculate directly to hinder the adaptive updating of orders.

As a special fractional-order derivative, the fractal derivative deals with the local characteristics. The fractal derivatives are widely used in the modeling of real physical systems [29], such as anomalous diffusion [30] and viscoelastic materials [31]. Meanwhile, the validity of the fractal derivative for achieving chaotic attractors was discussed in [32] to show that the nonlinear system described by fractal derivative can also produce complex, dynamic behavior. The Hausdorff derivative as one of the fractal derivative definitions does not contain the Gamma function [33]. Therefore, a novel discrete-time fractional-order PID controller can be constructed by using the discretized forms of the Hausdorff derivative and the Hausdorff integral, and the adaptive tuning method of parameters and orders can be realized by using neural networks. Besides, the power-law stability of nonlinear systems via the Hausdorff derivative was investigated in [34]. The fractal derivative mainly generalizes the first-order derivative with many definitions. The Borges derivative is another form of fractal derivative which is given in [35], and it can also describes the dynamic behavior of complex systems.

In this paper, fractal calculus theory is applied to the design of a neural network PID controller in two ways. According to the Hausdorff derivative and the Hausdorff integral, the definitions of the Hausdorff difference and Hausdorff sum are derived. Compared with the fractional-order PID controller via the G–L definition, the derivation of the

Gamma function is avoided in parameter updating based on neural networks. In addition, the fractional-order PID controller based on fractal theory has no memory of difference, and the updating of parameters and orders is simpler. Another improvement is to introduce the Borges derivative into the optimization of parameters and orders. The derivation of a composite function based on the Caputo derivative is very complex; hence, the chain rule of the derivative of an integer-order composite function cannot be used directly. However, the chain rule of the Borges derivative of a composite function is consistent with the integral-order derivative, and the effect can also be optimized by adjusting the order. Therefore, this paper applies the Borges derivative to the training of controller parameters and orders, and discusses the problem of order selection in the Borges derivative.

The main contributions of this paper are summarized as follows: (1) The definition and calculation method of Hausdorff difference and Hausdorff sum are proposed, and their introduction into PID controllers is described. (2) A novel type of fractional-order PID controller is proposed to realize the online adjustment of parameters and orders, and the online tuning method of parameters and orders based on the first order gradient information is given. (3) The Borges derivative is introduced into the tuning of parameters and orders, and the influence of the order in the Borges derivative on the optimization performance in regard to parameters and orders is offered. (4) The tuning methods of the parameters and orders of the proposed fractional-order PID controller based on the fractal derivative is proposed, and it improves the optimization speed of the parameters and orders.

The rest of this paper is organized as follows. In Section 2, the definitions of Hausdorff difference and sum derived from Hausdorff derivative and integral are given, and the fractional-order PID controller in the sense of a Riemann–Liouville (R–L) derivative and Caputo derivative with zero-initial-condition is also given. Based on the Hausdorff difference and sum, two kinds of neural network-based tuning methods are proposed for the parameters and fractional-orders of a fractional-order PID controller via the first order and the Borges derivative in Section 3. In Section 4, two illustrative examples are offered to verify the effectiveness of this paper. The conclusion is given in Section 5.

## 2. Problem Statement

In this paper, a novel type of fractional-order PID controller is constructed based on the difference and sum operators extended by the Hausdorff derivative and Hausdorff integral. The Hausdorff derivative with  $\alpha \in \mathbb{R}^+$  for  $t > 0$  is defined in [33] as

$${}^H D_t^\alpha f(t) = \lim_{t' \rightarrow t} \frac{f(t') - f(t)}{t'^\alpha - t^\alpha}, \quad (1)$$

where  $\mathbb{R}^+$  is the set of positive real numbers.

It is not difficult to calculate the Hausdorff derivative if the function is a differentiable function as follows:

$${}^H D_t^\alpha f(t) = \frac{1}{\alpha} \frac{df(t)}{dt} t^{1-\alpha}. \quad (2)$$

The Hausdorff difference for  $k > 0$  derived from the Hausdorff derivative is given as follows.

$${}^H \Delta_k^\alpha f(k) = \frac{1}{\alpha} (x(k) - x(k-1)) k^{1-\alpha}. \quad (3)$$

Similarly, the Hausdorff integral is defined in [32] based on (2) as follows.

$${}^H I_t^\alpha f(t) = \alpha \int_0^t \tau^{\alpha-1} f(\tau) d\tau. \quad (4)$$

Consequently, the Hausdorff sum for  $k > 0$  derived from the Hausdorff integral is determined by

$${}^H \Sigma_k^\alpha f(k) = \sum_{l=1}^k \alpha l^{\alpha-1} f(l). \quad (5)$$

For  $\alpha = 1$  in (3) and (5), the Hausdorff difference and Hausdorff sum become

$${}^H\Delta_k^1 f(k) = (x(k) - x(k-1)),$$

$${}^H\Sigma_k^1 f(k) = \sum_{l=1}^k f(l).$$

Thus means that the Hausdorff difference and Hausdorff sum include the first order difference and sum operations. Therefore, the main investigated issue of this paper is the tuning design of parameters and orders in the fractional-order PID control scheme with neural networks based on the Hausdorff difference and Hausdorff sum.

As the fractional-order PID controller has two more important parameters that are the integral-order and the differential-order, the controller design is more flexible compared with the integer-order PID controller. Generally, the basic definitions of fractional-order derivatives are the R–L definition and Caputo definition. In a continuous-time, fractional-order PID controller, the mathematical descriptions under these two definitions are usually used. The definitions are given as follows.

The Caputo derivative [25] is defined by

$${}_0^C D_t^\alpha f(t) = \frac{1}{\Gamma(P-\alpha)} \int_0^t \frac{f^{(P)}(\tau)}{(t-\tau)^{1+\alpha-P}} d\tau, \quad (6)$$

where  ${}_0^C D_t^\alpha$  is Caputo derivative from 0 to  $t$  with the order  $\alpha \in \mathbb{R}^+$ , and it satisfies  $P-1 < \alpha \leq P$  with  $P \in \mathbb{Z}^+$ ;  $\mathbb{Z}^+$  is the set of positive integer numbers, and the function  $\Gamma(\cdot)$  is the gamma function.

Accordingly, the R–L derivative [25] is defined by

$${}_0^{RL} D_t^\alpha f(t) = \frac{1}{\Gamma(P-\alpha)} \frac{d^P}{dt^P} \int_0^t \frac{f(\tau)}{(t-\tau)^{1+\alpha-P}} d\tau, \quad (7)$$

where  ${}_0^{RL} D_t^\alpha$  is R–L derivative from 0 to  $t$  with the order  $\alpha \in \mathbb{R}^+$  and satisfies  $P-1 \leq \alpha < P$  with  $P \in \mathbb{Z}^+$ .

If the initial conditions are zero for both the Caputo derivative and the R–L derivative; the Laplace transformation of fractional-order derivative with respect to the function  $f(t)$  with the fractional-order  $\alpha$  is  $s^\alpha F(s)$  with the Laplace operator  $s$ , where  $F(s)$  is the Laplace transformation of  $f(t)$ . Then, the transfer function of fractional-order PID controller [36] in the frequency domain is offered as follows.

$$U(s) = k_p + k_i s^{-\lambda} + k_d s^\mu, \quad (8)$$

where  $k_p \in \mathbb{R}^+$ ,  $k_i \in \mathbb{R}^+$ ,  $k_d \in \mathbb{R}^+$ ,  $\lambda \in (0, 2)$ , and  $\mu \in (0, 2)$  are the proportional coefficient, the integral coefficient, the differential coefficient, the integral order, and the differential order; and  $\mathbb{R}^+$  is the set of positive real numbers.

The fractional-order operation in the discrete-time fractional-order PID controller is described by G–L definition. The G–L difference is provided as follows.

$$\Delta^\alpha f(k) = \sum_{l=0}^k (-1)^l c_l^\alpha f(k-l), \quad (9)$$

where the factor  $c_j^\alpha$  is computed by

$$c_j^\alpha = \frac{\Gamma(\alpha+1)}{\Gamma(j+1)\Gamma(\alpha-j+1)}. \quad (10)$$

Generally, the fractional-order operators in the fractional-order controller (8) are discretized by the G–L difference, but the coefficients of the discretized fractional-order operators are all related to the orders, and the relationship between the corresponding coefficient and the order is more complex due to the Gamma function in terms of the order. Therefore, the adaptive adjustments of corresponding orders  $\lambda$  and  $\mu$  are quite difficult via the G–L difference. As the Hausdorff derivative, and its discretization are both local operators, the orders in the discrete-time fractional-order PID controller based on the Hausdorff difference and Hausdorff sum can be adjusted adaptively.

