applied sciences

MDPI

*Article*

# Design of a Predictive RBF Compensation Fuzzy PID Controller for 3D Laser Scanning System

**Minghui Zhao [1,2], Xiaobin Xu [1,2,*], Hao Yang [1,2] and Zhijie Pan [1,2]**

[1] College of Mechanical and Electrical Engineering, Hohai University, Changzhou 213022, China; 191319010036@hhu.edu.cn (M.Z.); 171318010032@hhu.edu.cn (H.Y.); 1861910208@hhu.edu.cn (Z.P.)
[2] Jiangsu Key Laboratory of Special Robot Technology, Hohai University, Changzhou 213022, China
* Correspondence: xxbtc@hhu.edu.cn; Tel.: +86-158-5054-6794

Received: 8 June 2020; Accepted: 4 July 2020; Published: 6 July 2020

**Abstract:** A new proportional integral derivative (PID) control method is proposed for the 3D laser scanning system converted from 2D Lidar with a pitching motion device. It combines the advantages of a fuzzy algorithm, a radial basis function (RBF) neural network and a predictive algorithm to control the pitching motion of 2D Lidar quickly and accurately. The proposed method adopts the RBF neural network and feedback compensation to eliminate the unknown nonlinear part in the Lidar pitching motion, adaptively adjusting the PID parameter by a fuzzy algorithm. Then, the predictive control algorithm is adopted to optimize the overall controller output in real time. Finally, the simulation results show that the step response time of the Lidar pitching motion system using the control method is reduced from 15.298 s to 1.957 s with a steady-state error of 0.07°. Meanwhile, the system still has favorable response performance for the sinusoidal and step inputs under model mismatch and large disturbance. Therefore, the control method proposed above can improve the system performance and control the pitching motion of the 2D Lidar effectively.

**Keywords:** predictive control; fuzzy algorithm; RBF; PID control; Lidar pitching motion

## 1. Introduction

With the development of laser technology, 2D Lidar has been widely used in various fields such as map navigation [1,2], simultaneous localization and mapping (SLAM) [3,4] and robots [5,6]. However, the exploration of the environment requires more information dimensions and a more comprehensive perspective with the advancement of navigation and robot technology. Therefore, 3D Lidar technology emerges [7–10], but mature multi-line 3D Lidar is too expensive to be widely used. By contrast, a 3D laser scanning system formed by 2D Lidar with the pitching motion mechanism is more cost-effective, which can be reasonably applied to multiple fields [11–15].

For the 3D laser scanning system, many researchers focus only on the optimization of the calibration process [16–20], while neglecting the influence of 2D Lidar motion control on the scanning results. In fact, there is a deviation between the actual pitching motion of the 2D Lidar and the ideal motion. The size of the deviation directly affects the scanning results of the system. The speed of the 2D Lidar pitching motion should be adapted to different objects or environments, so the accuracy of the obtained data information can be improved, and the phenomenon of target point loss and overlap can be reduced. In short, it is important to design a control method that can effectively and accurately control the pitching motion of the 2D Lidar.

Proportional integral derivative (PID) control is one of the most commonly used control methods in various fields. Since 1942, researchers have put forward many PID control methods [21,22]. However, the controlled object generally has great uncertainty or complex nonlinearity in practical applications. For the pitching motion of the 2D Lidar in the article, the 2D Lidar rotates around the central axis of the system, so the torque generated by gravity will change nonlinearly with the pitching angle. Moreover, the pitching motion is accompanied by nonlinear and unknown friction torque. Therefore, the model of Lidar pitching motion is a complex system with unknown nonlinearity. For this kind of model, how to adjust PID parameters properly is not an easy task. Most tuning methods are mainly determined by the system model [23,24]. Bassi S. aimed at a linear motor system to put forward a PID tuning method, which used particle swarm optimization (PSO) to tune PID parameters [25]. Qing Wang used the internal model control (IMC) to adjust PID parameters for integrator/dead-time processes [26], but it cannot deal with an unknown disturbance. Julio E. proposed a unified approach to tune PID controllers for stable, integrative and unstable dead-time processes [27], which further optimizes the robustness of the system. However, its control performance for nonlinear systems is still poor. Although traditional PID control is widely applied in various industrial processes, its performance may not always be satisfactory due to unknown and nonlinear effects.

At present, many intelligent control algorithms have been proposed and widely used. Many of these algorithms can be used to reasonably improve the PID controller. Zadeh L. A. firstly put forward fuzzy set theory [28]. Its core is to establish a mathematical model of language analysis for complex systems or processes. Thus, the modeling problem of the system, which cannot be accurately described by a rigorous mathematical method, is solved. Since nonlinear systems often occur in practical applications, fuzzy control is required to solve these control problems [29,30]. Chiou used PSO to effectively search the optimal fuzzy PID controller parameters of an active automobile suspension system, which improved the nonlinear defects in the system [31]. Savran proposed a fuzzy adaptive adjustment method for PID parameters in a nonlinear process [32]. However, the fuzzy rules above are all based on long-term practical experience accumulated by the operators; such practical experience is not easy to obtain, which is a key issue in fuzzy adaptive PID control.

Neural networks are also an important branch of intelligent control algorithms. They can be used to solve the control problems of nonlinear and uncertain systems. Wei He studied the tracking control problem of an uncertain n-link robot with all-state constraints, and designed an adaptive neural network control to deal with the uncertainty and interference of the system [33]. Lei Liu used a radial basis function (RBF) to identify unknown dynamic errors in nonlinear switching systems, but there was always a certain steady-state error in the identified output value [34]. In addition to the intelligent control algorithms mentioned above, model predictive control, as a new type of computer control algorithm, has good prediction ability, which can optimize and compensate the controller output in real time. In [35], Jose R. used a feedforward neural network to fit the unknown nonlinear autoregressive moving average (NARMA) model; combined with predictive control theory, the control output signal is obtained by using the gradient descent method to minimize the error. Yuzhong Wang proposed a new PID control method that inherits the advantages of fuzzy PID control and predictive function control [36]. It used the current process input signal to predict the future process to improve the traditional fuzzy PID control. Therefore, it is reasonable to apply a neural network and the prediction ability of predictive control to improve PID control.

This paper proposes a predictive RBF compensation fuzzy PID controller (PRFPID) to control the pitching motion of 2D Lidar quickly and accurately. The structure of the article is as follows: In Section 2, a mathematical model of 2D Lidar pitching motion is established. Section 3 introduces the design of a feedback compensation fuzzy PID controller (FPID), which uses feedback compensation to eliminate the nonlinear part of the system and a fuzzy algorithm to adaptively adjust PID parameters. Section 4 introduces the design of the RBF compensation fuzzy PID controller (RFPID), which uses an RBF neural network to fit the unknown nonlinear part of the system. Then, the PRFPID is proposed by using predictive control to optimize the output of the controller based on the RFPID. In Section 5, the proposed PRFPID is simulated and compared with FPID and RFPID on a Lidar pitching motion system. In Section 6, the results reveal that the PRFPID can achieve favorable tracking performance and disturbance rejection capability.

## 2. Lidar Pitching Motion Model

As shown in Figure 1, the pitching device drives the 2D Lidar to swing up and down and scan the target object or environment. The planar 2D data from the 2D Lidar and the pitching angle of the encoder can be collected. Through an improved simplex calibration algorithm, the 2D data and the pitching angle information are converted to construct 3D data of the point cloud image of the real environment.

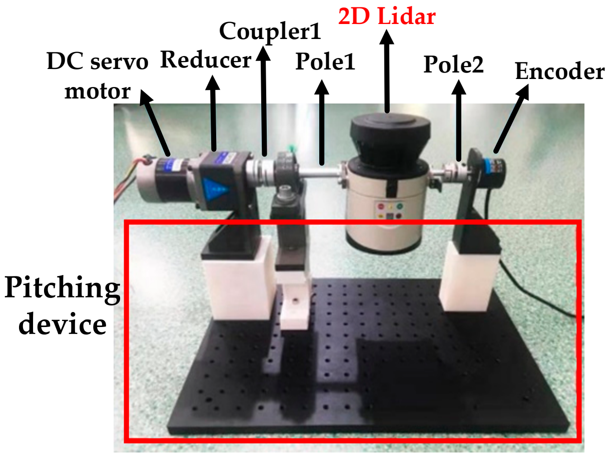
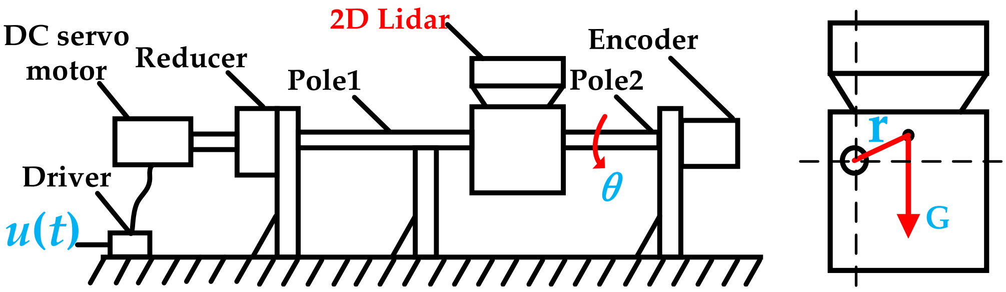


