

## Article

# Neural Network Self-Tuning Control for a Piezoelectric Actuator

Wenjun Li <sup>1</sup>, Chen Zhang <sup>2</sup>, Wei Gao <sup>2</sup> and Miaolei Zhou <sup>2,\*</sup><sup>1</sup> College of Transportation, Jilin University, Changchun 130022, China; liwj@jlu.edu.cn<sup>2</sup> College of Communication Engineering, Jilin University, Changchun 130022, China; zhangchen19@mails.jlu.edu.cn (C.Z.); gaow@jlu.edu.cn (W.G.)

\* Correspondence: zml@jlu.edu.cn

Received: 11 May 2020; Accepted: 10 June 2020; Published: 12 June 2020

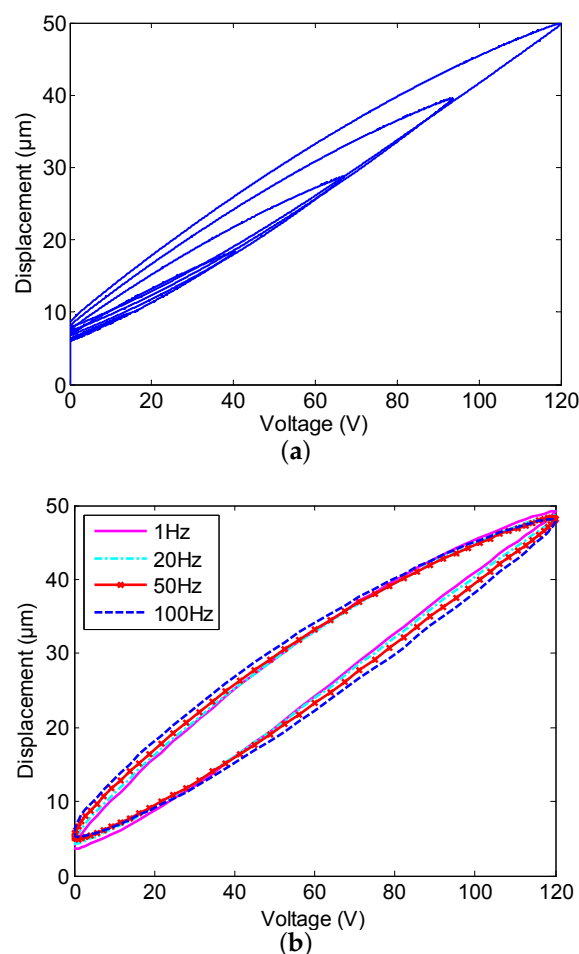
**Abstract:** Piezoelectric actuators (PEA) have been widely used in the ultra-precision manufacturing fields. However, the hysteresis nonlinearity between the input voltage and the output displacement, which possesses the properties of rate dependency and multivalued mapping, seriously impedes the positioning accuracy of the PEA. This paper investigates a control methodology without the hysteresis model for PEA actuated nanopositioning systems, in which the inherent drawback generated by the hysteresis nonlinearity aggregates the control accuracy of the PEA. To address this problem, a neural network self-tuning control approach is proposed to realize the high accuracy tracking with respect to the system uncertainties and hysteresis nonlinearity of the PEA. First, the PEA is described as a nonlinear equation with two variables, which are unknown. Then, using the capabilities of super approximation and adaptive parameter adjustment, the neural network identifiers are used to approximate the two unknown variables automatically updated without any off-line identification, respectively. To verify the validity and effectiveness of the proposed control methodology, a series of experiments is executed on a commercial PEA product. The experimental results illustrate that the established neural network self-tuning control method is efficient in damping the hysteresis nonlinearity and enhancing the trajectory tracking property.

**Keywords:** hysteresis; piezoelectric actuator; neural network; self-tuning

## 1. Introduction

A piezoelectric actuator (PEA) is a transducer that converts electrical energy into a mechanical displacement or stress based on a piezoelectric effect, or vice versa [1,2]. Compared with traditional electromechanical systems, the PEA actuated nanopositioning system has primary advantages [3–5] including the large mass ratio, sub-nanometer resolution, high positioning accuracy, and fast response characteristic; hence, it is extensively applied in atomic force microscope [6], ultra-precision mechanical control [7], biological manipulation [8], and other related fields. Nevertheless, the dominant challenge of the PEA derives from the hysteresis nonlinearity, which impedes the nanopositioning systems from obtaining the fine property notably [9]. Hysteresis nonlinearity is the non-smooth nonlinear behavior between the input voltage and the output displacement of the PEA [10,11]. Hysteresis nonlinearity is multivalued and nonlocal memoryless, which brings on the multivalued mapping behavior [12,13], as shown in Figure 1a. As the amplitude of the input signal increases, the input-output curves of the PEA will generate different hysteresis loops. Furthermore, the maximum value of the displacement changes from 40  $\mu\text{m}$  to 120  $\mu\text{m}$ , and this phenomenon indicates that the output displacement of the PEA is not only dependent on the current value of the input voltage, but also is related to the historical value. Beyond that, the work in [14] declared that the hysteresis nonlinearity is a rate-dependent behavior likewise, as shown in Figure 1b. The shape of the hysteresis loops becomes wider gradually as the frequency of the input voltage is increased from 1 Hz to 100 Hz, which means that the input

frequency of the driven signal is the inevitable factor to effect the output displacement. In addition, along with the growing of the input voltage frequency, the tracking errors caused by the hysteresis nonlinearity increase as well. To meet such a challenge, some control schemes have been proposed for the PEA during the past decade. The feedforward control approach is most widely used for compensating the hysteresis nonlinearity, structuring a controller by placing in a feedforward loop as a compensator with various models. These models include the Preisach model [15], the Prandtl–Ishlinskii model [16–18], the Krasnoselskii–Pokrovskii model [19], the Bouc–Wen model [20,21], the Duhem model [22,23], the Dahl model [24], and the neural network model [25,26] to suppress the undesirable behaviors. It is pointed out that the precision of open-loop control is affected by modeling errors concerning these hysteresis models; furthermore, open-loop control cannot suppress the influence of external interference.



**Figure 1.** Actual input-output hysteresis curves of a piezoelectric actuator (PEA). (a) multivalued mapping behavior; (b) rate-dependent behavior.