### 3. Main Results

#### 3.1. A Discrete-Time Fractional-Order PID Controller Based on the Hausdorff Difference and Hausdorff Sum

Since the fractional-order operators in fractional-order PID controller defined by Equation (11) are global operators, the adaptive adjustments of orders are not easy to calculate. Therefore, a fractional order PID controller based on the Hausdorff derivative and Hausdorff integral is proposed in this paper. The corresponding controller in the time domain is offered as follows:

$$u(t) = k_p e(t) + k_i {}^H I_t^\lambda e(t) + k_d {}^H D_t^\mu e(t), \quad (11)$$

where  $e(t)$  is the output error.

By using the Hausdorff difference and Hausdorff sum described by Equations (3) and (5), the discrete-time, fractional-order PID controller is given as follows.

$$u(k) = k_p e(k) + k_i {}^H \Sigma_k^\lambda e(k) + k_d {}^H \Delta_k^\mu e(k), \quad (12)$$

where  $e(k)$  is the sampling value of  $e(t)$  at  $t = kT$  with the sampling period  $T$ , and

$${}^H \Sigma_k^\lambda e(k) = \sum_{l=1}^k \lambda l^{\lambda-1} e(l),$$

$${}^H \Delta_k^\mu e(k) = \frac{1}{\mu} (e(k) - e(k-1)) k^{1-\mu}.$$

The influences of different orders on the control effect based on Hausdorff difference and the Hausdorff sum in the fractional-order controller (12) are discussed. Then, we define  $K_\lambda(k) = \lambda k^{\lambda-1}$  and  $K_\mu(k) = k^{1-\mu}/\mu$  in  $u(k)$  respectively, and the curves of  $K_\lambda(k)$  and  $K_\mu(k)$  are drawn in Figures 1 and 2 respectively.

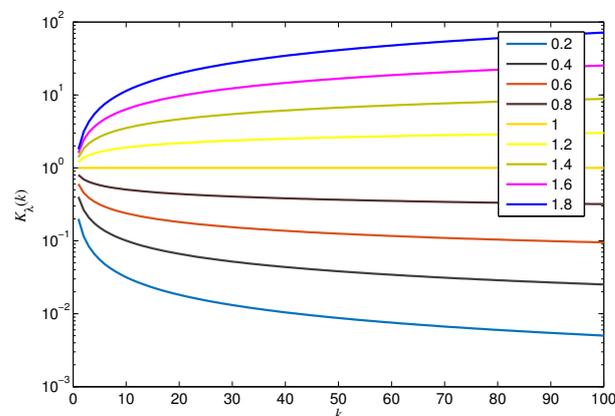
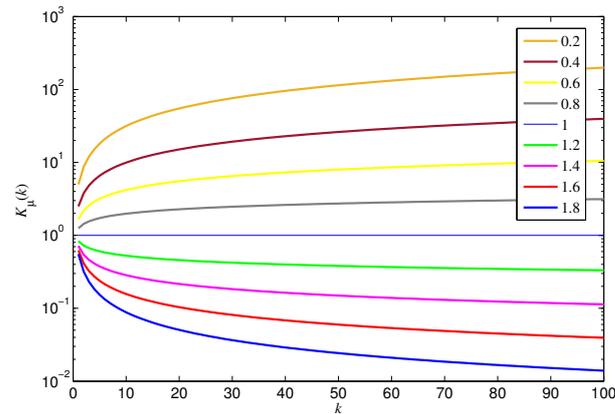


Figure 1. Curves of  $K_\lambda(k)$  with different orders  $\lambda$ .



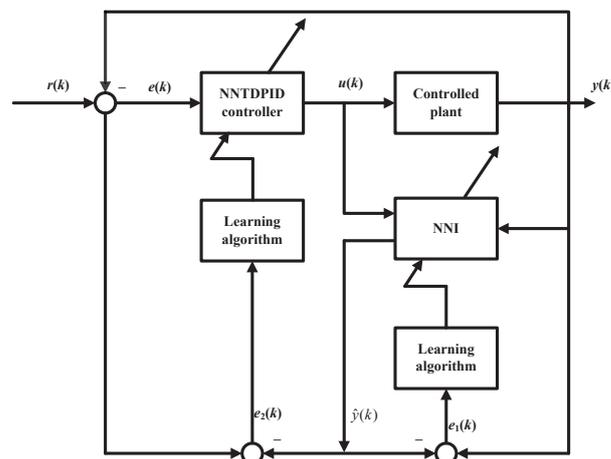
**Figure 2.** Curves of  $K_\mu(k)$  with different orders  $\mu$ .

It can be seen from the curves described in Figures 1 and 2 that  $K_\lambda(k)$  increases and  $K_\mu(k)$  decreases with an increase of fractional-order  $\lambda$  or  $\mu$ . If the order  $\lambda$  is set as a larger value, the role of the current time is larger, and the role of the historical information of the error is smaller. That is to say, more attention is paid to the error information of the current time in  ${}^H\Sigma_k^\lambda e(k)$ . However, a larger order  $\lambda$  makes the sum term too large; hence, it is necessary to choose a reasonable order  $\lambda$ . Correspondingly, the order  $\mu$  is set as a smaller value; then the effect of the difference becomes larger. Therefore, the order  $\mu$  should be made smaller with the increase of the iteration  $k$ .

Next, we give the calculation method of the parameters and orders of the discrete-time fractional-order PID controller (12) trained by the neural network method.

### 3.2. Tuning Method of Parameters and Orders of Discrete-Time Fractional-Order PID Controllers with Neural Networks

The neural network-based discrete-time fractional-order (NNDTFO) PID controller with the neural network identifier (NNI) is the focus, and its structure is shown in Figure 3.



**Figure 3.** Structure of the control system with a neural network-based discrete-time fractional-order (NNDTFO) PID controller and a neural network identifier (NNI).

The nonlinear controlled plant is presented by

$$y(k) = g(y(k - 1), y(k - 2), \dots, y(k - n_y), u(k), u(k - 1), \dots, u(k - n_u)), \quad (13)$$

where  $g(\cdot)$  is the nonlinear function in terms of the control input  $u(k)$  and output  $y(k)$ ,  $n_y \in \mathbb{Z}^+$ ,  $n_u \in \mathbb{Z}^+$ , and  $\mathbb{Z}^+$  is the set of positive integer numbers.

The weight matrices corresponding the connection from the input layer to the hidden layer and the connection from the hidden layer to the output layer are denoted as  $W^{(1)}(k) \in \mathbb{R}^{S_1 \times N}$  and  $W^{(2)}(k) \in \mathbb{R}^{1 \times S_1}$ , where  $N = n_y + n_u + 1$  is the number of the neurons in the input layer.

The input of NNI is  $x(k) = [y(k-1), y(k-2), \dots, y(k-n_y), u(k), u(k-1), \dots, u(k-n_u)]^T$ ; the output  $\hat{y}(k)$  of the output layer is determined by

$$\hat{y}(k) = W^{(2)}(k)O(k) + b^{(2)}(k), \quad (14)$$

where  $O(k) = f(W^{(1)}(k)x(k) + b^{(1)}(k))$  is the output of the hidden layer,  $f(\cdot)$  is activation function of the hidden layer, and  $b^{(1)}(k)$  and  $b^{(2)}(k)$  are the biases of the hidden layer and output layer.

Let the cost function  $J_1(k) = 0.5e_1^2(k)$  with  $e_1(k) = y(k) - \hat{y}(k)$ ; then the updating formulas of the weights and biases in NNI are determined by

$$W^{(1)}(k+1) = W^{(1)}(k) - \zeta s^{(1)}(k)x^T(k), \quad (15)$$

$$W^{(2)}(k+1) = W^{(2)}(k) - \zeta s^{(2)}(k)O^T(k), \quad (16)$$

$$b^{(1)}(k+1) = b^{(1)}(k) - \zeta s^{(1)}(k), \quad (17)$$

$$b^{(2)}(k+1) = b^{(2)}(k) - \zeta s^{(2)}(k), \quad (18)$$

where  $\zeta$  is the learning rate of NNI, and  $s^{(1)}(k)$  and  $s^{(2)}(k)$  are determined by

$$s^{(1)}(k) = \mathcal{F}^{(1)}(k)(W^{(2)}(k))^T s^{(2)}(k),$$

$$s^{(2)}(k) = -e_1(k),$$

$$\mathcal{F}^{(1)}(k) = \text{diag}(\partial f(r_1^{(1)}(k))/\partial r_1^{(1)}(k),$$

$$\partial f(r_2^{(1)}(k))/\partial r_2^{(1)}(k), \dots, \partial f(r_{S_1}^{(1)}(k))/\partial r_{S_1}^{(1)}(k))'$$

and  $r_i^{(1)}(k) = \sum_{j=1}^N w_{ij}^{(1)}(k)x_j(k)$  for  $i = 1, 2, \dots, S_1$ ;  $w_{ij}^{(1)}(k)$  is the entry in  $W^{(1)}(k)$  corresponding to the  $i$ th row and the  $j$ th column; and  $x_j(k)$  is the  $j$ th entry in the input  $x(k)$ .