**Figure 1.** 3D Laser scanning system.

The 3D laser scanning system can be simplified to a schematic diagram to better analyze the Lidar motion process, which is shown in Figure 2. The input signal $u(t)$ of the driver controls the output torque of the DC servo motor. Through the reducer, the output shaft of the motor is connected to Pole 1, which drives the 2D Lidar to pitch. The 2D Lidar is connected to the encoder through Pole 2, and the output pitching angle information can be gained. In the pitching motion, the 2D Lidar will be subjected to the torque of gravity and friction torque. Thus, the relationship between the input signal $u(t)$ and the output pitching angle $\theta$ is

$$\ddot{\theta} = \frac{\mathrm{K_u K_m} u(t)}{\mathrm{JR}} + \frac{\mathrm{Grcos}\theta}{\mathrm{J}} - \frac{F_{\mathrm{f}}(t)}{\mathrm{J}} \tag{1}$$

where $\theta$ is the pitching angle, $\mathrm{K_u}$ is the amplification coefficient of the pulse width modulation (PWM) power amplifier, $\mathrm{K_m}$ is the torque coefficient of the motor, R is the armature resistance, G is the gravity of the 2D Lidar, r is the distance from the center of Lidar to the center of rotation, J is the moment of inertia of the Lidar around the system rotation center, and $F_{\mathrm{f}}(t)$ is a friction torque based on the Stribeck friction model.



(a) (b)

**Figure 2.** (**a**) Schematic diagram of the 3D laser scanning system; (**b**) schematic diagram of the Lidar's gravity center and rotation center.

As shown in Figure 3, the Stribeck curve indicates the relationship between friction torque and velocity at different friction stages. It can be expressed as

$$
\begin{aligned}
&F_f(t) = \begin{cases} \mathrm{F_m} & F(t) > \mathrm{F_m} \\ F(t) & -\mathrm{F_m} < F(t) < \mathrm{F_m} \\ -\mathrm{F_m} & F(t) < -\mathrm{F_m} \end{cases} \quad \left(\left|\dot{\theta}(t)\right| < \alpha\right) \\
&F_f(t) = \left(\mathrm{F_c} + (\mathrm{F_m} - \mathrm{F_c})\mathrm{e}^{-\alpha_1|\dot{\theta}(t)|}\right)\mathrm{sgn}\left(\dot{\theta}(t)\right) + \mathrm{k_v}\dot{\theta}(t) \quad \left(\left|\dot{\theta}(t)\right| > \alpha\right)
\end{aligned}
\tag{2}
$$

where *F*(*t*) is the driving torque, $\mathrm{F_m}$ is the maximum static friction torque, Fc is the Coulomb friction torque, $\mathrm{k_v}$ is the proportional coefficient of the viscous friction torque, $\theta$ is the pitching angle, $\alpha$ is the critical angular velocity, and $\alpha_1$ is 0.1.

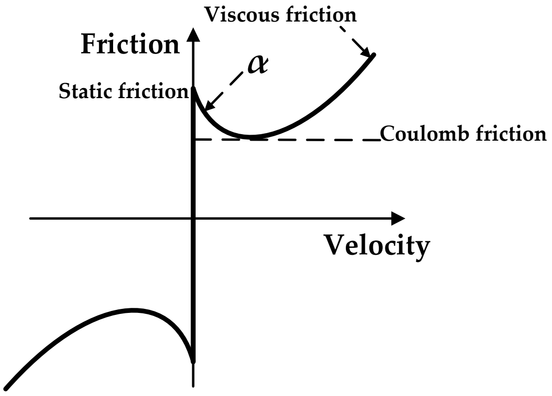


**Figure 3.** Stribeck curve.

## 3. Design of FPID

According to Section 2, the system contains nonlinear gravity torque and unknown friction torque. Therefore, the traditional PID control can hardly achieve the expected effect. For the nonlinear part of gravity, this section presents a control strategy based on FPID. As shown in Figure 4, the nonlinear part is firstly eliminated by feedback compensation. Then the relationship among the error *e*, the rate of error change *ec*, and the three PID parameters can be obtained through general fuzzy theory. Accordingly, the fuzzy rules are designed to adaptively adjust the three PID parameters.

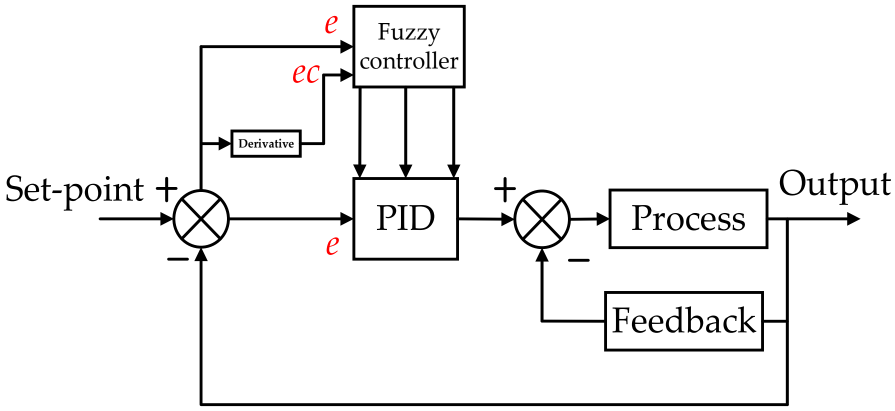


**Figure 4.** Fuzzy PID (FPID) control system.**Note:** PID = proportional integral derivative

### 3.1. Feedback Control Law

The feedback compensation can be used to eliminate the known nonlinear gravity of the system. The PID control law based on feedback compensation is

$$u_1^* = \frac{\mathrm{JR}}{\mathrm{K_u K_m}}\left[-\frac{\mathrm{Gr}\cos\theta}{\mathrm{J}} + \ddot{\theta}_\mathrm{d} + k_\mathrm{p} e + k_\mathrm{i}\int e\mathrm{d}t + k_\mathrm{d}\dot{e}\right] \tag{3}$$

where, $u_1$* is the control law of feedback compensation, $\theta_\mathrm{d}$ is the ideal angle value, $e$ is the deviation between the actual output value and the ideal angle value, $k_\mathrm{p}$, $k_\mathrm{i}$, $k_\mathrm{d}$ are the proportionality coefficient, integral coefficient and derivative coefficient respectively.

### 3.2. Fuzzy Adaptive Controller Description

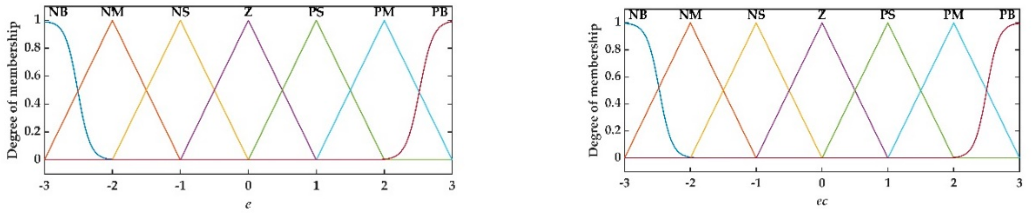
Fuzzy control is a type of rule-based control, which directly adopts language control rules. Since it does not require establishing an accurate mathematical model of the controlled object, the fuzzy control accuracy is often unsatisfactory. In this paper, it is combined with a PID controller to adjust the three PID parameters to enhance the performance of the controller. The input variables are $e$ and $ec$, and $k_\mathrm{p}$, $k_\mathrm{i}$ and $k_\mathrm{d}$ are output variables. Thus, the Mamdani two-input, three-output fuzzy controller is designed. The fuzzy operators are shown in Table 1.

**Table 1.** Fuzzy operators.

| And Method | Or Method | Implication | Aggregation | Defuzzification |
|---|---|---|---|---|
| min | max | min | max | centroid |