To overcome this problem, feedback control [27–30] is developed for high-precision control of the PEA. Al-Ghanimi et al. designed a fast non-singular terminal sliding mode controller based on perturbation estimation [27], and experimental results verified that this controller could track a reference signal precisely in the case of modeling error, external perturbation, and the system parameters' uncertainty. In [28], a composite control approach was developed for a piezo-actuated stage to track the desired reference trajectory accurately. An inverse multiplicative structure based feedforward hysteresis compensator was used to eliminate the hysteresis nonlinearity. Then, the linear model predictive control, which had a simple and clear solution, was adopted to achieve the precise tracking performance. On the basis of constructing a Bouc–Wen model to characterize the hysteresis

behavior of the PEA, Chen et al. introduced an internal model based controller, which guaranteed that the desired reference signal could be tracked under any initial conditions of the close-loop system [29]. The hysteresis nonlinearity of the PEA is either directly regarded as a disturbance or represented by the hysteresis models. By using the hysteresis model to describe the hysteresis nonlinearity, the inverse compensator with modified the Prandtl–Ishlinskii model [30] was designed to compensate for the hysteresis characteristic of a PEA, and a disturbance observer was used to compensate for the lumped disturbance and system uncertainties, then a tracking controller was adopted to further decrease errors. An adaptive tracking controller [31] was designed using the LuGre model and achieved fine performance in displacement tracking. In [32], sliding mode control with a sigmoid function was developed for the tracking control of the piezo-actuated stage, in which the Bouc–Wen model was adopted to describe the hysteresis nonlinearity. Similar to the control scheme in [32], the repetitive controller [33] was developed to improve the tracking capability of the piezo-actuated stage with the Prandtl–Ishlinskii model. By treating the hysteresis nonlinearity as a nonlinear bounded disturbance, an approximate hysteresis model that combined a variable gain and a variable time-delay term was proposed, and on the basis of this model, a Smith predictor based  $H_\infty$  controller [34] was designed to implement high-precision motion tracking control of a piezo-actuated stage. Meanwhile, an integrated PID based sliding mode controller [35] was presented to enhance control performance with a novel PID based sliding mode observer for the unknown disturbances. The designed observer relaxed the traditional matching conditions when establishing the unknown-input observers. Experimental results showed that the presented control method could estimate the PEA state precisely and attain an excellent tracking effect.

Most control methods are model based to compensate for the hysteresis nonlinearity. However, the uncertainties of the system models and the complex hysteresis nonlinear behavior make it difficult to describe adequately. Neural networks have powerful capabilities such as input-output mapping, function approximation, and adaptability [36–39], which can approximate the system dynamics and the hysteresis nonlinear behavior. In [40], a feedforward neural network controller was proposed to improve the tracking precision of the PEA, and the bat search algorithm was adopted to decrease the computational burden. Simulation results indicated that this developed method could considerably improve the steady-state error and settling time. Lin et al. designed an adaptive wavelet neural network controller with hysteresis estimation for a piezo-positioning mechanism in [41]. According to the theoretical analysis and experimental results, the asymptotic stability of the proposed controller was guaranteed. Furthermore, the robustness and the tracking performance were greatly improved. In [42], a control strategy of utilizing the neural network as the system feedback controller was proposed, and by conducting the simulations with the collected data, the effectiveness of the developed approach was validated and the positioning precision of the PEA could be improved greatly.

Self-tuning control is a kind of adaptive control technique, for which the parameters of the object are estimated online via the identifier, and then, these parameters can be adjusted automatically by the controller [43,44]. When applying the traditional self-tuning control, the object is generally described by a linear or linearized model to achieve the control effect. However, for a complex nonlinear system with the hysteresis behavior, such as the PEA, it is a challenging task to design the self-tuning control. The self-tuning control has the merits of accommodating the system uncertainty and suppressing random disturbances effectively. In [45], a neural network based self-tuning PID control was utilized to deal with the problem of the fixed gain existing in PID control, and this control approach was implemented on an underwater remotely-operated vehicle in real-time. Experimental results verified the effectiveness of the proposed method. The work in [46] introduced a self-tuning proportional double derivative-like neural network controller for an unmanned aerial vehicle in the presence of parametric uncertainties and unknown disturbances. Various kinds of self-tuning controllers have been successfully applied in different nonlinear systems and have shown excellent control performance. Some scholars have investigated the tracking problem of the PEA via self-tuning neuro-PID control [47] and fuzzy-reasoning based self-tuning PID control methods [48]. However, the positioning issue in

the presence of the hysteresis nonlinearity of the PEA has not been fully discussed by using the neural network based self-tuning control approach. Hence, in this paper, the self-tuning control is combined with the neural network to construct the neural network self-tuning controller for the PEA. Unlike the model based controllers, which need to obtain an accurate hysteresis model in advance, the PEA is described by the nonlinear function expression with two unknown variables, and two neural networks are applied as the identifiers to approximate the unknown variables of the PEA in online self-tuning control, so that the proposed controller can track the desired reference trajectory with high precision. Experiments are made on a commercial PEA product, and the results illustrate the control performance of the PEA in trajectory tracking.

The rest of this paper is organized as follows. In Section 2, the principle of the neural network self-tuning control algorithm is introduced, and the neural network identifiers are designed to achieve the high precision tracking for the PEA in the presence of hysteresis nonlinearity. Then, the experimental validations under the sinusoidal reference trajectories with various frequencies and the mixed triangular reference trajectory are conducted in Section 3. Finally, this paper is concluded in Section 4.

## 2. Controller Design

In practical operation, such as the micro-positioning system, the hysteresis nonlinearity that widely exists in the PEA can seriously deteriorate the systems performance and even leads to the oscillation and instability of the actual system. Therefore, the objective of our developed controller is to eliminate the hysteresis nonlinearity and achieve the high-precision tracking effect, so as to facilitate the further application of the PEA as a crucial driving mechanism. To realize this aim, the neural network self-tuning control scheme is developed in this paper.

### 2.1. Design Philosophy

The neural network has distinguished abilities of nonlinearity approximation and adaptiveness. Taking the shortcomings of the conventional self-tuning control into account, the neural network self-tuning control algorithm [49] is adopted in this paper to achieve the tracking precision of the PEA.