A BP neural network with one hidden layer is adopted as the NNI for the controlled plant (13). Let the controller for  $k > 0$  be

$$u(k) = k_p(k)e(k) + k_i(k)^H \Sigma_k^{\lambda(k)} e(k) + k_d(k)^H \Delta_k^{\mu(k)} e(k), \quad (19)$$

where  $e(k) = r(k) - y(k)$ ,  $r(k)$  is the reference signal, and

$${}^H \Sigma_k^{\lambda(k)} e(k) = \sum_{l=1}^k \lambda(l) l^{\lambda(l)-1} e(l),$$

$${}^H \Delta_k^{\mu(k)} e(k) = \frac{1}{\mu(k)} (e(k) - e(k-1)) k^{1-\mu(k)}.$$

Define the cost function as  $J_2(k) = 0.5e_2^2(k)$  with  $e_2(k) = r(k) - \hat{y}(k)$ ; then the tuning formulas of parameters are determined by

$$\begin{aligned} k_p(k+1) &= k_p(k) - \eta_p \frac{\partial J_2(k)}{\partial k_p(k)} \\ &= k_p(k) - \eta_p \frac{\partial J_2(k)}{\partial \hat{y}(k)} \frac{\partial \hat{y}(k)}{\partial u(k)} \frac{\partial u(k)}{\partial k_p(k)} \\ &= k_p(k) + \eta_p e_2(k) \frac{\partial \hat{y}(k)}{\partial u(k)} e(k), \end{aligned} \quad (20)$$

$$\begin{aligned}
k_i(k+1) &= k_i(k) - \eta_i \frac{\partial J_2(k)}{\partial k_i(k)} \\
&= k_i(k) - \eta_i \frac{\partial J_2(k)}{\partial \hat{y}(k)} \frac{\partial \hat{y}(k)}{\partial u(k)} \frac{\partial u(k)}{\partial k_i(k)} \\
&= k_i(k) + \eta_i e_2(k) \frac{\partial \hat{y}(k)}{\partial u(k)} \Sigma_k^{\lambda(k)} e(k),
\end{aligned} \tag{21}$$

$$\begin{aligned}
k_d(k+1) &= k_d(k) - \eta_d \frac{\partial J_2(k)}{\partial k_d(k)} \\
&= k_d(k) - \eta_d \frac{\partial J_2(k)}{\partial \hat{y}(k)} \frac{\partial \hat{y}(k)}{\partial u(k)} \frac{\partial u(k)}{\partial k_d(k)} \\
&= k_d(k) + \eta_d e_2(k) \frac{\partial \hat{y}(k)}{\partial u(k)} {}_H \Delta_k^{\mu(k)} e(k),
\end{aligned} \tag{22}$$

where  $\eta_k$ ,  $\eta_i$ , and  $\eta_d$  are the learning rates for the parameters  $k_p$ ,  $k_i$ , and  $k_d$ . Consider that

$$\frac{\partial u(k)}{\partial \lambda(k)} = k_i(k) (\lambda^2(k) - \lambda(k) + k) k^{\lambda(k)-2} e(k),$$

$$\frac{\partial u(k)}{\partial \mu(k)} = k_d(k) (-k\mu^{-2}(k) + \mu^{-1}(k) - 1) k^{-\mu(k)} e(k);$$

then the tuning formulas of orders are also determined by

$$\begin{aligned}
&\lambda(k+1) \\
&= \lambda(k) - \eta_\lambda \frac{\partial J_2(k)}{\partial \lambda(k)} \\
&= \lambda(k) - \eta_\lambda \frac{\partial J_2(k)}{\partial \hat{y}(k)} \frac{\partial \hat{y}(k)}{\partial u(k)} \frac{\partial u(k)}{\partial \lambda(k)} \\
&= \lambda(k) + \eta_\lambda e_2(k) \frac{\partial \hat{y}(k)}{\partial u(k)} k_i(k) (\lambda^2(k) - \lambda(k) + k) k^{\lambda(k)-2} e(k),
\end{aligned} \tag{23}$$

$$\begin{aligned}
&\mu(k+1) \\
&= \mu(k) - \eta_\mu \frac{\partial J_2(k)}{\partial \mu(k)} \\
&= \mu(k) - \eta_\mu \frac{\partial J_2(k)}{\partial \hat{y}(k)} \frac{\partial \hat{y}(k)}{\partial u(k)} \frac{\partial u(k)}{\partial \mu(k)} \\
&= \mu(k) + \eta_\mu e_2(k) \frac{\partial \hat{y}(k)}{\partial u(k)} k_d(k) (-k\mu^{-2}(k) + \mu^{-1}(k) - 1) k^{-\mu(k)} e(k),
\end{aligned} \tag{24}$$

where  $\eta_\lambda$  and  $\eta_\mu$  are the learning rates for the orders  $\lambda$  and  $\mu$ .

There is the function term  $\hat{y}(k)/\partial u(k)$  in the above updating formulas of the parameters and orders; hence, the calculation method of  $\hat{y}(k)/\partial u(k)$  is as follows:

$$\frac{\partial \hat{y}(k)}{\partial u(k)} = W^{(2)}(k) \mathcal{F}^{(1)}(k) \overline{W}^{(1)}(k),$$

where

$$\overline{W}^{(1)}(k) = [w_{1,n_y+1}^{(1)}(k), w_{2,n_y+1}^{(1)}(k), \dots, w_{S_1, n_y+1}^{(1)}(k)]^T.$$

Therefore, the updating formulas of parameters and orders of NNDTFO PID controller (19) are determined by

$$k_p(k+1) = k_p(k) + \eta_p e_2(k) W^{(2)}(k) \mathcal{F}^{(1)}(k) \overline{W}^{(1)}(k) e(k), \tag{25}$$

$$k_i(k+1) = k_i(k) + \eta_i e_2(k) W^{(2)}(k) \mathcal{F}^{(1)}(k) \overline{W}^{(1)}(k) \Sigma_k^{\lambda(k)} e(k), \quad (26)$$

$$k_d(k+1) = k_d(k) + \eta_d e_2(k) W^{(2)}(k) \mathcal{F}^{(1)}(k) \overline{W}^{(1)}(k) {}^H \Delta_k^{\mu(k)} e(k), \quad (27)$$

$$\lambda(k+1) = \lambda(k) + \eta_\lambda e_2(k) W^{(2)}(k) \mathcal{F}^{(1)}(k) \overline{W}^{(1)}(k) k_i(k) (\lambda^2(k) - \lambda(k) + k) k^{\lambda(k)-2} e(k), \quad (28)$$

$$\mu(k+1) = \mu(k) + \eta_\mu e_2(k) W^{(2)}(k) \mathcal{F}^{(1)}(k) \overline{W}^{(1)}(k) k_d(k) (-k\mu^{-2}(k) + \mu^{-1}(k) - 1) k^{-\mu(k)} e(k). \quad (29)$$

Considering the effective selection intervals of parameters and orders in the controller (19), we restrict the optimization results of parameters and orders as  $k_p(k) = 0$ ,  $k_i(k) = 0$ ,  $k_d(k) = 0$ ,  $\lambda(k) = 0$ , or  $\mu(k) = 0$  if  $k_p(k) < 0$ ,  $k_i(k) < 0$ ,  $k_d(k) < 0$ ,  $\lambda(k) < 0$ , or  $\mu(k) < 0$  occurs; and  $\lambda(k) = 2$  or  $\mu(k) = 2$  if  $\lambda(k) > 2$  or  $\mu(k) > 2$  occurs.

### 3.3. Tuning Method of Parameters and Orders Based on the Borges Derivative

In the updating Formulas (45)–(49), the updating methods of parameters  $k_p(k)$ ,  $k_i(k)$ , and  $k_d(k)$  and orders  $\lambda(k)$  and  $\mu(k)$  depend on the information of the first order negative gradient. We investigate the updating method based on the improved Borges gradient to improve the optimizations of parameters and fractional-orders in NNTDFO PID controller (19).