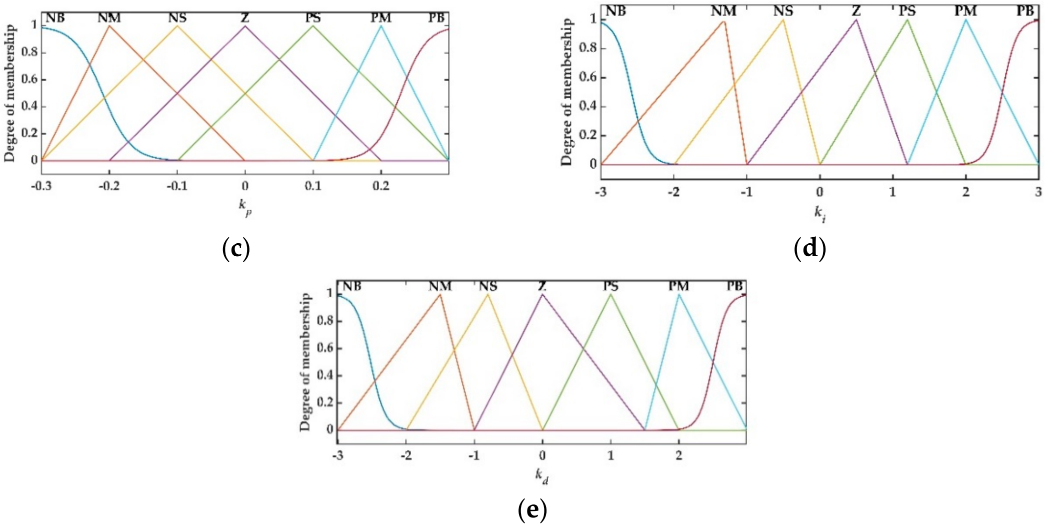
Firstly, the basic domains of the input variables are determined, as [−$\mathrm{x_e}$, $\mathrm{x_e}$], [−$\mathrm{x_{ec}}$, $\mathrm{x_{ec}}$]. The quantization factors $\mathrm{k_1}$, $\mathrm{k_2}$ are introduced to transform the input variables from their basic domain to the domain of the corresponding fuzzy set, as [−$\mathrm{n_e}$, $\mathrm{n_e}$], [−$\mathrm{n_{ec}}$, $\mathrm{n_{ec}}$]. The transformation relationship is $\mathrm{k_1 = n_e/x_e}$, $\mathrm{k_2 = n_{ec}/x_{ec}}$. The output variables are in the domain of the fuzzy set, as [−$\mathrm{n_p}$, $\mathrm{n_p}$], [−$\mathrm{n_i}$, $\mathrm{n_i}$] and [−$\mathrm{n_d}$, $\mathrm{n_d}$]. They must be converted into an acceptable domain of the controlled object. Hence, proportional factors of $\mathrm{k_3}$, $\mathrm{k_4}$ and $\mathrm{k_5}$ are introduced. Similarly, the basic domains of output variables are [$\mathrm{x_{p1}}$, $\mathrm{x_{p2}}$], [$\mathrm{x_{i1}}$, $\mathrm{x_{i2}}$] and [$\mathrm{x_{d1}}$, $\mathrm{x_{d2}}$]. According to the basic domains of output variables, the controller directly uses the fuzzy algorithm to adjust the PID parameter values. The transformation relationships are $\mathrm{k_3 = 2n_p/(x_{p2} - x_{p1})}$, $\mathrm{k_4 = 2n_i/(x_{i2} - x_{i1})}$ and $\mathrm{k_5 = 2n_d/(x_{d2} - x_{d1})}$. The input and output variables are selected as [−3, 3], [−3, 3], [−0.3, 0.3], [−3, 3], [−3, 3] and the fuzzy language set is expressed as [NB NM NS ZO PS PM PB].

Then the proper membership functions are set. Different shapes of the membership functions will cause different influences on the control performance. A low-resolution membership function should be selected for large errors, and a high-resolution membership function should be selected for errors that are close to zero. The triangular membership function and the sigmoid membership function are adopted respectively, which are shown in Figure 5.



(**a**) (**b**)

**Figure 5.** *Cont.*



(**c**) (**d**)

(**e**)

**Figure 5.** (**a**) Membership function *e*; (**b**) membership function *ec*; (**c**) membership function $k_p$; (**d**) membership function $k_i$; and (**e**) membership function $k_d$.

Fuzzy rules are the core of the fuzzy controller, which describe the fuzzy relationships between PID parameters and *e*, *ec*. The effect of the PID parameters on the system performance is shown in Table 2.

**Table 2.** Effect of PID parameters.

| PID Parameters | Effect |
|---|---|
| $k_p$ | Speed up system response and improve system accuracy. Large $k_p$ is likely to produce overshoot and even lead to system instability. Small $k_p$ will reduce system accuracy and slow down system response. |
| $k_i$ | Eliminate the steady-state error of system. Large $k_i$ can produce integral saturation or cause large overshoots and small $k_i$ will make it difficult to eliminate static errors in the system. |
| $k_d$ | Improve the dynamic performance of the system and suppress the change of error. A large $k_d$ will stop the response process prematurely and reduce the immunity of the system. |

According to Table 2, the PID parameters can be obtained to satisfy the different *e* and *ec*. Combined with the general rules in fuzzy adaptive PID control [37], the specific fuzzy rules are shown in Tables 3–5.

**Table 3.** Fuzzy rule $k_p$.

| *e* | *ec* | | | | | | |
|---|---|---|---|---|---|---|---|
| | $k_p$ | | | | | | |
| | **NB** | **NM** | **NS** | **ZO** | **PS** | **PM** | **PB** |
| NB | PB | PB | PM | PM | PS | ZO | ZO |
| NM | PB | PB | PM | PS | PS | ZO | NS |
| NS | PM | PM | PM | PS | ZO | NS | NS |
| ZO | PM | PM | PS | ZO | NS | NM | NM |
| PS | PS | PS | ZO | NS | NS | NM | NM |
| PM | PS | ZO | NS | NM | NM | NM | NB |
| PB | ZO | ZO | NM | NM | NM | NB | NB |

**Table 4.** Fuzzy rule $k_{\mathrm{i}}$.

| $e$ | $ec$ | | | | | | |
|---|---|---|---|---|---|---|---|
| | $k_{\mathrm{i}}$ | | | | | | |
| | **NB** | **NM** | **NS** | **ZO** | **PS** | **PM** | **PB** |
| NB | NB | NB | NM | NM | NS | ZO | ZO |
| NM | NB | NB | NM | NS | NS | ZO | ZO |
| NS | NB | NM | NS | NS | ZO | PS | PS |
| ZO | NM | NM | NS | ZO | PS | PM | PM |
| PS | NM | NS | ZO | PS | PS | PM | PB |
| PM | ZO | ZO | PS | PS | PM | PB | PB |
| PB | ZO | ZO | PS | PM | PM | PB | PB |

**Table 5.** Fuzzy rule $k_{\mathrm{d}}$.

| $e$ | $ec$ | | | | | | |
|---|---|---|---|---|---|---|---|
| | $k_{\mathrm{d}}$ | | | | | | |
| | **NB** | **NM** | **NS** | **ZO** | **PS** | **PM** | **PB** |
| NB | PS | NS | NB | NB | NB | NM | PS |
| NM | PS | NS | NB | NM | NM | NS | ZO |
| NS | ZO | NS | NM | NM | NS | NS | ZO |
| ZO | ZO | NS | NS | NS | NS | NS | ZO |
| PS | ZO | ZO | ZO | ZO | ZO | ZO | ZO |
| PM | PB | NS | PS | PS | PS | PS | PB |
| PB | PB | PM | PM | PM | PS | PS | PB |

Thus, according to the above fuzzy controller description, $k_{\mathrm{p}}$, $k_{\mathrm{i}}$ and $k_{\mathrm{d}}$ can be obtained by fuzzy approximate reasoning, as shown in Figure 6.

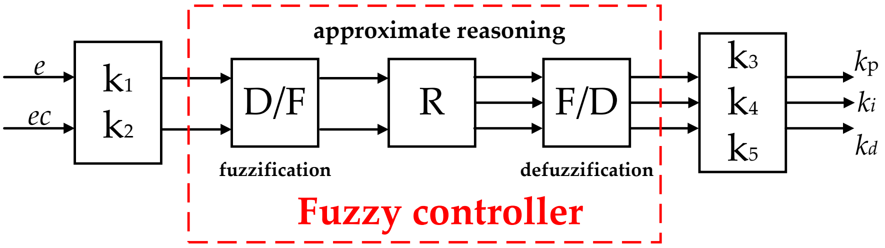


**Figure 6.** Fuzzy controller.

## 4. Design of RFPID and PRFPID

### 4.1. RFPID Controller Description

There is unknown nonlinear friction torque in the Lidar pitching motion system, which is difficult to account for using the simple feedback compensation method. Neural networks are a significant branch of intelligent control, which have self-learning and adaptive capabilities. A neural network system can be used to approximate the friction torque to achieve adaptive neural network compensation. As shown in Figure 7, the RBF neural network is used to approximate the unknown and nonlinear friction torque based on the FPID in the previous section. In this way, it can be utilized to compensate for the friction torque in the control law. Thus, the nonlinear influence caused by friction torque can be eliminated.

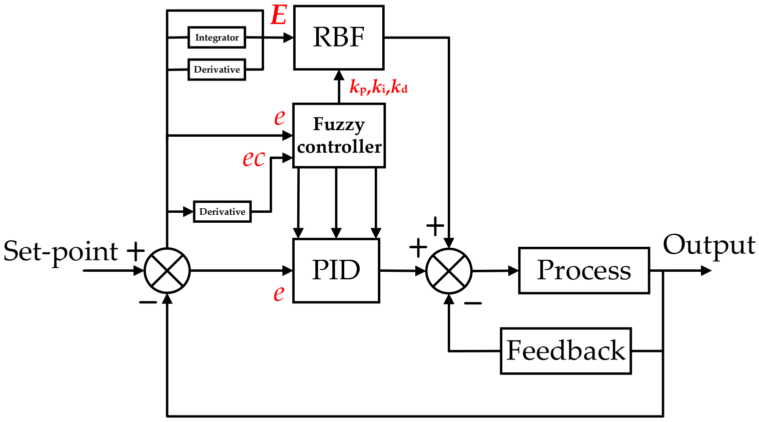


**Figure 7.** RBF compensation fuzzy PID controller (RFPID) control system.**Note:** RBF = radial basis function

For the system mathematical model in (1), $F_{\mathrm{f}}(t)$ is a friction torque based on the Stribeck friction model, whose coefficients are unknown. $F_{\mathrm{f}}(t)$ cannot be calculated by the model, but it can be obtained by the RBF neural network through error training. The RBF neural network is designed to fit the unknown nonlinear part. The algorithm of the RBF neural network is

$$\begin{aligned} \boldsymbol{c} &= \left[c_{\mathrm{ij}}\right] \\ h_{\mathrm{j}} &= \exp\left(\frac{-\|\boldsymbol{x}-\boldsymbol{c}_{\mathrm{j}}\|^2}{2\delta^2}\right) \\ f_{\mathrm{R}} &= \boldsymbol{W}^{\mathrm{T}}\boldsymbol{h} \end{aligned} \tag{4}$$

where $\boldsymbol{x}$ is network inputs which are the error, the rate of error change and the integral of the error, $i$ is the number of network inputs, $j$ is the number of hidden layer nodes in the network, $\boldsymbol{c}$ is the central value of the hidden node Gaussian function, $\delta$ is a normalized constant of the hidden node, $\boldsymbol{h}$ is the output of the Gaussian function and $\boldsymbol{W}$ is the weight of the neural network.