The PEA is described as the nonlinear equation as follows:

$$y(k+1) = g[y(k)] + f[y(k)]u(k) \quad (1)$$

where  $y(k)$  is the hysteresis output,  $u(k)$  is the input voltage, and  $g[y(k)]$  and  $f[y(k)]$  are nonlinear functions.  $k$  denotes the  $k^{\text{th}}$  training time.

When  $g[y(k)]$  and  $f[y(k)]$  are exactly known, according to Equation (1), the control law can be derived as follows:

$$u(k) = \frac{y(k+1) - g[y(k)]}{f[y(k)]} \quad (2)$$

The output  $y(k+1)$  of the PEA can exactly track the desired reference signal  $y_d(k+1)$ , namely:

$$y(k+1) = y_d(k+1) \quad (3)$$

Substituting Equation (3) into Equation (2), the control law Equation (2) is restated as follows:

$$u(k) = \frac{y_d(k+1) - g[y(k)]}{f[y(k)]} \quad (4)$$

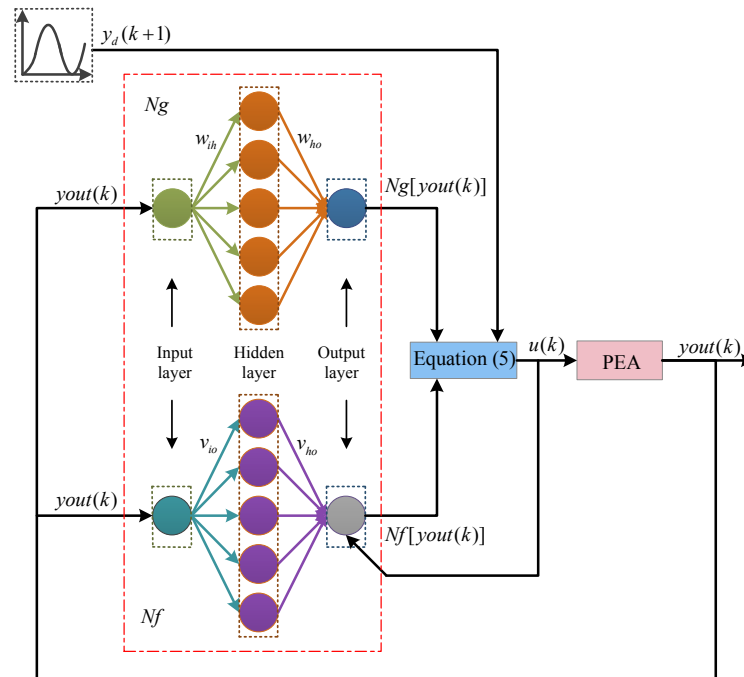
However, it is tough to describe the uncertainties of the PEA and its complex hysteresis nonlinear behavior sufficiently. Therefore,  $g[y(k)]$  and  $f[y(k)]$  are commonly unknown. To solve this problem,

two neural network identifiers  $Ng$  and  $Nf$  are adopted to approximate the nonlinear functions  $g[y(k)]$  and  $f[y(k)]$ , respectively. Then, the control law in Equation (4) is rewritten as follows:

$$u(k) = \frac{y_d(k+1) - Ng[y(k)]}{Nf[y(k)]} \quad (5)$$

where the estimated values of  $g[y(k)]$  and  $f[y(k)]$  are represented by  $Ng[y(k)]$  and  $Nf[y(k)]$ , respectively.

The control goal is that the hysteresis output  $y(k+1)$  can track a given signal  $y_d(k+1)$  exactly. The actual hysteresis output  $y(k)$  is fed into the input of back-propagation neural network identifiers  $Ng$  and  $Nf$ , which have the ability to “learn” system features based on nonlinear mapping. Both the input layer and output layer of the neural network identifiers adopt one neuron, and the variables of the neuron in the hidden layer are five. Then, the estimated control variable  $u(k)$  in Equation (5) is adopted to control the PEA. The block diagram of the proposed neural network self-tuning control scheme for the PEA is depicted in Figure 2.



**Figure 2.** The block diagram of the proposed neural network self-tuning control scheme for PEA.

The estimated hysteresis output  $\tilde{y}(k+1)$  of the PEA is expressed as follows:

$$\begin{aligned} \tilde{y}(k+1) &= Ng[y(k)] + Nf[y(k)]u(k) \\ &= Ng[y(k); W(k)] + Nf[y(k); V(k)]u(k) \end{aligned} \quad (6)$$

$$W(k) = [w_0, w_1(k), w_2(k), w_3(k), w_4(k), w_5(k)] \quad (7)$$

$$V(k) = [v_0, v_1(k), v_2(k), v_3(k), v_4(k), v_5(k)] \quad (8)$$

where  $W(k)$  and  $V(k)$  represent the weights of the two neural network identifiers  $Ng$  and  $Nf$ , respectively. The sigmoid Equation (9) is selected as the activation function of the hidden layer in neural network identifier  $Ng$ . The hyperbolic tangent Equation (10) is selected as the activation function of the hidden layer in neural network identifier  $Nf$ .

$$f_{Ng}(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (9)$$

$$f_{Nf}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (10)$$

Equation (5) is substituted into Equation (1), and the hysteresis output of the PEA is described as follows:

$$y(k+1) = f[y(k)] \frac{y_d(k+1) - Ng[y(k)]}{Nf[y(k)]} + g[y(k)] \quad (11)$$

It is obvious that when the following Equations (12) and (13) are satisfied, the actual hysteresis output  $y(k+1)$  can be infinitely close to the desired displacement output  $y_d(k+1)$ , namely  $y(k+1)$  is equivalent to  $y_d(k+1)$ .

$$Ng[y(k); W(k)] \rightarrow g[y(k)] \quad (12)$$

$$Nf[y(k); V(k)] \rightarrow f[y(k)] \quad (13)$$

## 2.2. Neural Network Identifiers $Ng$ and $Nf$

In the overall control algorithm, the input vector of input layer for neural network identifiers  $Ng$  and  $Nf$  is  $y(k)$ .  $W(k)$  and  $v(k)$  are the weight vectors for training in neural network identifiers  $Ng$  and  $Nf$ , respectively. Then, the optimal  $W(k)$  and  $v(k)$  can minimize the mean squared error formula, that is:

$$E(k) = \frac{1}{2} [y_d(k+1) - y(k+1)]^2 = \frac{1}{2} e^2(k+1) \quad (14)$$

where  $e(k+1)$  is the error between  $y_d(k+1)$  and  $y(k+1)$ .