The Borges derivative is provided in [35] as follows.

$${}^B D_x^q f(x) = \lim_{y \rightarrow x} \frac{f(x) - f(y)}{x \ominus_q y}, \quad (30)$$

where  $x \ominus_q y = (x - y) / (1 + (1 - q)y)$  with  $y \neq 1 / (q - 1)$  and  $q > 0$ .

If the function  $f(x)$  is differentiable, we have

$${}^B D_x^q f(x) = (1 + (1 - q)x) \frac{df(x)}{dx}. \quad (31)$$

If we set  $q = 1$  in (31), the Borges derivative is equivalent to the first-order derivative. Therefore, we use the Borges gradient consisting of the Borges derivatives to generalize the updating Equations (45)–(49). In addition, the Borges derivative is a local operator, which satisfies the chain rule; hence, the training of parameters and orders of the controller (19) based on the neural network algorithm can be applied. Let  $z = m(v)$  and  $v = n(x)$ ; then the Borges derivative of  $z = m(n(x))$  is obtained as

$${}^B D_{xz}^q z(x) = (1 + (1 - q)x) \frac{dm(v)}{dv} \Big|_{v=n(x)} \frac{dn(x)}{dx}. \quad (32)$$

Based on (32), we have

$${}^B D_{k_p(k)}^{q_p(k)} J_2(k) = -[1 + (1 - q_p(k))k_p(k)] e_2(k) \frac{\partial \hat{y}(k)}{\partial u(k)} e(k), \quad (33)$$

$${}^B D_{k_i(k)}^{q_i(k)} J_2(k) = -[1 + (1 - q_i(k))k_i(k)] e_2(k) \frac{\partial \hat{y}(k)}{\partial u(k)} \Sigma_k^{\lambda(k)} e(k), \quad (34)$$

$${}^B D_{k_d(k)}^{q_d(k)} J_2(k) = -[1 + (1 - q_d(k))k_d(k)] e_2(k) \frac{\partial \hat{y}(k)}{\partial u(k)} {}^H \Delta_k^{\mu(k)} e(k), \quad (35)$$

$${}^B D_{\lambda(k)}^{q_\lambda(k)} J_2(k) = -[1 + (1 - q_\lambda(k))\lambda(k)] e_2(k) \frac{\partial \hat{y}(k)}{\partial u(k)} k_i(k) (\lambda^2(k) - \lambda(k) + k) k^{\lambda(k)-2} e(k), \quad (36)$$

$${}^B D_{\mu(k)}^{q_\mu(k)} J_2(k) = -[1 + (1 - q_\mu(k))\mu(k)] e_2(k) \frac{\partial \hat{y}(k)}{\partial u(k)} k_d(k) (-k\mu^{-2}(k) + \mu^{-1}(k) - 1) k^{-\mu(k)} e(k), \quad (37)$$

where  $q_p$ ,  $q_i$ ,  $q_d$ ,  $q_\lambda$ , and  $q_\mu$  are orders which are positive real numbers.

In order to give the adjustment methods of these orders, we take the order of the updating formula of the proportional coefficient as an example, and the relative speed is defined as

$$L(k_p(k)) = \frac{{}^B D_{k_p(k)}^{q_p(k)} J_2(k)}{{}^B D_{k_p(k)}^1 J_2(k)} = 1 + (1 - q_p(k))k_p(k). \tag{38}$$

The relationship between the index  $L(k_p(k))$  and  $k_p(k)$  is shown in Figure 4.

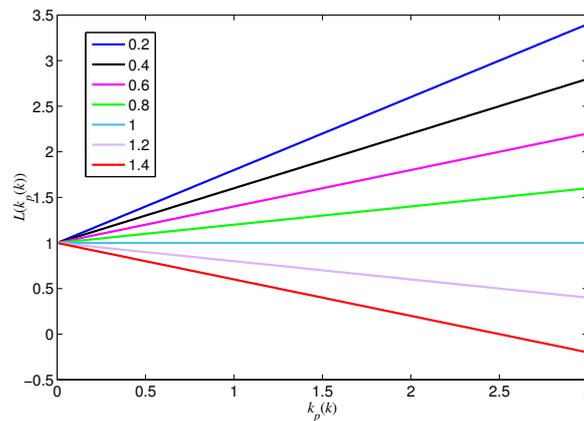


Figure 4. Curves of  $L(k_p(k))$  with different orders  $q_{k_p}(k)$ .

It is obvious from Figure 4 that the Borges derivative of  $J_2(k)$  with respect to  $k_p(k)$  is larger if the order  $q_{k_p}$  is set to be smaller. If  $q_{k_p}$  is larger than a certain value, the sign of derivative changes, and this is an undesirable result. With the increase of parameter  $k_p(k)$ , the index  $L(k_p(k))$  corresponding to a smaller order becomes larger, while the index  $L(k_p(k))$  corresponding to a larger order becomes smaller, and it may be less than zero.

Let  $q_p(k) = 0$ ; then we have  $L(k_p(k)) = 1 + k_p(k)$ , which is the maximum of  $L(k_p(k))$ . Thereby, the range of  $L(k_p(k))$  is  $(0, 1 + k_p(k)]$ . If the indexes  $L(k_p(k))$  corresponding to different parameters  $k_p(k)$  are the same as  $L_m \in (0, 1 + k_p(k)]$ , we have

$$q_p(k) = 1 - \frac{L_m - 1}{k_p(k)}. \tag{39}$$

We hope that the Borges derivative is less than the first-order derivative to accelerate the optimization at the beginning of parameter optimization. Meanwhile, the Borges derivative is greater than the first-order derivative to improve the accuracy of parameter optimization at the end of the optimization. Then, the order  $q_p(k)$  is determined by

$$q_p(k) = \begin{cases} 1 - \frac{L_2^p - 1}{k_p(k)} & \frac{k}{M} < \tilde{r}_p \\ 1 - \frac{L_1^p - 1}{k_p(k)} & \frac{k}{M} \geq \tilde{r}_p \end{cases}, \tag{40}$$

where  $L_1^p$  and  $L_2^p$  satisfy  $0 < L_1^p < 1 < L_2^p \leq 1 + k_p(k)$ ;  $\tilde{r}_p$  is a random number uniformly distributed between 0 and 1; and  $M$  is the number of iterations.

Similarly, the tuning methods of other fractional-orders are offered as follows:

$$q_i(k) = \begin{cases} 1 - \frac{L_2^i - 1}{k_i(k)} & \frac{k}{M} < \tilde{r}_i \\ 1 - \frac{L_1^i - 1}{k_i(k)} & \frac{k}{M} \geq \tilde{r}_i \end{cases}, \tag{41}$$

$$q_d(k) = \begin{cases} 1 - \frac{L_2^d - 1}{k_d(k)} \frac{k}{M} < \tilde{r}_d \\ 1 - \frac{L_1^d - 1}{k_d(k)} \frac{k}{M} \geq \tilde{r}_d \end{cases}, \quad (42)$$

$$q_\lambda(k) = \begin{cases} 1 - \frac{L_2^\lambda - 1}{k_\lambda(k)} \frac{k}{M} < \tilde{r}_\lambda \\ 1 - \frac{L_1^\lambda - 1}{k_\lambda(k)} \frac{k}{M} \geq \tilde{r}_\lambda \end{cases}, \quad (43)$$

$$q_\mu(k) = \begin{cases} 1 - \frac{L_2^\mu - 1}{k_\mu(k)} \frac{k}{M} < \tilde{r}_\mu \\ 1 - \frac{L_1^\mu - 1}{k_\mu(k)} \frac{k}{M} \geq \tilde{r}_\mu \end{cases}, \quad (44)$$

where  $\tilde{r}_i$ ,  $\tilde{r}_d$ ,  $\tilde{r}_\lambda$ , or  $\tilde{r}_\mu$  is a random number uniformly distributed between 0 and 1; and the conditions  $0 < L_1^i < 1 < L_2^i \leq 1 + k_i(k)$ ,  $0 < L_1^d < 1 < L_2^d \leq 1 + k_d(k)$ ,  $0 < L_1^\lambda < 1 < L_2^\lambda \leq 1 + \lambda(k)$ , and  $0 < L_1^\mu < 1 < L_2^\mu \leq 1 + \mu(k)$  are also required.