The relationship between the network output and actual friction torque is

$$\frac{F_{\mathrm{f}}}{\mathrm{J}} = f_{\mathrm{R}} + \varepsilon = \boldsymbol{W}^{\mathrm{T}}\boldsymbol{h} + \varepsilon \tag{5}$$

where $\varepsilon$ is the network approximation error.

The PID control law based on RBF compensation is designed as

$$u_2^* = \frac{\mathrm{JR}}{\mathrm{K_u K_m}}\left[f_{\mathrm{R}} - \frac{\mathrm{Grcos}\theta}{\mathrm{J}} + \ddot{\theta}_{\mathrm{d}} + k_{\mathrm{p}}e + k_{\mathrm{i}}\int e\mathrm{d}t + k_{\mathrm{d}}\dot{e}\right] \tag{6}$$

By substituting (6) into (1), the closed-loop dynamic equation of the system can be obtained as

$$\begin{cases} \dot{\boldsymbol{E}} = \boldsymbol{A}\boldsymbol{E} + \boldsymbol{b}\left[f_{\mathrm{R}} - \frac{F_{\mathrm{f}}}{\mathrm{J}}\right] \\ \boldsymbol{E} = \begin{bmatrix} \int e\mathrm{d}t \\ e \\ \dot{e} \end{bmatrix}, \boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -k_{\mathrm{i}} & -k_{\mathrm{p}} & -k_{\mathrm{d}} \end{bmatrix}, \boldsymbol{b} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{cases} \tag{7}$$

where $e$ is the error of system output, $k_{\mathrm{p}}$, $k_{\mathrm{i}}$ and $k_{\mathrm{d}}$ are respectively the proportional coefficient, integral coefficient and derivative coefficient and $\boldsymbol{A}$ and $\boldsymbol{b}$ are the corresponding coefficient matrix.

The optimal weight parameter of the neural network is

$$\boldsymbol{W}^* = \underset{WI\Omega}{\mathrm{argmin}}\left[\sup\left|f_{\mathrm{R}} - \frac{F_{\mathrm{f}}}{\mathrm{J}}\right|\right] \tag{8}$$

where $\boldsymbol{\Omega}$ is the set of $\boldsymbol{W}$.

Thus, the minimum approximation error $\omega$ is defined as

$$\omega = f_{\mathrm{R}}(x|\boldsymbol{W}^{*}) - \frac{F_{\mathrm{f}}}{\mathrm{J}} \tag{9}$$

According to (7) and (8), the final closed-loop dynamic equation of the system is

$$\dot{\boldsymbol{E}} = \boldsymbol{AE} + \boldsymbol{b}\left[\left(\hat{\boldsymbol{W}} - \boldsymbol{W}^{*}\right)^{\mathrm{T}}\boldsymbol{h} + \omega\right] \tag{10}$$

Based on (10), the adaptive law of $\boldsymbol{W}$ is determined to minimize the system error. The Lyapunov function of the closed-loop system is

$$V = \frac{1}{2}\boldsymbol{E}^{\mathrm{T}}\boldsymbol{PE} + \frac{1}{2\gamma}\left(\hat{\boldsymbol{W}} - \boldsymbol{W}^{*}\right)^{\mathrm{T}}\left(\hat{\boldsymbol{W}} - \boldsymbol{W}^{*}\right) \tag{11}$$

where $\gamma$ is a constant coefficient and $\boldsymbol{P}$ is the positive-determined matrix that satisfies the Lyapunov equation.

$\boldsymbol{Q}$ is defined as a matrix which satisfies the following equation:

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{P} + \boldsymbol{PA} = -\boldsymbol{Q} \tag{12}$$

where $\boldsymbol{Q}$ is an arbitrary $3 \times 3$ positive-determined matrix.

By substituting (10) into (11), the derivation of Lyapunov is

$$\dot{V} = -\frac{1}{2}\boldsymbol{E}^{\mathrm{T}}\boldsymbol{QE} + \boldsymbol{E}^{\mathrm{T}}\boldsymbol{Pb}\omega + \frac{1}{\gamma}\left(\hat{\boldsymbol{W}} - \boldsymbol{W}^{*}\right)^{\mathrm{T}}\left[\dot{\hat{\boldsymbol{W}}} + \gamma\boldsymbol{E}^{\mathrm{T}}\boldsymbol{Pbh}\right] \tag{13}$$

The adaptive law of $\boldsymbol{W}$ can be written as

$$\dot{\hat{\boldsymbol{W}}} = -\gamma\boldsymbol{E}^{\mathrm{T}}\boldsymbol{Pbh} \tag{14}$$

The derivation of Lyapunov can be simplified to

$$\dot{V} = -\frac{1}{2}\boldsymbol{E}^{\mathrm{T}}\boldsymbol{QE} + \boldsymbol{E}^{\mathrm{T}}\boldsymbol{Pb}\omega \tag{15}$$

In (15), the derivative of the Lyapunov equation is less than or equal to zero by choosing the appropriate $\boldsymbol{Q}$ and the minimum approximation error $\omega$. According to Lyapunov's second theorem of stability, the system is stable in the sense of Lyapunov.

Based on (15), the inequality is

$$2\boldsymbol{E}^{\mathrm{T}}\boldsymbol{Pb}\omega \le \mathrm{d}\left(\boldsymbol{E}^{\mathrm{T}}\boldsymbol{Pb}\right)\left(\boldsymbol{E}^{\mathrm{T}}\boldsymbol{Pb}\right)^{\mathrm{T}} + \frac{\omega^{2}}{\mathrm{d}} \tag{16}$$

where d is a positive constant.

The inequality can be rewritten as

$$\dot{V} = -\frac{1}{2}\boldsymbol{E}^{\mathrm{T}}\boldsymbol{QE} + \boldsymbol{E}^{\mathrm{T}}\boldsymbol{Pb}\omega \le -\frac{1}{2}\boldsymbol{E}^{\mathrm{T}}\boldsymbol{QE} + \frac{\mathrm{d}}{2}\left(\boldsymbol{E}^{\mathrm{T}}\boldsymbol{Pb}\right)\left(\boldsymbol{E}^{\mathrm{T}}\boldsymbol{Pb}\right)^{\mathrm{T}} + \frac{\omega^{2}}{2\mathrm{d}} \tag{17}$$

The inequality can be simplified to

$$\dot{\boldsymbol{V}} \le -\frac{1}{2}l_{\mathrm{min}}\left(\boldsymbol{Q} - \mathrm{d}\left(\boldsymbol{Pbb}^{\mathrm{T}}\boldsymbol{P}^{\mathrm{T}}\right)\right)\|\boldsymbol{E}\|^{2} + \frac{\boldsymbol{\omega}_{\mathrm{max}}^{2}}{2\mathrm{d}} \tag{18}$$

where $l(\bullet)$ is the eigenvalue of the matrix, and $l(\boldsymbol{Q}) > l(\mathrm{d}\boldsymbol{P}\boldsymbol{b}\boldsymbol{b}^{\mathrm{T}}\boldsymbol{P}^{\mathrm{T}})$.

According to (18), when the derivative of the Lyapunov equation is less than or equal to zero, the convergence result is

$$\|\boldsymbol{E}\| \leq \frac{|\omega_{\max}|}{\sqrt{\mathrm{d}\boldsymbol{l}_{\min}\left(\boldsymbol{Q} - \mathrm{d}\left(\boldsymbol{P}\boldsymbol{b}\boldsymbol{b}^{\mathrm{T}}\boldsymbol{P}^{\mathrm{T}}\right)\right)}} \tag{19}$$

It can be seen from (19) that the convergence error is related to the eigenvalues of $\boldsymbol{Q}$, $\boldsymbol{P}$ and the approximation error $\omega$. The larger the eigenvalue of $\boldsymbol{Q}$ is, the smaller the eigenvalue of $\boldsymbol{P}$ is. Meanwhile, the smaller that $|\omega_{\max}|$ is, the smaller the convergence error will be. The control system does not have asymptotic stability in its equilibrium state $\mathrm{x}_{\mathrm{e}} = 0$. Hence, there will be a certain steady-state error between the output value and the target value. Finally, the $f_{\mathrm{R}}$ pseudocode is shown in Algorithm 1.