Then, updated weight  $\Delta w$  in neural network identifier  $Ng$  and updated weight  $\Delta v$  in neural network identifier  $Nf$  are adjusted such that  $E(k)$  can be reduced. It can be verified as follows:

$$\Delta w(k) = \eta_w \frac{\partial E(k)}{\partial w(k)} = \eta_w \frac{\partial E(k)}{\partial y(k)} \frac{\partial y(k)}{\partial w(k)} \quad (15)$$

$$\Delta v(k) = \eta_v \frac{\partial E(k)}{\partial v(k)} = \eta_v \frac{\partial E(k)}{\partial y(k)} \frac{\partial y(k)}{\partial v(k)} \quad (16)$$

where  $\eta_w$  and  $\eta_v$  are the learning rate, which compensate for the weight adjustment error in neural network identifiers  $Ng$  and  $Nf$ , respectively.

Equations (11) and (14) are substituted into Equation (15), and the algorithm of weight adjustment in the neural network identifier  $Ng$  is derived as follows:

$$\begin{aligned} \Delta w(k) &= \eta_w \frac{\partial \left( \frac{1}{2} e^2(k+1) \right)}{\partial y(k)} \frac{\partial y(k)}{\partial w(k)} \\ &= \frac{\partial \left[ g[y(k)] + f[y(k)] \frac{y_d(k+1) - Ng[y(k); W(k)]}{Nf[y(k); V(k)]} \right]}{\partial y(k)} (-\eta_w) \frac{\partial y(k)}{\partial w(k)} e(k+1) \\ &= \eta_w \frac{f[y(k)]}{Nf[y(k); V(k)]} \frac{\partial Ng[y(k); W(k)]}{\partial w(k)} e(k+1) \end{aligned} \quad (17)$$

In the neural network identifier  $Ng$ ,  $w_{ih}$  represents the weight matrix between the input and the hidden layer, and  $w_{ho}$  represents the weight matrix between the hidden and the output layer.

The process of weight adjustment between the hidden and the output layer in the neural network identifier  $Ng$  is expressed as follows:

$$\begin{aligned} \frac{\partial Ng[y(k); W(k)]}{\partial w_{ho}} &= \frac{\partial Ng[y(k); W(k)]}{\partial Net1_o^i} \frac{\partial Net1_o^i}{\partial w_{ho}} \\ &= Ng[y(k); W(k)] OH1_h (1 - Ng[y(k); W(k)]) \end{aligned} \quad (18)$$

where  $Net1_o^i$  represents the input of the output layer node  $o$  in the neural network identifier  $Ng$  and  $OH1_h$  represents the output vector of the hidden layer. Equation (18) is substituted into Equation (17), then we can obtain:

$$\begin{aligned}\Delta w_{ho} &= \eta_w \frac{f[y(k)]}{Nf[y(k); V(k)]} \frac{\partial Ng[y(k); W(k)]}{\partial w_{ho}(k)} e(k+1) \\ &= \eta_w \frac{f[y(k)]}{Nf[y(k); V(k)]} Ng[y(k); W(k)] (1 - Ng[y(k); W(k)]) OH1_h e(k+1)\end{aligned}\quad (19)$$

Although  $f[y(k)]$  is unknown, it can be assumed that its notation is known. Beyond that, the positive and negative of  $f[y(k)]$  determine the convergence direction in the weight adjustment algorithm. Therefore,  $f[y(k)]$  is replaced by  $\text{sgn}\{f[y(k)]\}$  before (19) is used in the next update criterion.

$$\begin{aligned}w_{ho}(k) &= w_{ho}(k-1) + \Delta w_{ho}(k-1) + \alpha_1 (w_{ho}(k-1) - w_{ho}(k-2)) \\ &= w_{ho}(k-1) + \eta_w \frac{\text{sgn}\{f[y(k)]\}}{Nf[y(k); V(k)]} e(k+1) Ng[y(k); W(k)] (1 - Ng[y(k); W(k)]) OH1_h \\ &\quad + \alpha_1 (w_{ho}(k-1) - w_{ho}(k-2))\end{aligned}\quad (20)$$

where  $\alpha_1$  is a constant.

The process of weight adjustment between the input and the hidden layer in the neural network identifier  $Ng$  is described as follows:

$$\begin{aligned}\frac{\partial Ng[y(k); W(k)]}{\partial w_{ih}} &= \frac{\partial Ng[y(k); W(k)]}{\partial OH1_h} \frac{\partial OH1_h}{\partial Net1_h^i} \frac{\partial Net1_h^i}{\partial w_{ih}} \\ &= \frac{\partial Ng[y(k); W(k)]}{\partial OH1_h} OH1_h (1 - OH1_h) I_1\end{aligned}\quad (21)$$

where  $Net1_h^i$  represents the input of the hidden layer node  $h$  in the neural network identifier  $Ng$  and  $I_1$  is the input vector of the hidden layer.

$$\begin{aligned}\frac{\partial Ng[y(k); W(k)]}{\partial OH1_h} &= \frac{\partial Ng[y(k); W(k)]}{\partial Net1_o^i} \frac{\partial Net1_o^i}{\partial OH1_h} \\ &= Ng[y(k); W(k)] (1 - Ng[y(k); W(k)]) w_{ho}\end{aligned}\quad (22)$$

Equations (21) and (22) are substituted into Equation (17), and we can obtain:

$$\begin{aligned}\Delta w_{ih} &= \eta_w \frac{f[y(k)]}{Nf[y(k); V(k)]} \frac{\partial Ng[y(k); W(k)]}{\partial w_{ih}(k)} e(k+1) \\ &= Ng[y(k); W(k)] (1 - Ng[y(k); W(k)]) e(k+1) \eta_w \frac{f[y(k)]}{Nf[y(k); V(k)]} OH1_h (1 - OH1_h) I_1 w_{ho}\end{aligned}\quad (23)$$