Based on the Equations (33)–(37), the updating formulas of parameters and orders in the controller (19) based on the Borges derivative are offered as follows.

$$k_p(k+1) = k_p(k) + \eta_p[1 + (1 - q_p(k))k_p(k)]e_2(k)(k)W^{(2)}(k)\mathcal{F}^{(1)}(k)\overline{W}^{(1)}(k)e(k), \quad (45)$$

$$k_i(k+1) = k_i(k) + \eta_i[1 + (1 - q_i(k))k_i(k)]e_2(k)(k)W^{(2)}(k)\mathcal{F}^{(1)}(k)\overline{W}^{(1)}(k)\Sigma_k^{\lambda(k)}e(k), \quad (46)$$

$$k_d(k+1) = k_d(k) + \eta_d[1 + (1 - q_d(k))k_d(k)]e_2(k)(k)W^{(2)}(k)\mathcal{F}^{(1)}(k)\overline{W}^{(1)}(k)^H\Delta_k^{\mu(k)}e(k), \quad (47)$$

$$\begin{aligned} &\lambda(k+1) \\ &= \lambda(k) + \eta_\lambda[1 + (1 - q_\lambda(k))\lambda(k)]e_2(k)(k)W^{(2)}(k)\mathcal{F}^{(1)}(k)\overline{W}^{(1)}(k) \times \\ &\quad k_i(k)(\lambda^2(k) - \lambda(k) + k)k^{\lambda(k)-2}e(k), \end{aligned} \quad (48)$$

$$\begin{aligned} &\mu(k+1) \\ &= \mu(k) + \eta_\mu[1 + (1 - q_\mu(k))\mu(k)]e_2(k)W^{(2)}(k)\mathcal{F}^{(1)}(k)\overline{W}^{(1)}(k) \times \\ &\quad k_d(k)(-k\mu^{-2}(k) + \mu^{-1}(k) - 1)k^{-\mu(k)}e(k). \end{aligned} \quad (49)$$

## 4. Illustrative Examples

### 4.1. Example 1

Consider the following discrete-time nonlinear controlled plant as

$$y(k) = 0.3y(k-1) - 0.1\sin(y(k-1)) + u(k-1). \quad (50)$$

The reference signal  $r(k)$  is chosen as  $r(k) = 0.2 + 0.1\sin(0.02k)$ , and the initial value  $y(0) = 0$  is set. The NNI is used to identified the model of the nonlinear plant and the activation function in the hidden layer of NNI is selected as

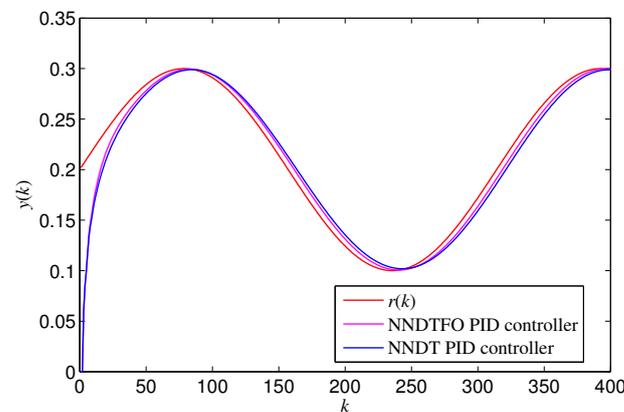
$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}}. \quad (51)$$

In the NNI, the number of the neurons in the hidden layer is  $S_1 = 6$ , and  $N = 2$  is obtained with  $x(k) = [y(k), u(k)]^T$  and  $L = 500$ . To verify the effectiveness of the proposed

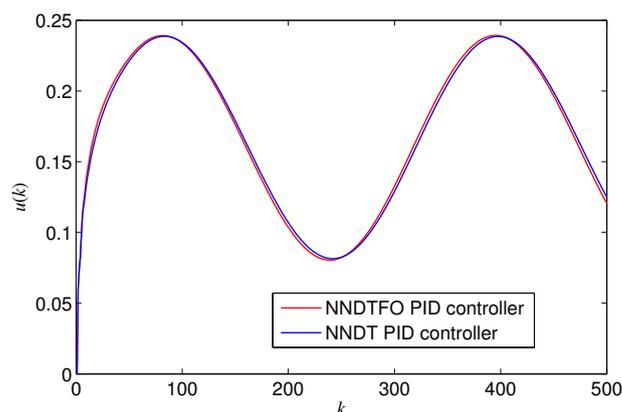
NNDTFO PID controller, the results were compared with those of a neural network discrete-time (NNDT) PID controller. The learning rates in NNI were both set as  $\zeta = 0.5$  for the two types of controller. The learning rates of the parameters in two types of controllers were both set as  $\eta_p = 5000$ ,  $\eta_i = 10$ , and  $\eta_d = 10$ , and the learning rates of the orders were set as  $\eta_\lambda = 10,000$  and  $\eta_\mu = 10,000$ . The initial value for weights and biases in NNI was chosen as 0.01, and the weight in the NNDTFO PID controller and NNDT PID controller was set as 0.1;  $u(0) = 0$  was used. Meanwhile, the initial value of the orders  $\lambda$  and  $\mu$  was set as 1 in the NNDTFO PID controller.

The responses of the outputs  $y(k)$  via two types of controllers, and the responses of two types of controllers  $u(k)$ , are shown in Figures 5 and 6, respectively.

It can be seen from the output responses via the different controllers described in Figure 5 that the NNDTFO PID controller with Hausdorff difference and Hausdorff sum can obtain a more accurate response compared with the output response via the NNDT PID controller. From the curves of the two types of PID controllers described in Figure 6, it can be seen that the outputs of the two types of PID controller are bounded with no significant control effects.



**Figure 5.** Responses of outputs  $y(k)$  for the closed loop system of example 1 via two types of controller.



**Figure 6.** Responses of controllers  $u(k)$  for the closed loop system of example 1 with a sinusoidal signal.

In order to show that Hausdorff sum and difference operations can reduce the response error, the index function is defined as

$$E = \sum_{l=1}^L e^2(l). \quad (52)$$

While letting  $r(k) = 0.2 + A \sin(\omega k)$ , the parameters were set as  $\omega = 0.01, 0.02, \dots, 0.5$  with  $A = 0.2$ , and  $A = 0.1, 0.15, \dots, 0.3$  with  $\omega = 0.02$  respectively; then the corresponding indexes  $E$  are offered in Tables 1 and 2.

**Table 1.** Index  $E$  for different parameters  $\omega$  with  $A = 0.1$  via two types of controller.

$\omega$	$E$ via NNDTFO PID Controller	$E$ via NNDT PID Controller
0.01	0.0827	0.0895
0.02	0.0943	0.1118
0.03	0.1066	0.1395
0.04	0.1204	0.1763
0.05	0.1318	0.2044

**Table 2.** Index  $E$  for different parameters  $A$  with  $\omega = 0.02$  via two types of controller.

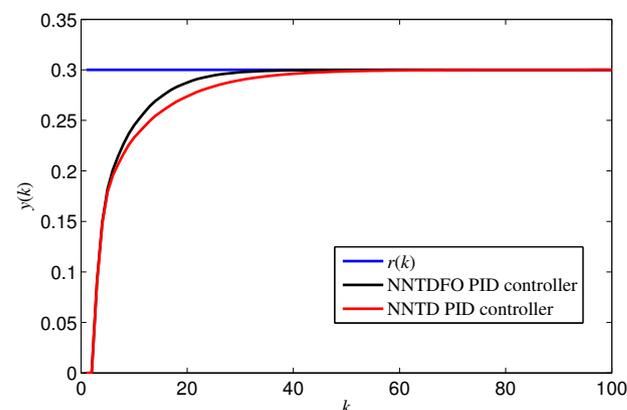
$A$	$E$ via NNDTFO PID Controller	$E$ via NNDT PID Controller
0.10	0.0943	0.1118
0.15	0.1074	0.1434
0.20	0.1204	0.1832
0.25	0.1320	0.2295
0.30	0.1417	0.2810

It can be seen from the data in Tables 1 and 2 that the response error of system output can be reduced by introducing the Hausdorff difference and Hausdorff sum for different reference signals. These results show that the proposed NNDTFO PID controller can further improve the accuracy of the output response.

To illustrate the improvement of the proposed controller in regard to the dynamic response of the controlled plant, the reference input signal was changed to  $r(k) = 0.3$  with  $L = 100$ .

From the experimental results in Figure 7, it is obvious that the controller with the Hausdorff sum and difference operations can make the response of the output  $y(k)$  faster, which is another advantage of the control strategy proposed in this paper. Besides, the parameters and orders in the NNDTFO PID controller are shown in Figure 8 respectively.

From the curves of control parameters and orders in Figure 8, it can be seen that parameters  $k_p(k)$ ,  $k_i(k)$ , and  $k_d(k)$ , and orders  $\lambda(k)$  and  $\mu(k)$  converge to the fixed values quickly. The orders  $\lambda(k)$  and  $\mu(k)$  in the proposed controller are greater than 1 or less than 1 for the reference signal  $r(k) = 0.3$ , and this phenomenon corresponds to the analysis of orders in Section 3.