**Algorithm 1.** $f_{\mathrm{R}}$ pseudocode.

```
Input: $c$; $\delta$; $h$; $W$; $E$; $P$; $\gamma$; $b$; $t_s$.
Output: $f_R$
for $j$ = 1 to 5 do
    $h_j$ = exp(-($E$-$c_j$)$^2$/(2*$\delta^2$));
end for
$S$=-$\gamma$*$E^T$*$P$*$b$*$h$;
for $i$ = 1 to 5 do
    $W_i$= $S_i$*$t_s$+$W_i$;
end for
$f_R$ = -$W^T$*$h$;
```

*4.2. PRFPID Controller Description*

Predictive control with good predictive ability should consider the influence of uncertainty and perform real-time optimization during the process operation. From the analysis in the previous section, the RFPID controller does not guarantee acceptable convergence error and the anti-interference ability is weak. Based on the RFPID controller, this section combines the algorithmic idea of predictive control to compensate for the system steady-state error and enhance its robustness.

As shown in Figure 8, the predictive controller is added to modify the RFPID controller. The output of the prediction controller is calculated to compensate the output of the overall controller in real time. Ignoring the fitting error $\varepsilon$ of the RBF neural network, the discrete state space expression of the closed-loop control system is

$$\begin{cases} \boldsymbol{x}(k+1) = \boldsymbol{A}_{\mathrm{P}}\boldsymbol{x}(k) + \boldsymbol{b}_{\mathrm{P}}u_3(k) \\ y(k) = \boldsymbol{c}_{\mathrm{P}}^{\mathrm{T}}\boldsymbol{x}(k) \\ \\ \boldsymbol{x} = \begin{bmatrix} \int e\mathrm{d}t \\ e \\ \dot{e} \end{bmatrix}, \boldsymbol{A}_{\mathrm{P}} = \begin{bmatrix} 1 & t_{\mathrm{s}} & 0 \\ 0 & 1 & t_{\mathrm{s}} \\ -k_{\mathrm{i}}t_{\mathrm{s}} & -k_{\mathrm{p}}t_{\mathrm{s}} & 1-k_{\mathrm{d}}t_{\mathrm{s}} \end{bmatrix}, \boldsymbol{b}_{\mathrm{P}} = \begin{bmatrix} 0 \\ 0 \\ t_{\mathrm{s}} \end{bmatrix}, \boldsymbol{c}_{\mathrm{P}} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \end{cases} \tag{20}$$

where $u_3(k)$ is the output of predictive controller at time $k$, $\boldsymbol{x}$ is the state variable of the closed-loop system, $y$ is the output of the system, $e$ is the output error, $t_{\mathrm{s}}$ is the sampling time, $\boldsymbol{A}_{\mathrm{p}}$ is the system matrix, $\boldsymbol{b}_{\mathrm{p}}$ is the control matrix and $\boldsymbol{c}_{\mathrm{p}}$ is the output matrix.

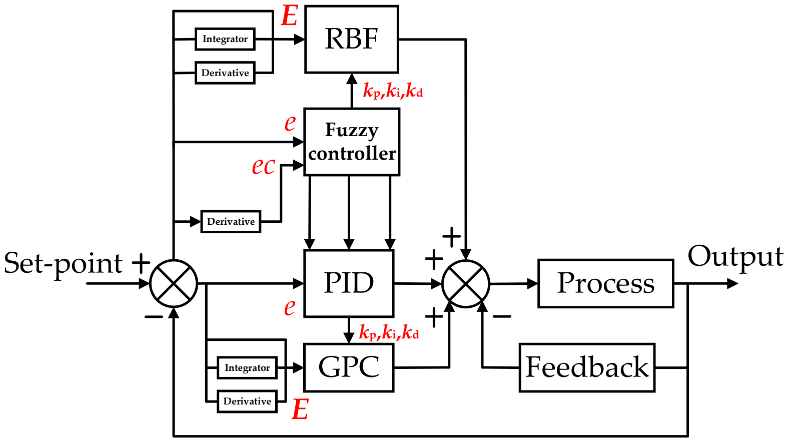


**Figure 8.** RFPID control system.

According to (20), the predicted step number is set to $P$ and the control step number is set to $M$. Thus, the model of the predictive controller is

$$\begin{cases} \boldsymbol{Y}(k)=\boldsymbol{F}_{\mathrm{y}}\boldsymbol{x}(k)+\boldsymbol{G}_{\mathrm{y}}\boldsymbol{U}_3(k) \\ \boldsymbol{Y}(k)=\begin{bmatrix} y(k+1)\\ \vdots \\ y(k+P)\end{bmatrix}_{P\times1}, \boldsymbol{F}_y=\begin{bmatrix} \boldsymbol{c}_{\mathrm{P}}^{\mathrm{T}}\boldsymbol{A}_{\mathrm{P}}\\ \vdots\\ \boldsymbol{c}_{\mathrm{P}}^{\mathrm{T}}\boldsymbol{A}_{\mathrm{P}}^{\mathrm{P}}\end{bmatrix}_{P\times 3} \\ \boldsymbol{G}_{\mathrm{y}}=\begin{bmatrix} \boldsymbol{c}_{\mathrm{P}}^{\mathrm{T}}\boldsymbol{b}_{\mathrm{P}} & 0 & 0 \\ \vdots & \ddots & 0 \\ \boldsymbol{c}_{\mathrm{P}}^{\mathrm{T}}\boldsymbol{A}_{\mathrm{P}}^{M-1}\boldsymbol{b}_{\mathrm{P}} & \cdots & \boldsymbol{c}_{\mathrm{P}}^{\mathrm{T}}\boldsymbol{b}_{\mathrm{P}} \\ \vdots & \ddots & \vdots \\ \boldsymbol{c}_{\mathrm{P}}^{\mathrm{T}}\boldsymbol{A}_{\mathrm{P}}^{P-1}\boldsymbol{b}_{\mathrm{P}} & \cdots & \sum\limits_{i=1}^{P-M+1}\boldsymbol{c}_{\mathrm{P}}^{\mathrm{T}}\boldsymbol{A}_{\mathrm{P}}^{i-1}\boldsymbol{b}_{\mathrm{P}} \end{bmatrix}_{P\times M} \end{cases} \tag{21}$$

where $\boldsymbol{Y}$ is the predicted $P$ output vector expression, $\boldsymbol{F}_{\mathrm{y}}$ is the coefficient matrix of the state variable and $\boldsymbol{G}_{\mathrm{y}}$ is the input coefficient matrix.

According to the model of predictive control in (21), the $M$ control quantity in the future moments can be obtained by minimizing the error of the controlled object in $P$ times and suppressing the fluctuation of the control quantity. The optimized performance index equation is

$$\min_{U_3(k)} J(k)=\|\boldsymbol{N}(k)-\boldsymbol{Y}(k)\|_L^2+\|\boldsymbol{U}_3(k)\|_O^2 \tag{22}$$

where $\boldsymbol{N}(k) = [n(k+1) \ldots n(k+P)]$ is the output target value vector and $\boldsymbol{L}$ and $\boldsymbol{O}$ are the weighted matrices of the output and control quantity respectively.

In predictive control algorithms, $\boldsymbol{N}$ usually needs to specify a smooth curve that is close to the target value to improve the stability of the system. The curve is

$$n(k+j)=\mu^j y(k)+\left(1-\mu^j\right)y_{\mathrm{r}} \quad (j=1,2,\cdots,P) \tag{23}$$

where $\mu$ is the flexibility coefficient, $0 < \mu < 1$, $y(k)$ is the actual output value of the system and $y_{\mathrm{r}}$ is the target value.

According to (22), the direction of the fastest gradient is selected, and the optimal output of predictive controller is

$$U_3^*(k) = \boldsymbol{Z}\Big(\big(\boldsymbol{G}_y^T \boldsymbol{L} \boldsymbol{G}_y + \boldsymbol{O}\big)^{-1} \boldsymbol{G}_y^T \boldsymbol{L}\big(\boldsymbol{N}(k) - \boldsymbol{F}_y \boldsymbol{x}(k)\big)\Big) \tag{24}$$

where $\boldsymbol{Z} = [1\ 0 \ldots\ 0]_{1\times M}$.

Finally, the $U^*_3$ pseudocode is shown in the Algorithm 2.

**Algorithm 2.** $U^*_3$ pseudocode.

```
Input: $E$; $P$; $M$; $A_p$; $b_p$; $c_p$; $L$; $O$; $Z$.
Output: $U_3^*$.
for $i = 1$ to $P$ do
  if $i<=M$ then
    $m = i$;
  else
    $m = M$;
  end if
  for $j = 1$ to $m$ then
    if $j==M$ then
      for $k = 1$ to $i - j + 1$ do
        $G_{ij} = G_{ij} + c_p^{T} * A_p^{k-1} * b_p$;
      end for
    else
      $G_{ij} = c_p^{T} * A_p^{i-j} * b_p$;
    end if
  end for
end for
for $i = 1$ to $P$ do
  for $j = 1$ to 3 do
    $F_i = c_p^{T} * A_p^{i}$;
  end for
end for
$U_3^* = Z*(G^T*L*G+O)^{-1}*G^T*L*(N-F*E)$;
```

## 5. Simulation

Simulink in MATLAB is used to verify whether PRFPID can accurately control the pitching motion of the 2D Lidar. The first step is to discretize the controller and the controlled object. The discrete state space expression of (1) is

$$\begin{array}{l} x_1(k+1) = x_1(k) + x_2(k)t_s \\ x_2(k+1) = x_2(k) + \frac{t_s \mathrm{Grcos}(x_1(k))}{\mathrm{J}} + \frac{t_s \mathrm{K_u K_m} u(k)}{\mathrm{JR}} - \frac{t_s F_f(k)}{\mathrm{J}} \\ y(k) = x_1(k) \end{array} \tag{25}$$

where $u(k)$ is the system input at time $k$, $x_1$ is the pitching angle, $x_2$ is the pitching angular velocity and $t_s$ is the sampling time.