In the same way,  $f[y(k)]$  is replaced by  $\text{sgn}\{f[y(k)]\}$ , and the learning rate  $\eta_w$  is used to compensate for the weight adjustment error caused by approximate substitution.

$$\begin{aligned}w_{ih}(k) &= w_{ih}(k-1) + \Delta w_{ih}(k-1) + \alpha_1 (w_{ih}(k-1) - w_{ih}(k-2)) \\ &= \eta_w \frac{\text{sgn}\{f[y(k)]\}}{Nf[y(k); V(k)]} Ng[y(k); W(k)] (1 - Ng[y(k); W(k)]) OH1_h (1 - OH1_h) I_1 e(k+1) w_{ho} \\ &\quad + w_{ih}(k-1) + \alpha_1 (w_{ih}(k-1) - w_{ih}(k-2))\end{aligned}\quad (24)$$



Similar to the progress solving for  $w_{ho}$  and  $w_{ih}$  of the neural network identifier  $Ng$ , we can obtain the weight matrix  $v_{ho}$  between the hidden and the output layer of the neural network identifier  $Nf$  and the weight adjustment  $v_{ih}$  between the input and the hidden layer of the neural network identifier  $Nf$ .

$$v_{ho}(k) = v_{ho}(k-1) + \eta_v \text{sgn}\{f[y(k)]\} (1 - Nf[y(k); V(k)]) OH2_h e(k+1) u(k) + \alpha_2 (v_{ho}(k-1) - v_{ho}(k-2)) \quad (25)$$

$$v_{ih}(k) = v_{ih}(k-1) + \eta_v \text{sgn}\{f[y(k)]\} e(k+1) u(k) v_{ho} (1 - Nf[y(k); V(k)]) OH2_h (1 - OH2_h) I_2 + \alpha_2 (v_{ih}(k-1) - v_{ih}(k-2)) \quad (26)$$

where  $OH2_h$  represents the output vector of the hidden layer in the neural network identifier  $Nf$  and  $\alpha_2$  is a constant.

### 3. Experimental Study

To confirm the efficiency of the proposed neural network self-tuning control method, plenty of experiments were executed on a commercial PEA product, in which the trajectory tracking performances were certified through a PC based control system. The experimental system, shown in Figure 3, was composed of a data acquisition (DAQ) interface (PCI-1710, Advantech, Taiwan, China), a voltage amplifier (HPV-3C, SuZhou Micro Automation Technology Co., Ltd, Suzhou China), a PEA (PZT 150/7×7/50, SuZhou Micro Automation Technology Co., Ltd, Suzhou China), a position sensor (MDSL-0500M6-1A, SuZhou Micro Automation Technology Co., Ltd, Suzhou China), and a host computer. The specifications of the considered PEA product are presented in Table 1. The input voltage for the PEA was sent to actuate the PEA by a digital-to-analog converter and amplified by a voltage amplifier later on. The displacement output of the PEA was detected by a position sensor and sampled by an analog-to-digital converter later on. The experiments were implemented by using the host computer with the MATLAB/Simulink software installed.

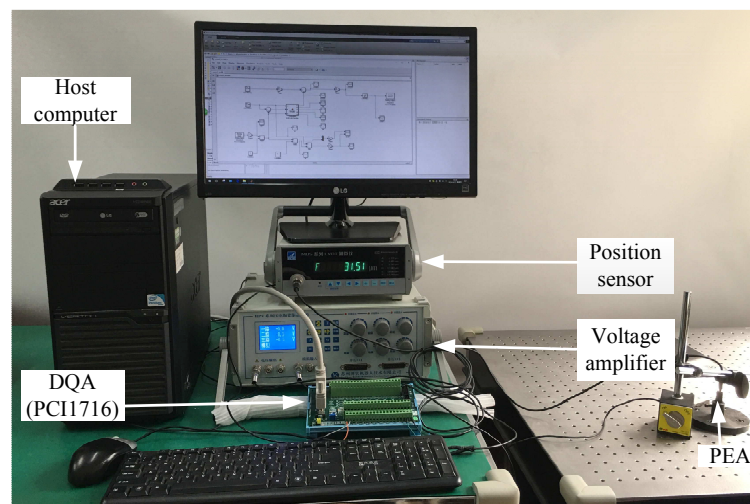


Figure 3. Picture of the experimental setup.



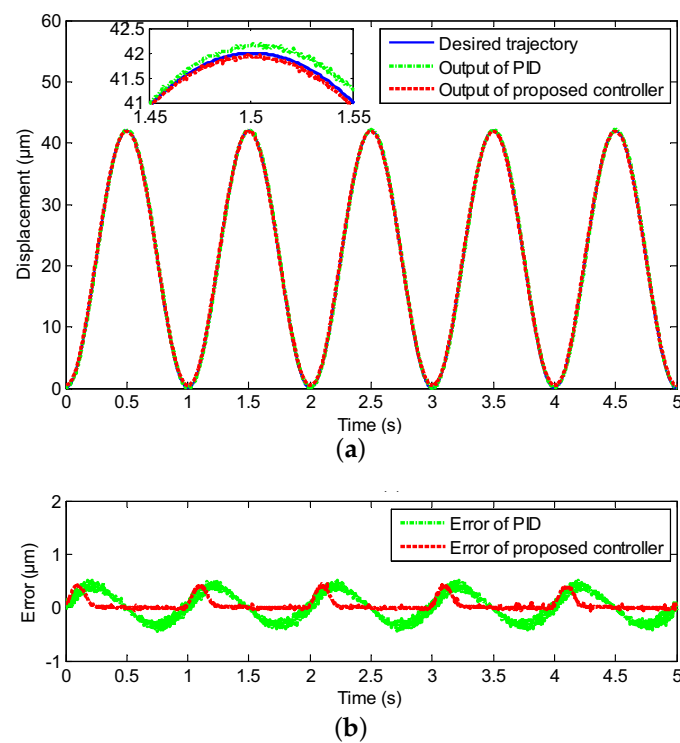
**Table 1.** Basic properties of the PEA product.