**Figure 7.** Responses of outputs  $y(k)$  closed loop system for example 1 with a constant signal.

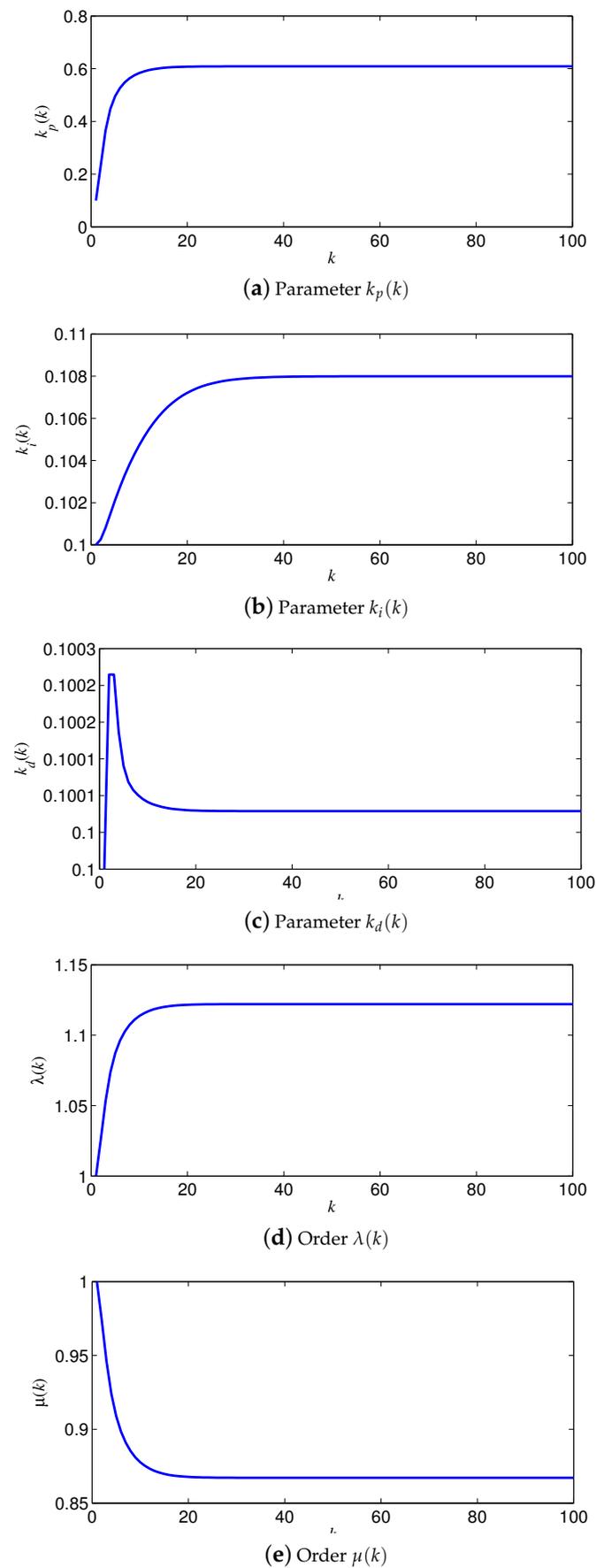


Figure 8. Parameters and orders.

#### 4.2. Example 2

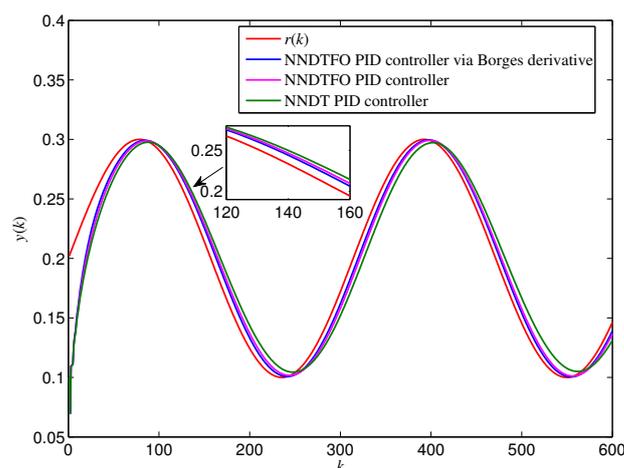
In this example, the Borges derivative is focused on to improve the control performance based on the NNTD PID controller. The nonlinear controlled plant used was

$$y(k) = -0.3y(k-1) + 0.1 \cos(y(k-1)) + u(k-1). \quad (53)$$

The parameters of the NNI and NNDTFO PID controller were consistent with the corresponding ones in example 1. Other parameters were set as  $L_1^p = L_1^i = L_1^d = L_1^\lambda = L_1^\mu = 0.2$  and  $L_2^p = 0.9 + k_p(k)$ ,  $L_2^i = 0.9 + k_i(k)$ ,  $L_2^d = 0.9 + k_d(k)$ ,  $L_2^\lambda = 0.9 + \lambda(k)$ , and  $L_2^\mu = 0.9 + \mu(k)$ . In consideration of the existence of random functions with the use of the Borges derivative to optimize the parameters and orders of the controller, the average result of 100 experiments was used for the evaluation in this case. The reference signal was set as  $r(k) = 0.2 + 0.1 \sin(0.02k)$  with  $L = 600$ ; then the response of the average value of output  $y(k)$  via the NNDTFO PID controller with Borges derivative, and the responses of outputs  $y(k)$  via the NNDTFO PID controller and NNDT PID controller were plotted in Figure 9. The corresponding errors based on (52) via the NNDTFO PID controller via Borges derivative, the NNDTFO PID controller, and the NNDT PID controller were 0.0729 (the average result for 100 times), 0.0892, 0.1296 respectively.

It can be seen from the experimental results in Figure 9 that the NNDTFO PID controller with Borges derivative achieved the best control effect in three types of PID controller. By using Borges derivative, the output of the system can track the reference signal well; the tracking errors via the three controllers were the smallest. This phenomenon illustrates that the introduction of the Borges derivative can improve the optimization effect of parameters and orders in NNTDFO controller. For this example, we can draw the conclusion that the use of the Hausdorff difference and Hausdorff sum can improve the control accuracy. On this basis, the control accuracy can be further improved by introducing the Borges derivative. It was verified that the control effect of the NNTD controller can be improved by introducing fractal derivatives into two areas, i.e., the controller structure and the optimization strategy.

Figure 10 shows the average values of the relative speeds  $u$  corresponding to the parameters and orders of the NNTD PID controller with the Borges derivative from 100 experiments.



**Figure 9.** Responses of outputs  $y(k)$  via three types of PID controller of a closed loop system for example 2 with a sinusoidal signal.

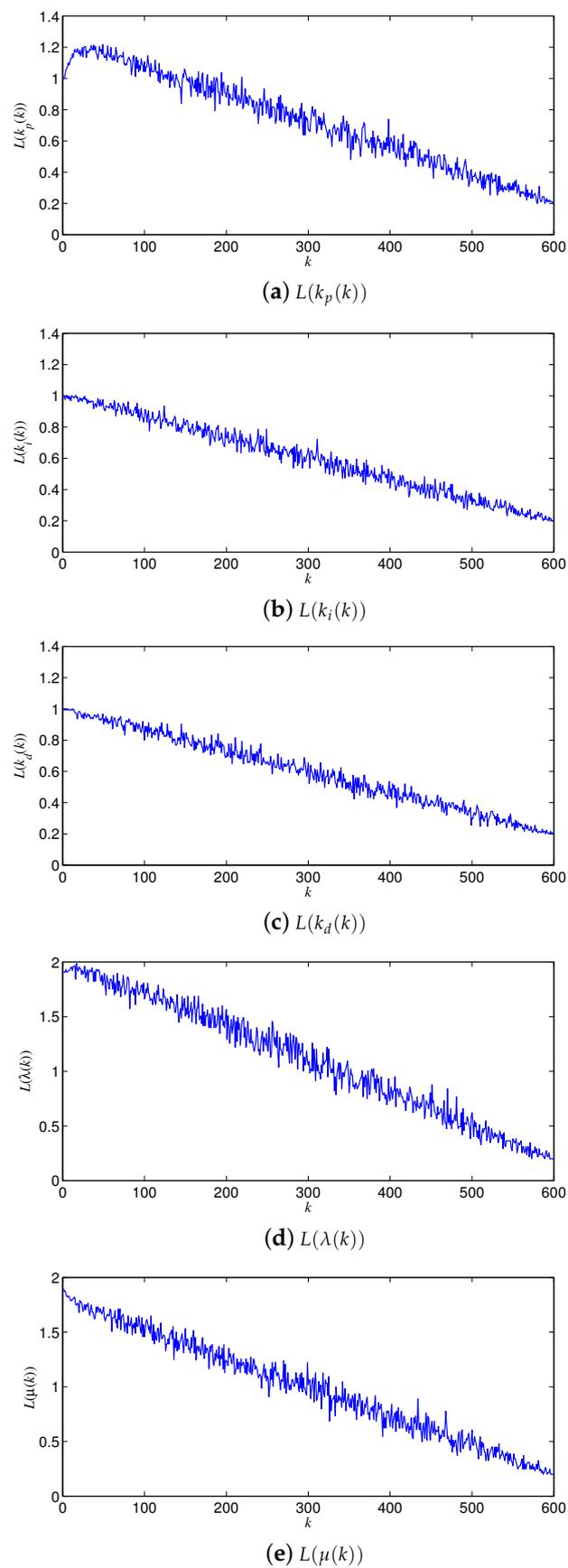


Figure 10. Relative average speeds of parameters and orders.