According to (25) and the pseudocode of Section 4, the S function in Simulink is used to describe the pitching motion of the 2D Lidar, RBF and generalized predictive control (GPC). The fuzzy controller is built by using the fuzzy toolbox in MATLAB. The boundary parameters of the Lidar pitching motion system are set as shown in Table 6. Based on Figure 8, the simulation analysis of the PRFPID is carried out and compared with the RFPID and FPID [38].

**Table 6.** System parameters.

| **Parameters** | **$K_u$(V/V)** | **$K_m$(N•m/A)** | **R(Ω)** | **G(N)** | **r(m)** |
|---|---|---|---|---|---|
| **Value** | 11 | 0.6 | 7.77 | 11 | 0.058 |
| **Parameters** | J(kg•m$^2$) | $F_m$(N•m) | $F_c$(N•m) | $k_v$(N•m•s/°) | α(°/s) |
| **Value** | 0.00662 | 0.2 | 0.15 | 0.02 | 0.01 |

### 5.1. Step Response

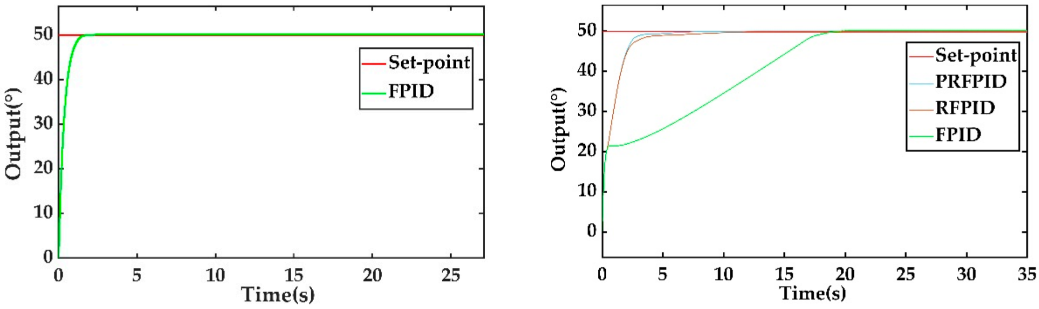
The sampling time is 0.01 s, and the controller parameters are set according to the range of the system error and error change rate, which are shown in Table 7.

**Table 7.** Controller parameters of step response.

| **Parameters** | **Type** | | |
|---|---|---|---|
| | **FPID** | **RFPID** | **PRFPID** |
| $k_1$, $k_2$, $k_3$, $k_4$, $k_5$ | 0.1, 0.1, 0.5, 0.5, 1 | 0.1, 0.1, 0.5, 0.5, 1 | 0.1, 0.1, 0.5, 0.5, 1 |
| RBF nodes | | 5 | 5 |
| ***c*** | | [0, 10, 25, 35, 50; 0, 10, 25, 35, 50; −100, −50, 0, 50, 100] | |
| ***Q*** | | [5000, 0, 0; 0, 5000, 0; 0, 0, 5000] | |
| $\delta$, $\gamma$ | | 10, 15 | 10,15 |
| $P$, $M$, $\mu$ | | | 10, 3, 0.6 |
| Set-point | 50 | 50 | 50 |

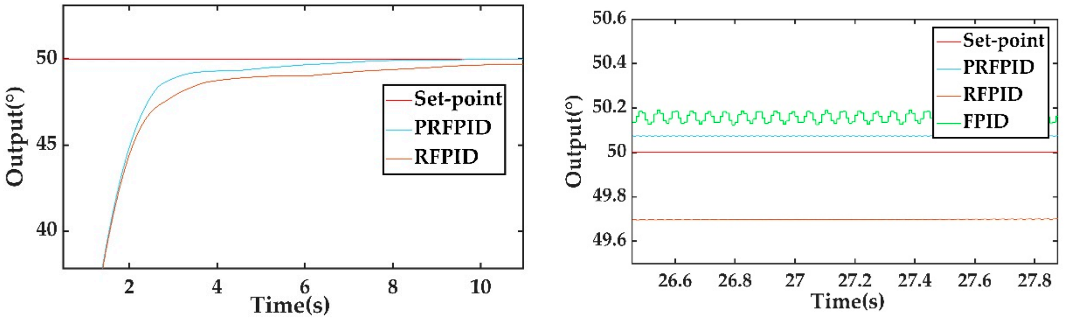
In Table 7, careful selection of flexibility coefficient is needed. The larger the coefficient of flexibility $\mu$ is, the stronger the anti-interference ability is. However, too large a coefficient results in system overshoot or oscillation. The step signal with a target value of 50 degrees is set. Three kinds of controller response curves are shown below.

Figure 9a shows the control result of FPID when the unknown friction of the system is ignored. The result, which has favorable response performance, indicates the feedback compensation can eliminate the nonlinear factors to obtain great control performance. However, the above result was simulated in the case of zero friction. When the system has the unknown and nonlinear influence factors, FPID cannot achieve a satisfying control effect. In Figure 9b,c the comparison of the three controllers shows that the response speed of FPID is very slow while the response speed of RFPID and PRFPID is significantly accelerated after RBF compensation. The error of PRFPID is the smallest in Figure 9d, and it indicates that the predictive control effectively reduces the steady-state error in RFPID. Moreover, it can be seen from Table 8 that PRFPID has the fastest response speed and the smallest steady-state error. Hence, the overall system performance has been improved by the RBF neural network and the predictive control algorithm.



(**a**) (**b**)

**Figure 9.** *Cont.*



(**c**) (**d**)

**Figure 9.** (**a**) FPID without friction; (**b**) step responses; (**c**) step response amplification; (**d**) steady-state error.

**Table 8.** Response index of control.

| | Response Time (s) | Overshoot (%) | Steady-State Error (°) | Setting Time (s) |
|---|---|---|---|---|
| **FPID** | 15.298 | 0.38% | 0.19 | 17.658 |
| **GFPID** | 2.03 | 0 | 0.3 | 5.171 |
| **PGFPID** | 1.957 | 0.14% | 0.07 | 3.146 |

### *5.2. Robustness Simulation*

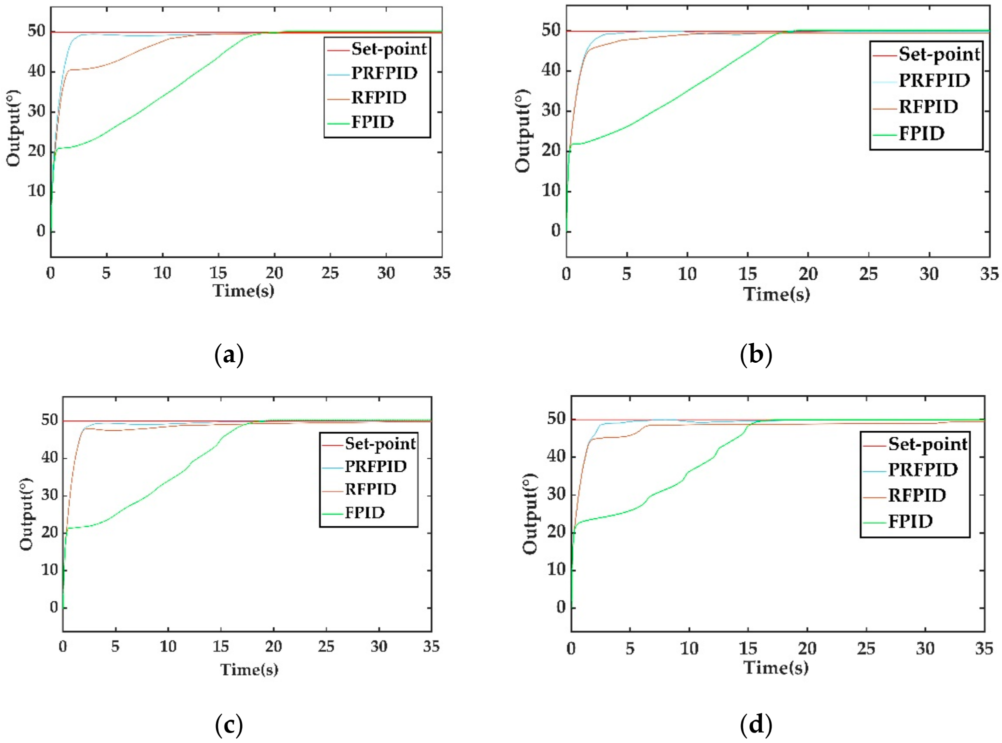
In practice, mismatch could occur between the established system model and the actual process, and the system may be affected by some unknown disturbance. In order to verify whether PRFPID can improve the robustness of the system, simulation analysis is carried out on the three controllers in the case of model mismatch and large disturbance.