Basic Properties	Values	Units
Dimension	$7 \times 7 \times 42$	mm
Nominal displacement	$50 \pm 10\%$	$\mu\text{m}$
Maximum push force	1800	N
Stiffness	36	$\text{N}/\mu\text{m}$
Capacitance	$6.9 \pm 20\%$	$\mu\text{F}$

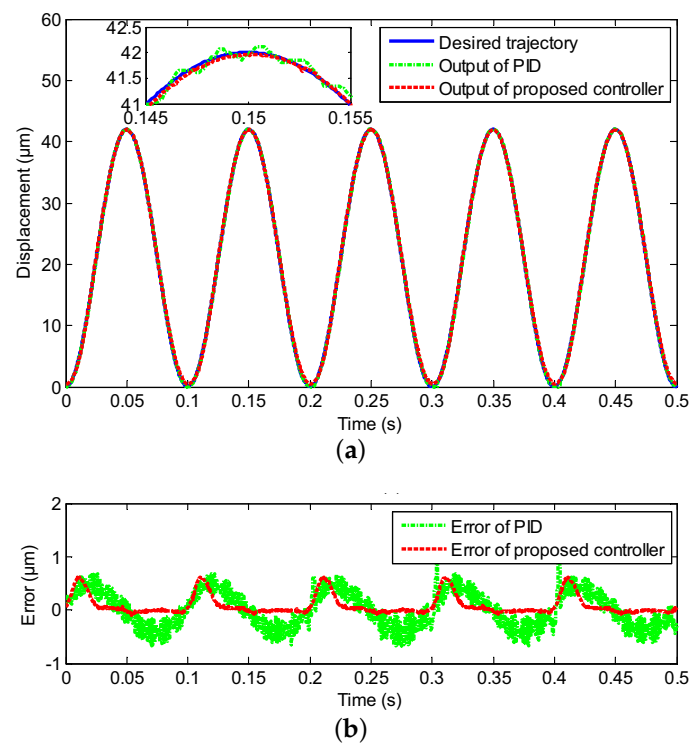
### 3.1. Tracking of Sinusoidal Trajectories with Diverse Frequencies

In this section, the experimental test is deployed to follow the sinusoidal signal under the various frequencies with the proposed neural network for self-tuning control approach, where the reference trajectories were  $y(t) = 21 + 21 \sin(2\pi\omega t - \frac{\pi}{2})$ ,  $\omega = 1, 10, 20, 40$ , respectively, and the classical PID control was utilized as a comparison. The compared experimental results, including controller output tracking curves and control errors, are clearly depicted in Figures 4–7, where excellent tracking accuracies of the proposed controller are demonstrated.

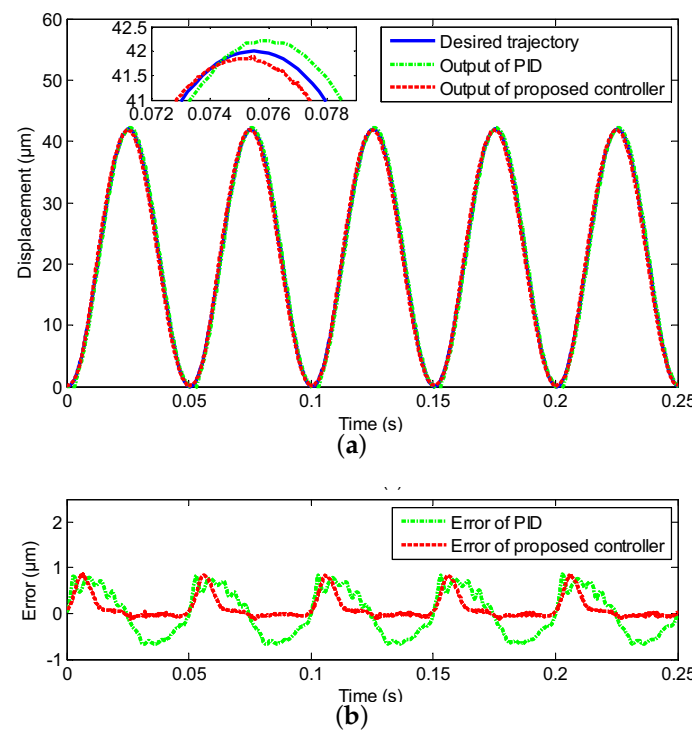
The real displacement outputs of the PEA are marked as the red line, and it can track the desired trajectories noted by the blue line effectively. The errors caused by the model uncertainties and hysteresis nonlinearity could be decreased owing to the proposed neural network self-tuning control approach. Combined with the above advantage, the trajectory tracking errors decreased to a satisfying range. Furthermore, the tracking errors were evaluated by two types of performance indicators, i.e., the root mean squared error (RMSE) and the maximum error (MAXE) were calculated to assess the experimental results. As shown in Table 2, the MAXEs of the proposed controller under the different input frequencies were 1.02%, 1.26%, 1.85%, and 2.45%, respectively, and they were 50%, 55%, 26%, and 25% less than those of the classical PID controller. In addition, the RMSEs were 0.1133  $\mu\text{m}$ , 0.2431  $\mu\text{m}$ , 0.2545  $\mu\text{m}$ , and 0.3148  $\mu\text{m}$  with diverse input frequencies of 1 Hz, 10 Hz, 20 Hz, and 40 Hz by the proposed neural network self-tuning control approach, and they were reduced by 49%, 22 %, 47%, and 40%, respectively, compared with the classical PID controller. The input-output relations of the neural network self-tuning control system are shown in Figure 8, which illustrates that the proposed scheme was valid in diverse frequencies. Furthermore, compared with the PID control method, the hysteresis nonlinearity of the PEA could be significantly suppressed by the proposed controller, and the high-precision tracking goal was achieved via our developed control approach. In addition, in spite of the tracking errors arising by the rate-dependent hysteresis nonlinearity being increased with the growth of frequency, the tracking errors were still within the allowable range. Moreover, according to Table 1, the nominal displacement of the PEA was  $(50 \pm 10\%) \mu\text{m}$ , and the maximum displacement in the experimental results was 42  $\mu\text{m}$ . This implied that our experimental range covered  $(84.84 \pm 8.48)\%$  of the nominal displacement, which demonstrated the rationality of our experiments for eliminating the hysteresis within this range. Therefore, it was obvious that excellent tracking accuracies of the proposed controller could be obtained, which demonstrated the effectiveness of the developed control method.