It can be seen from the average values of relative speeds corresponding to different parameters and orders described in Figure 10 that all of the average relative speeds can be gradually reduced from a number greater than 1 to a number less than 1, by adopting the order adjustment strategy proposed in this paper. According to this strategy, the optimization speed can be accelerated at the beginning of optimization stage for parameters and orders in the NNTDFO PID controller, and the optimization precision and effect can be improved at the end of the optimization stage.

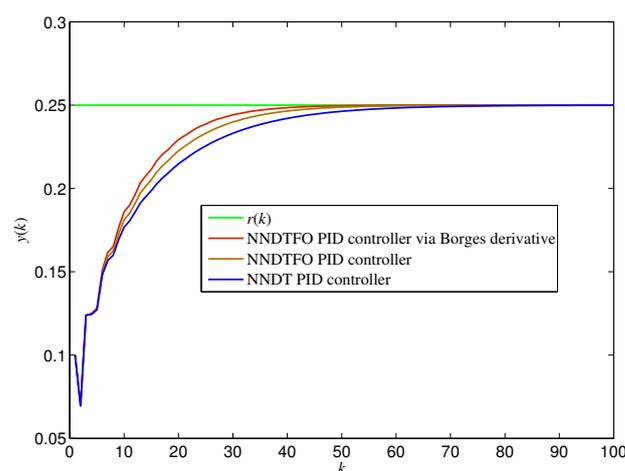
To compare the robustness of the three types of controllers, the coefficient 0.1 before the nonlinear function term  $\cos(y(k-1))$  in the model was changed to  $0.1 + B \sin(0.5k)$ , and the error indexes for different values of parameter  $B$  are shown in Table 3. (The index of the NNTDFO PID controller is the average result of 100 rounds).

**Table 3.** Index  $E$  for different parameters  $B$  via three types of controller.

$B$	$E$ via NNTDFO PID Controller via Borges Derivative	$E$ via NNTDFO PID Controller	$E$ via NNDT PID Controller
0.002	0.0731	0.0894	0.1297
0.004	0.0737	0.0900	0.1303
0.006	0.0749	0.0911	0.1313
0.008	0.0765	0.0927	0.1328
0.010	0.0786	0.0947	0.1347

The experimental results in Table 3 show that the three controllers based on neural networks have robustness and can adapt to disturbances in parameters. In addition, the fractal derivative can reduce the control error  $E$  to improve the control accuracy of the closed-loop system even, if parameter 0.1 is disturbed.

In order to investigate the response of the Borges derivative to the convergence speed of the control effect, the reference signal was set to  $r(k) = 0.25$ , and  $L = 100$  was set for this case; other parameters in three types of controllers were not changed. Then, from the average value of output  $y(k)$  via the NNTDFO PID controller with Borges derivative, the responses of outputs  $y(k)$  via NNTDFO PID controller and NNDT PID controller are drawn in Figure 11.



**Figure 11.** Responses of outputs  $y(k)$  via three types of PID controller using the closed loop system of example 2 with a constant signal.

The control effect of the three controllers described in Figure 11 shows that the NNTDFO PID controller with the Borges derivative achieved the fastest convergence speed, followed by the NNTDFO PID controller without the Borges derivative, and the

NNTD PID controller without the fractal derivative had the slowest convergence speed. It is shown that the Borges derivative can further accelerate the response of the controlled plant.

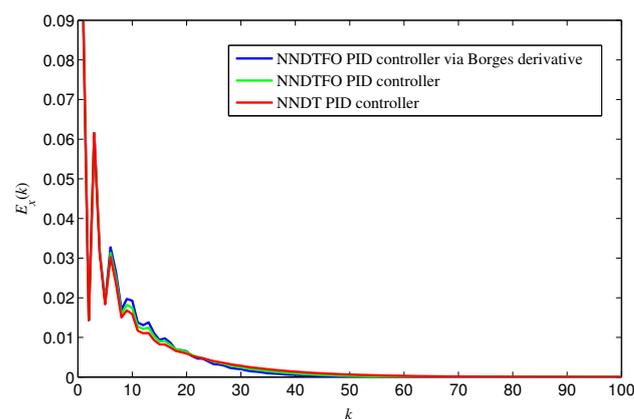
Next, we study the error index for a differing constant reference signal, and the results are given in Table 4.

**Table 4.** Index  $E$  for constant reference signal via three types of controller.

$r(k)$	$E$ via NNDTFO PID Controller via Borges Derivative	$E$ via NNDTFO PID Controller	$E$ via NNDT PID Controller
0.025	0.0766	0.0812	0.0877
0.020	0.0432	0.0447	0.0468
0.015	0.0166	0.0169	0.0172

From the experimental results described in Table 4, it can be seen that the rule of the control effect is consistent for constant input signals of different values. In other words, the use of a fractal derivative to optimize controller parameters and the use of fractal difference and fractal sum can improve control effect.

To verify the fact that the designed NNI can identify the system model well based on three kinds of neural network PID controllers, the errors between the output of NNI  $\hat{y}(k)$  and the output  $y(k)$  as  $E_x(k) = |\hat{y}(k) - y(k)|$  for  $L = 100$  and  $r(k) = 2.5$  are shown in Figure 12.



**Figure 12.** Errors  $E_x(k)$  for three types of neural network PID controller.

It can be seen from the error curves in Figure 12 that NNIs are not affected by the three types of neural network PID controllers, and NNIs can identify the dynamic characteristics of the controlled plant effectively, and the identification error converges to zero quickly.

## 5. Conclusions

This paper mainly studied the applications of two kinds of fractal derivative in the design of a neural network-based PID controller. Based on the definition of the Hausdorff derivative and Hausdorff integral, the calculation formulas of Hausdorff difference and Hausdorff sum were given to construct NNTDFO PID controllers. By using the local operation characteristics of Hausdorff difference and Hausdorff sum, the tuning method of parameters and orders in the proposed NNTDFO PID controllers were given. The orders corresponding to the Hausdorff difference and Hausdorff sum were introduced to make the controller design more flexible, and the adaptive adjustment algorithm of orders is also provided in this paper.

In addition, the learning rules of parameters and orders were further discussed. The first derivative was replaced by the Borges derivative; then the parameters and orders with the adaptive relative velocity were adjusted to improve the optimization performance. The adaptive tuning rules of the orders in the Borges derivatives were given to gain a more

satisfactory control effect via NNTDFO PID controllers. The use of a fractal derivative, like the integer derivative, satisfies the chain rule, which provides the basis for the training parameters of neural networks. The applications of fractal difference and fractal sum derived from the fractal derivative and integral can avoid the derivation of the Gamma function, which is convenient for calculations and can also increase the flexibility of controller design. In this paper, two kinds of fractional-order PID controller based on fractal derivative and derived fractal difference and fractal sum were proposed, which can improve the response speed and accuracy of the closed-loop control system. Compared with the integer-order PID algorithm, the calculation is more complex and the calculation time is longer. However, in view of the fact that many controllers are operated by computers, the calculation burden of the proposed algorithm is not large. In future research, the basic neural network learning and construction method based on fractal derivatives will be extended to a more complex neural network model and neural network control algorithm.

**Funding:** This research was funded by Liaoning Revitalization Talents Program under grant number XLYC1807229, the Natural Science Foundation of Liaoning Province, China under grant number 20180520009, the China Postdoctoral Science Foundation Funded Project under grant number 2019M651206, and the Scientific Research Fund of Liaoning Provincial Education Department, China under grant LJC202010.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The author declares no conflict of interest.