In order to simulate the model mismatch in the actual process, the Monte Carlo method is used to randomly generate four kinds of mismatch models for the parameters that may have deviations in the system mathematical model. The maximum uncertainty is 30%. As shown in Table 9, the three parameters are gravity of 2D Lidar G, the distance from the center of Lidar to the center of rotation r and the moment of inertia of Lidar around the system rotation center J.

**Table 9.** Mismatch model parameters.

| Case | G (N) | r (m) | J (kg•m$^2$) |
|---|---|---|---|
| **1** | 13.0772 | 0.044 | 0.0085 |
| **2** | 13.6782 | 0.053 | 0.0053 |
| **3** | 8.5381 | 0.0596 | 0.0066 |
| **4** | 11.8736 | 0.0739 | 0.0048 |

Figure 10 shows the response curves of the four models under mismatch conditions. It can be seen that the PRFPID controller can still achieve favorable control results in the case of four random model mismatches. Compared with RFPID and FPID, it can track input settings more quickly. When the output curves of RFPID and FPID are not stable, the output curve of PRFPID is stable and does not have large oscillations or fluctuations.



(**a**) (**b**)

(**c**) (**d**)

**Figure 10.** (**a**) Step responses for case 1; (**b**) step responses for case 2; (**c**) step responses for case 3; (**d**) step responses for case 4.

In order to simulate the response curves of the three controllers under the effect of large disturbance, a step disturbance of −40 degrees is added to the system output at 25 s. The flexibility coefficient $\mu$ is changed to 0.8 to enhance the anti-interference ability of PRFPID. Figure 11 shows the comparison results of the three controllers under the step disturbance. PRFPID can quickly return to the target setting value while RFPID and FPID return slowly. Therefore, PRFPID has strong anti-interference ability and can deal with the case of model mismatch properly.

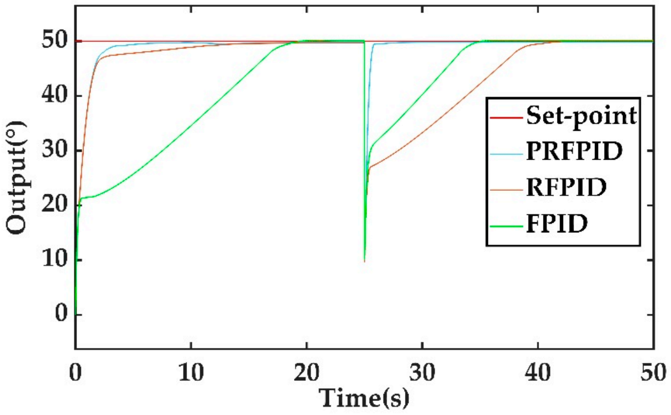


**Figure 11.** Step response under disturbance.

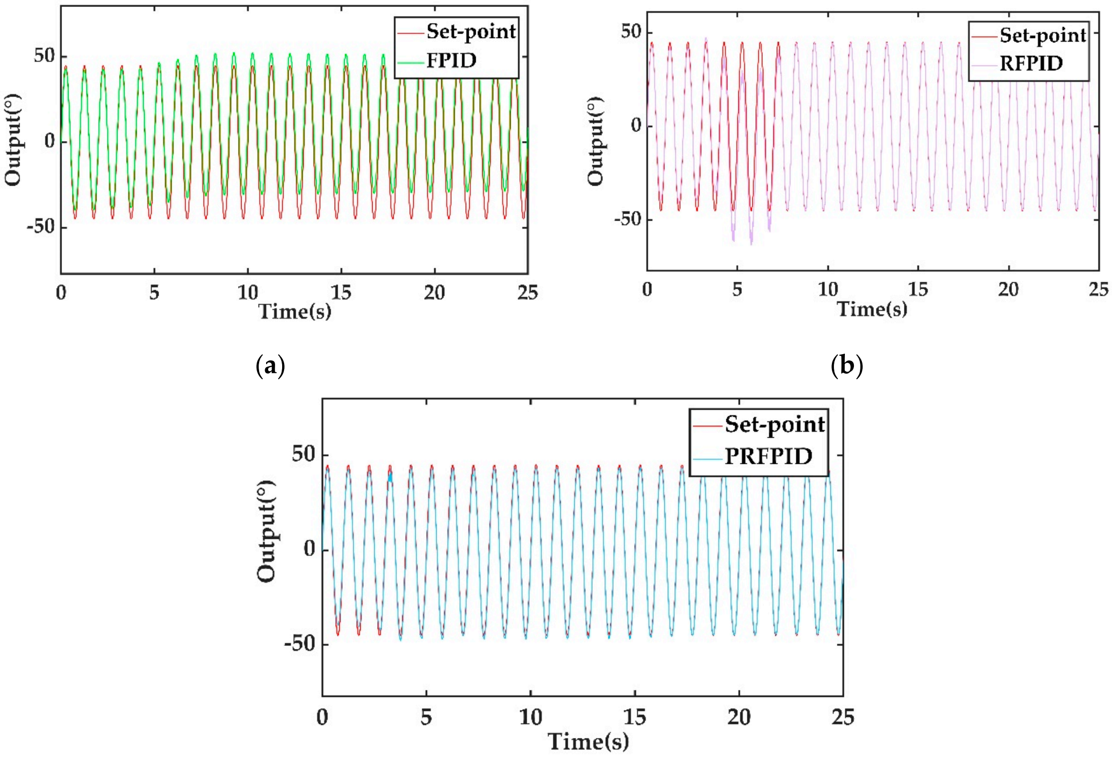
### 5.3. Sinusoidal Response

In view of the actual situation of Lidar pitching motion, the target curve is generally periodic. Lidar needs to swing back and forth periodically, so a sinusoidal input signal is adopted for simulation. The algorithm parameters need to be changed according to the range of error and error change rate. Because the input signal value changes frequently, the prediction period is reduced. The parameters required for the algorithm are set, as shown in Table 10. The target curve is set as a sinusoidal signal with $y = 45\sin(2\pi t)$ and three controller response results are achieved.

**Table 10.** Controller parameters of sine response.

| Parameters | Type | | |
|---|---|---|---|
| | FPID | RFPID | PRFPID |
| $k_1$, $k_2$, $k_3$, $k_4$, $k_5$ | 0.5, 0.5, 50, 25, 20 | 0.5, 0.5, 50, 25, 20 | 0.5, 0.5, 50, 25, 20 |
| RBF nodes | | 5 | 5 |
| $\boldsymbol{c}$ | | [−6, −3, 0, 3, 6; −6, −3, 0, 3, 6; −200, −100, 0, 100, 200] | |
| $\boldsymbol{Q}$ | | [500, 0, 0; 0, 500, 0; 0, 0, 500] | |
| $\delta$, $\gamma$ | | 10, 3 | 10, 3 |
| $P$, $M$, $\mu$ | | | 5, 3, 0.6 |
| Set-point | $45\sin(2\pi t)$ | $45\sin(2\pi t)$ | $45\sin(2\pi t)$ |

Figure 12a shows the sinusoidal response curve of FPID. The response curve deviates significantly from the target curve. Figure 12b is the sinusoidal response curve of RFPID. After the compensation effect of RBF, the system response curve is basically consistent with the input curve, but there is a huge fluctuation and the curve is unstable in the range of 4 to 7 s. As shown in Figure 12c, the PRFPID sinusoidal response curve is stable. It eliminates the deviation in FPID and the fluctuation phenomenon in RFPID, tracking the target curve quickly and accurately.



(**a**) (**b**)

(**c**)

**Figure 12.** (**a**) FPID sine response; (**b**) RFPID sine response; and (**c**) PRFPID sine response.

## 6. Conclusions

This paper studies the control of Lidar pitching motion in a 3D laser scanning system. A predictive RBF compensation fuzzy PID controller is proposed, which combines the advantages of a fuzzy algorithm, an RBF neural network and predictive control. Compared with traditional PID, PRFPID eliminates the unknown nonlinear part of the system and has good tracking performance and robustness.

Through the simulation comparison with the FPID and RFPID controller, the general step response is obtained, and the response time is reduced from 15.298 s to 1.957 s with a steady-state error of 0.07°. Meanwhile, the PRFPID remains as the one with the most favorable response performance for the step input under model mismatch and large disturbance. Moreover, the PRFPID tracks the target curve quickly and accurately and eliminates the deviation in FPID and the fluctuation phenomenon in RFPID by the sinusoidal response comparison. In conclusion, the results show that the PRFPID controller improves the system performance and can effectively control the pitching motion of the 2D Lidar.