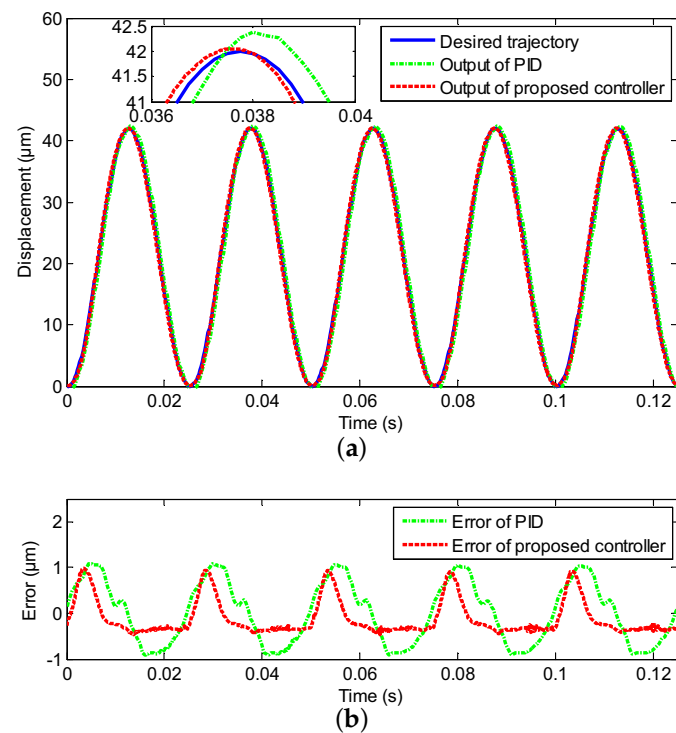
**Figure 4.** Compared tracking performance of the PEA under the 1 Hz sinusoidal reference, including (a) the controller output tracking curve and (b) control error.



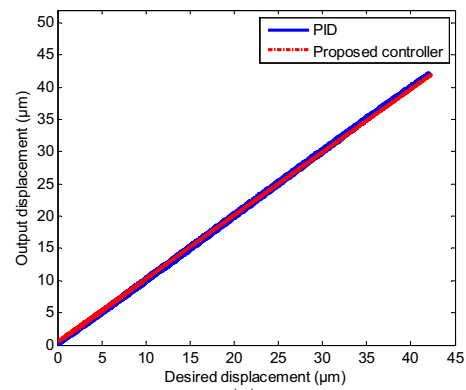
**Figure 5.** Compared tracking performance of the PEA under the 10 Hz sinusoidal reference, including (a) the controller output tracking curve and (b) control error.



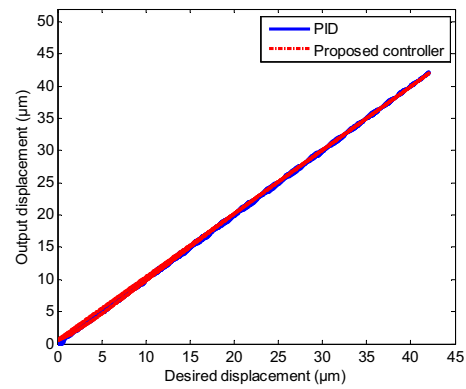
**Figure 6.** Compared tracking performance of the PEA under the 20 Hz sinusoidal reference, including (a) the controller output tracking curve and (b) control error.



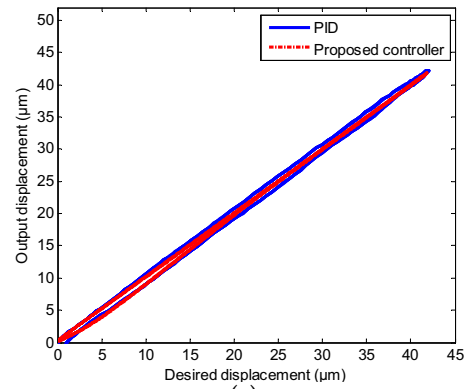
**Figure 7.** Compared tracking performance of the PEA under the 40 Hz sinusoidal reference, including (a) the controller output tracking curve and (b) control error.



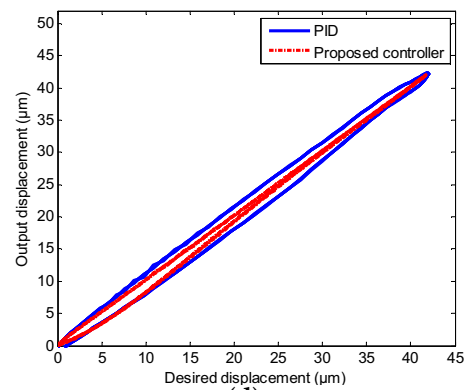
(a)



(b)



(c)



(d)

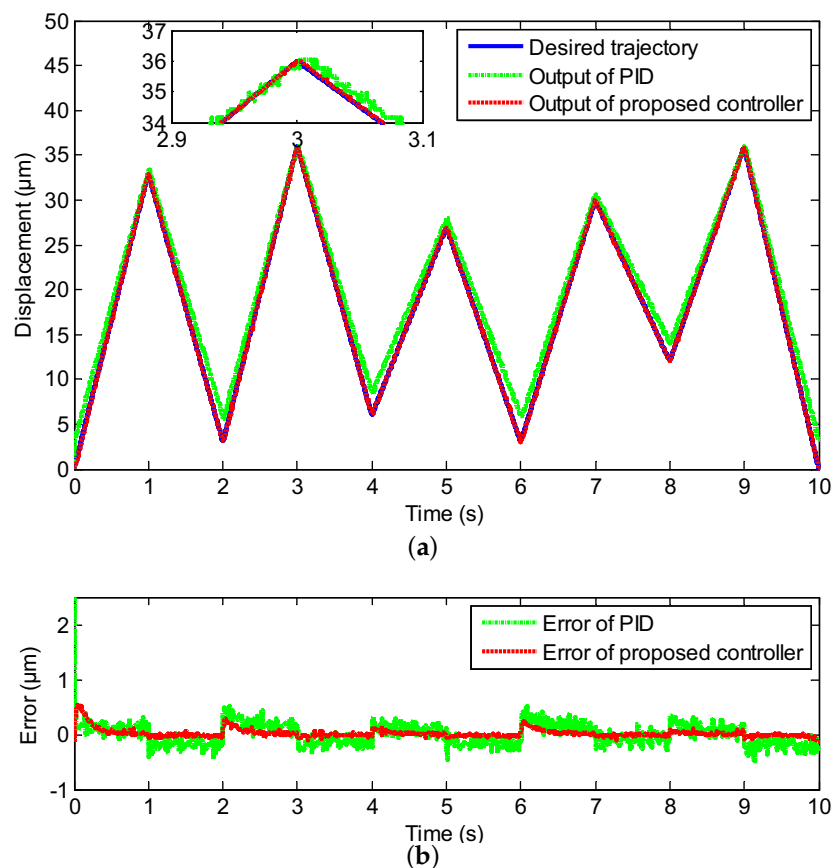
**Figure 8.** Compared input-output relations of the neural network self-tuning controller and PID controller under the sinusoidal trajectories with diverse input frequencies. (a) 1 Hz; (b) 10 Hz; (c) 20 Hz; (d) 40 Hz.