## References

1. Shah, P.; Agashe, S. Review of fractional PID controller. *Mechatronics* **2016**, *38*, 29–41. [[CrossRef](#)]
2. Tejado, T.; Vinagre, B.M.; Traver, J.E.; Prieto-Arranz, J.; Nuevo-Gallardo, C. Back to basics: Meaning of the parameters of fractional order PID controllers. *Mathematics* **2019**, *7*, 530. [[CrossRef](#)]
3. Haroun, A.H.G.; Li, Y.Y. A novel optimized fractional-order hybrid fuzzy intelligent PID controller for interconnected realistic power systems. *Trans. Inst. Meas. Control* **2019**, *41*, 3065–3080. [[CrossRef](#)]
4. Hamidian, H.; Beheshti, M.T.H. A robust fractional-order PID controller design based on active queue management for TCP network. *Int. J. Syst. Sci.* **2018**, *49*, 211–216. [[CrossRef](#)]
5. Viola, J.; Angel, L.; Sebastian, J.M. Design and robust performance evaluation of a fractional order PID controller applied to a DC motor. *IEEE-CAA J. Autom. Sin.* **2017**, *4*, 304–314. [[CrossRef](#)]
6. Yang, B.; Wang, J.B.; Wang, J.T.; Shu, H.C.; Li, D.Y.; Zeng, C.Y.; Chen, Y.J.; Zhang, X.S.; Yu, T. Robust fractional-order PID control of supercapacitor energy storage systems for distribution network applications: A perturbation compensation based approach. *J. Clean. Prod.* **2021**, *279*, 123362. [[CrossRef](#)]
7. Kumar, J.; Kumar, V.; Rana, K.P.S. Fractional-order self-tuned fuzzy PID controller for three-link robotic manipulator system. *Neural Comput. Appl.* **2020**, *32*, 7235–7257. [[CrossRef](#)]
8. Ray, P.K.; Paital, S.R.; Mohanty, A.; Foo, Y.S.E.; Krishnan, A.; Gooi, H.B.; Amaratunga, G.A.J. A hybrid firefly-swarm optimized fractional order interval type-2 fuzzy PID-pss for transient stability improvement. *IEEE Trans. Ind. Appl.* **2019**, *55*, 6486–6498. [[CrossRef](#)]
9. Gao, P.; Zhang, G.M.; Ouyang, H.M.; Mei, L. An adaptive super twisting nonlinear fractional order PID sliding mode control of permanent magnet synchronous motor speed regulation system based on extended state observer. *IEEE Access* **2020**, *8*, 53498–53510. [[CrossRef](#)]
10. Chevalier, A.; Francis, C.; Copot, C.; Ionescu, C.M.; Keyser, R.D. Fractional-order PID design: Toward transition from state-of-art to state-of-use. *ISA Trans.* **2019**, *84*, 178–186. [[CrossRef](#)] [[PubMed](#)]
11. Mandic, P.D.; Sekara, T.B.; Lazarevic, M.P.; Boskovic, M. Dominant pole placement with fractional order PID controllers: D-decomposition approach. *ISA Trans.* **2017**, *67*, 76–86. [[CrossRef](#)]
12. Feliu-Batlle, V.; Castillo-Garcia, F.J. On the robust control of stable minimum phase plants with large uncertainty in a time constant. A fractional-order control approach. *Automatica* **2014**, *50*, 218–224. [[CrossRef](#)]
13. Luo, Y.; Chen, Y.Q.; Wang, C.Y.; Pi, Y.G. Tuning fractional order proportional integral controllers for fractional order systems. *J. Process Control* **2010**, *20*, 823–831. [[CrossRef](#)]
14. Machado, J.A.T. Optimal tuning of fractional controllers using genetic algorithms. *Nonlinear Dyn.* **2010**, *62*, 447–452.
15. Hekimoglu, B. Optimal tuning of fractional order PID controller for DC motor speed control via chaotic atom search optimization algorithm. *IEEE Access* **2019**, *7*, 38100–38114. [[CrossRef](#)]
16. Liu, L.; Shan, L.; Dai, Y.W.; Liu, C.L.; Qi, Z.D. Improved quantum bacterial foraging algorithm for tuning parameters of fractional-order PID controller. *J. Syst. Eng. Electron.* **2018**, *29*, 166–175. [[CrossRef](#)]

17. Zhao, J.B.; Jing, W.C.; Wang, J.Z. An indirect optimization scheme for tuning a fractional order PI controller using extremum seeking. *Mechatronics* **2018**, *56*, 146–156. [[CrossRef](#)]
18. Liu, X.Y. Optimization design on fractional order PID controller based on adaptive particle swarm optimization algorithm. *Nonlinear Dyn.* **2016**, *84*, 379–386. [[CrossRef](#)]
19. Asgharnia, A.; Jamali, A.; Shahnazi, R.; Maheri, A. Load mitigation of a class of 5-MW wind turbine with RBF neural network based fractional-order PID controller. *ISA Trans.* **2020**, *96*, 272–286. [[CrossRef](#)]
20. Chen, C.Y.; Zhu, S.; Wei, Y.C.; Yang, C.Y. Finite-time stability of delayed memristor-based fractional-order neural networks. *IEEE Trans. Cybern.* **2020**, *50*, 1607–1616. [[CrossRef](#)]
21. Lv, Y.J.; Hu, C.; Yu, J.; Jiang, H.J.; Huang, T.W. Edge-based fractional-order adaptive strategies for synchronization of fractional-order coupled networks with reaction–diffusion terms. *IEEE Trans. Cybern.* **2020**, *50*, 1582–1594. [[CrossRef](#)] [[PubMed](#)]
22. Zhang, F.H.; Zeng, Z.G. Multiple mittag-leffler stability of delayed fractional-order Cohen–Grossberg neural networks via mixed monotone operator pair. *IEEE Trans. Cybern.* **2020**, 31995512. [[CrossRef](#)]
23. Liu, P.; Zeng, Z.G.; Wang, J. Global synchronization of coupled fractional-order recurrent neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 2358–2368. [[CrossRef](#)] [[PubMed](#)]
24. Wang, F.X.; Liu, X.G.; Tang, M.L.; Chen, L.F. Further results on stability and synchronization of fractional-order Hopfield neural networks. *Neurocomputing* **2019**, *346*, 12–19. [[CrossRef](#)]
25. Krishna, B.T. Studies on fractional order differentiators and integrators: A survey. *Signal Process.* **2011**, *91*, 386–426. [[CrossRef](#)]
26. El-Khazali, R. Discretization of fractional-order differentiators and integrators. *IFAC Proc. Vol.* **2014**, *47*, 2016–2021. [[CrossRef](#)]
27. Sierociuk, D.; Dzielinski, A. Fractional Kalman filter algorithm for the states, parameters and order of fractional system estimation. *Int. J. Appl. Math. Comput. Sci.* **2006**, *16*, 129–140.
28. Batir, N. On some properties of the Gamma function. *Expo. Math.* **2008**, *26*, 187–196. [[CrossRef](#)]
29. Chen, W.; Liang, Y.J. New methodologies in fractional and fractal derivatives modeling. *Chaos Solitons Fractals* **2017**, *102*, 72–77. [[CrossRef](#)]
30. Liang, Y.J.; Ye, A.Q.; Chen, W.; Gatto, R.G.; Colon-Perez, L.; Mareci, T.H.; Magin, R.L. A fractal derivative model for the characterization of anomalous diffusion in magnetic resonance imaging. *Commun. Nonlinear Sci. Numer. Simul.* **2016**, *39*, 529–537. [[CrossRef](#)]
31. Cai, W.; Chen, W.; Xu, W.X. Characterizing the creep of viscoelastic materials by fractal derivative models. *Int. J. Non-Linear Mech.* **2016**, *87*, 58–63. [[CrossRef](#)]
32. Atangana, A.; Khan, M.A. Validity of fractal derivative to capturing chaotic attractors. *Chaos Solitons Fractals* **2019**, *126*, 50–59. [[CrossRef](#)]
33. Chen, W. Time-space fabric underlying anomalous diffusion. *Chaos Solitons Fractals* **2006**, *28*, 923–929. [[CrossRef](#)]
34. Hu, D.L.; Chen, W.; Sun, H.G. Power-law stability of Hausdorff derivative nonlinear dynamical systems. *Int. J. Syst. Sci.* **2020**, *51*, 601–607. [[CrossRef](#)]
35. Borges, E.P. A possible deformed algebra and calculus inspired in nonextensive thermostatistics. *Phys. A Stat. Mech. Its Appl.* **2004**, *340*, 95–101. [[CrossRef](#)]
36. Podlubny, I. Fractional-order systems and  $PI^\lambda D^\mu$  controllers. *IEEE Trans. Autom. Control* **1999**, *44*, 208–214. [[CrossRef](#)]