**Author Contributions:** Conceptualization, X.X.; validation, X.X.; writing—original draft preparation, X.X. and M.Z.; writing—review and editing, M.Z., H.Y. and Z.P. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by National Natural Science Foundation of China, grant number 51805146; the Fundamental Research Funds for the Central Universities, grant numbers B200202221; Changzhou Sci&Tech Program, grant numbers CJ20180047; Jiangsu Key R&D Program, grant numbers BE2018004 and BE2018004-1; Water Resources Department of Jiangsu Province, grant number 2018031.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Kohlbrecher, S.; Meyer, J.; Graber, T.; Petersen, K.; Klingauf, U.; von Stryk, O. Hector open source modules for autonomous mapping and navigation with rescue robots. In Proceedings of the Robot Soccer World Cup; Springer: Berlin/Heidelberg, Germany, 2013; pp. 624–631.
2. Li, R.; Liu, J.; Zhang, L.; Hang, Y. LIDAR/MEMS IMU integrated navigation (SLAM) method for a small UAV in indoor environments. In Proceedings of the 2014 DGON Inertial Sensors and Systems (ISS), Karlsruhe, Germany, 16–17 September 2014; pp. 1–15.
3. Hess, W.; Kohler, D.; Rapp, H.; Andor, D. Real-time loop closure in 2D LIDAR SLAM. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 1271–1278.
4. Yagfarov, R.; Ivanou, M.; Afanasyev, I. Map comparison of lidar-based 2d slam algorithms using precise ground truth. In Proceedings of the 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), Singapore, 18–21 November 2018; pp. 1979–1983.
5. Denysyuk, P.; Teslyuk, V.; Chorna, I. Development of mobile robot using LIDAR technology based on Arduino controller. In Proceedings of the 2018 XIV-th International Conference on Perspective Technologies and Methods in MEMS Design (MEMSTECH), Lviv, Ukraine, 18–22 April 2018; pp. 240–244.
6. Adiwahono, A.H.; Saputra, V.B.; Ng, K.P.; Gao, W.; Ren, Q.; Tan, B.H.; Chang, T. Human tracking and following in dynamic environment for service robots. In Proceedings of the TENCON 2017-2017 IEEE Region 10 Conference, Penang, Malaysia, 5–8 November 2017; pp. 3068–3073.
7. Trochta, J.; Krůček, M.; Vrška, T.; Král, K. 3D Forest: An application for descriptions of three-dimensional forest structures using terrestrial LiDAR. *PLoS ONE* **2017**, *12*, e0176871. [CrossRef] [PubMed]
8. Xu, N.; Zhang, W.; Zhu, L.; Li, C.; Wang, S. Object 3D surface reconstruction approach using portable laser scanner. In *IOP Conference Series: Earth and Environmental Science*; IOP Publishing: Bristol, UK, 2017; Volume 69, p. 12119.
9. Chi, S.; Xie, Z.; Chen, W. A laser line auto-scanning system for underwater 3D reconstruction. *Sensors* **2016**, *16*, 1534. [CrossRef] [PubMed]
10. Wen, C.; Pan, S.; Wang, C.; Li, J. An indoor backpack system for 2-D and 3-D mapping of building interiors. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 992–996. [CrossRef]
11. Zlot, R.; Bosse, M. Efficient large-scale three-dimensional mobile mapping for underground mines. *J. Field Robot.* **2014**, *31*, 758–779. [CrossRef]
12. Jung, J.; Yoon, S.; Ju, S.; Heo, J. Development of kinematic 3D laser scanning system for indoor mapping and as-built BIM using constrained SLAM. *Sensors* **2015**, *15*, 26430–26456. [CrossRef] [PubMed]
13. Du, Z.; Wu, Z.; Yang, J. Error ellipsoid analysis for the diameter measurement of cylindroid components using a laser radar measurement system. *Sensors* **2016**, *16*, 714. [CrossRef] [PubMed]

14. Jichen, C.; Xiu, W.; Jian, S.; Songlin, W.; Shuo, Y.; Chunjiang, Z. Development of real-time laser-scanning system to detect tree canopy characteristics for variable-rate pesticide application. *Int. J. Agric. Biol. Eng.* **2017**, *10*, 155–163. [CrossRef]
15. Reiser, D.; Vázquez-Arellano, M.; Paraforos, D.S.; Garrido-Izard, M.; Griepentrog, H.W. Iterative individual plant clustering in maize with assembled 2D LiDAR data. *Comput. Ind.* **2018**, *99*, 42–52. [CrossRef]
16. Zeng, Y.; Yu, H.; Dai, H.; Song, S.; Lin, M.; Sun, B.; Jiang, W.; Meng, M.Q.-H. An improved calibration method for a rotating 2D LiDAR system. *Sensors* **2018**, *18*, 497. [CrossRef]
17. Guo, Y.; Du, Y.; Du, Z.; Yao, Z. Calibration of 3D laser measurement system based on projective transformation. In Proceedings of the Sixth International Symposium on Precision Engineering Measurements and Instrumentation; International Society for Optics and Photonics, Hangzhou, China, 31 December 2010; Volume 7544, p. 75445I.
18. Kurnianggoro, L.; Hoang, V.-D.; Jo, K.-H. Calibration of a 2D laser scanner system and rotating platform using a point-plane constraint. *Comput. Sci. Inf. Syst.* **2015**, *12*, 307–322. [CrossRef]
19. So, E.W.Y.; Basso, F.; Menegatti, E. Calibration of a rorating 2d laser range-finder using point-plane coistraints. *J. Autom. Mob. Robot. Intell. Syst.* **2013**, *7*, 30–38.
20. Gao, Z.; Huang, J.; Yang, X.; An, P. Calibration of rotating 2D LIDAR based on simple plane measurement. *Sens. Rev.* **2019**, *39*, 190–198. [CrossRef]
21. Ang, K.H.; Chong, G.; Li, Y. PID control system analysis, design, and technology. *IEEE Trans. Control. Syst. Technol.* **2005**, *13*, 559–576.
22. Gundes, A.; Ozguler, A. PID stabilization of MIMO plants. *IEEE Trans. Autom. Control.* **2007**, *52*, 1502–1508. [CrossRef]
23. Wang, Q.-G.; Hang, C.C.; Yang, X.-P. Single-loop controller design via IMC principles. *Automatica* **2001**, *37*, 2041–2048. [CrossRef]
24. Zhang, R.; Wu, S.; Gao, F. Improved PI controller based on predictive functional control for liquid level regulation in a coke fractionation tower. *J. Process Control* **2014**, *24*, 125–132. [CrossRef]
25. Bassi, S.; Mishra, M.; Omizegba, E. Automatic tuning of proportional-integral-derivative (PID) controller using particle swarm optimization (PSO) algorithm. *Int. J. Artif. Intell. Appl.* **2011**, *2*, 25. [CrossRef]
26. Wang, Q.; Lu, C.; Pan, W. IMC PID controller tuning for stable and unstable processes with time delay. *Chem. Eng. Res. Des.* **2016**, *105*, 120–129. [CrossRef]
27. Normey-Rico, J.E.; Guzmán, J.L. Unified PID tuning approach for stable, integrative, and unstable dead-time processes. *Ind. Eng. Chem. Res.* **2013**, *52*, 16811–16819. [CrossRef]
28. Zadeh, L.A. Fuzzy sets. *Inf. Control* **1965**, *8*, 338–353. [CrossRef]
29. Chen, B.; Liu, X.P.; Ge, S.S.; Lin, C. Adaptive fuzzy control of a class of nonlinear systems by fuzzy approximation approach. *IEEE Trans. Fuzzy Syst.* **2012**, *20*, 1012–1021. [CrossRef]
30. Wang, F.; Liu, Z.; Zhang, Y.; Chen, C.P. Adaptive fuzzy control for a class of stochastic pure-feedback nonlinear systems with unknown hysteresis. *IEEE Trans. Fuzzy Syst.* **2015**, *24*, 140–152. [CrossRef]
31. Chiou, J.-S.; Tsai, S.-H.; Liu, M.-T. A PSO-based adaptive fuzzy PID-controllers. *Simul. Model. Pract. Theory* **2012**, *26*, 49–59. [CrossRef]
32. Savran, A.; Kahraman, G. A fuzzy model based adaptive PID controller design for nonlinear and uncertain processes. *ISA Trans.* **2014**, *53*, 280–288. [CrossRef] [PubMed]
33. He, W.; Chen, Y.; Yin, Z. Adaptive neural network control of an uncertain robot with full-state constraints. *IEEE Trans. Cybern.* **2015**, *46*, 620–629. [CrossRef] [PubMed]
34. Liu, L.; Liu, Y.-J.; Tong, S. Neural networks-based adaptive finite-time fault-tolerant control for a class of strict-feedback switched nonlinear systems. *IEEE Trans. Cybern.* **2018**, *49*, 2536–2545. [CrossRef]
35. Noriega, J.R.; Wang, H. A direct adaptive neural-network control for unknown nonlinear systems and its application. *IEEE Trans. Neural Netw.* **1998**, *9*, 27–34. [CrossRef]
36. Wang, Y.; Jin, Q.; Zhang, R. Improved fuzzy PID controller design using predictive functional control structure. *ISA Trans.* **2017**, *71*, 354–363. [CrossRef]

37. Nuchkrua, T.; Leephakpreeda, T. Fuzzy self-tuning PID control of hydrogen-driven pneumatic artificial muscle actuator. *J. Bionic Eng.* **2013**, *10*, 329–340. [CrossRef]
38. Su, H.; Hao, G.; Li, P.; Luo, X. Feed forward fuzzy PID controller for common-rail pressure control of diesel engine. In Proceedings of the 2010 International Conference on Measuring Technology and Mechatronics Automation, Changsha, China, 13–14 March 2010; Volume 2, pp. 264–267.