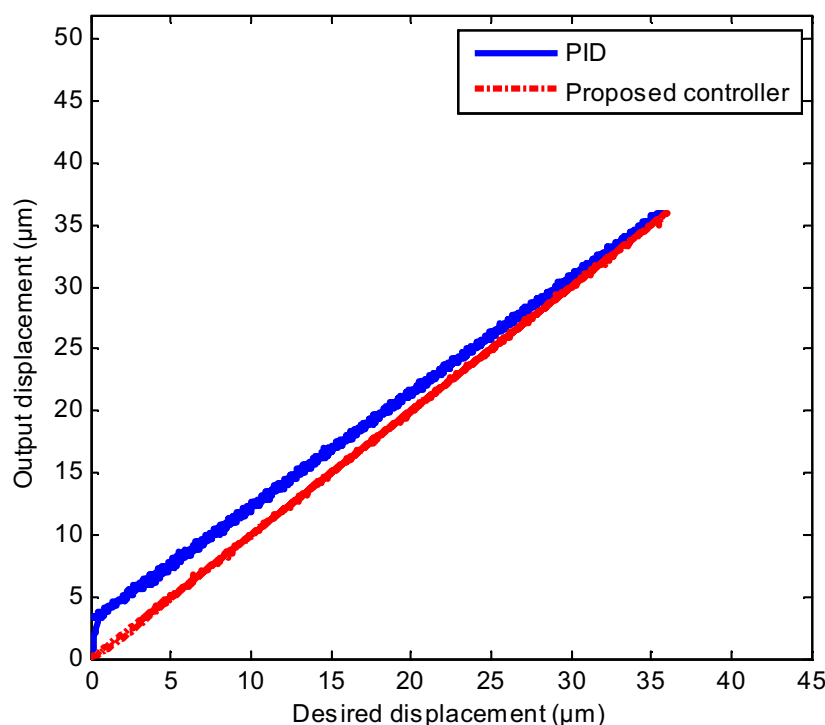
**Table 2.** Compared tracking errors under the sinusoidal trajectories with diverse input frequencies.

Input Frequencies	PID (RMSE ( $\mu\text{m}$ )/MAXE (%))	Proposed Controller (RMSE ( $\mu\text{m}$ )/MAXE (%))
1 Hz	0.2223/2.04	0.1133/1.02
10 Hz	0.3136/2.77	0.2431/1.26
20 Hz	0.4788/2.51	0.2545/1.85
40 Hz	0.5283/3.29	0.3148/2.45

### 3.2. Tracking of Mixed Triangular Trajectory

To further verify the excellent performance of the proposed method, a more complex trajectory tracking experiment was executed, where the desired trajectory was the mixed triangular wave with different amplitudes. The mixed triangular trajectory tracking result is depicted in Figure 9, where the real displacement outputs of the proposed controller marked with the red line could track the desired trajectory marked with the blue line well. Figure 10 represents the input-output relation of the neural network self-tuning controller and the PID controller. As shown in Figure 10, compared with the PID controller, the output displacement and the desired displacement could almost maintain a linear relationship, which manifested the validity of the proposed scheme when tracking the mixed triangular reference. The MAXE and RMSE of constructed controller were 1.56% and 0.1245  $\mu\text{m}$ , and compared with the PID controller, they were improved by 54% and 47%, respectively, which demonstrated that the neural network self-tuning control scheme was effective for the complex trajectory tracking requirement.

**Figure 9.** Compared tracking performance of the PEA under the mixed triangular reference, including (a) the controller output tracking curve and (b) control error .



**Figure 10.** Compared input-output relations of the neural network self-tuning controller and PID controller with the mixed triangular reference.

In the cases of sinusoidal reference trajectories with various frequencies and the mixed triangular reference trajectory with different amplitudes, the MAXEs and RMSEs were less than 2.45% and 0.3148  $\mu\text{m}$ , respectively. The performance indicators revealed that the constructed control strategy had a satisfactory tracking performance in terms of the different reference trajectories.

#### 4. Conclusions

In this paper, a neural network self-tuning control approach described as a nonlinear equation with two unknown variables was proposed to enhance the displacement tracking performance of the PEA. Using the prominent approximation capability of the neural networks, two unknown variables were automatically updated by the neural network identifiers without off-line identification, and then, the neural network self-tuning controller was constructed. The extremely distinguishable characteristic of the proposed strategy was that the hysteresis model was no longer a requisite, which avoided the complex process of the hysteresis modeling and the imprecise inversion issue occurred in inversion based control methods. In addition, the developed controller had the adaptive ability to adjust the system unknown functions owing to the neural network identifiers. Finally, the proposed control methodology was successfully implemented on a commercial PEA product, the comparative experimental results of which indicated that the neural network self-tuning controller had excellent performance in trajectory tracking.

**Author Contributions:** Conceptualization and methodology, W.L. and M.Z.; validation, C.Z. and W.G.; data analysis, C.Z.; investigation, C.Z. and W.G.; writing, original draft preparation, W.L.; writing, review and editing, W.L., C.Z., W.G., and M.Z.; supervision, W.G. and M.Z.; project administration, M.Z.; funding acquisition, M.Z. All authors read and agreed to the published version of the manuscript.

**Funding:** This work is funded by the National Natural Science Foundation of China under Grant 51675228 and the Program of Science and Technology Development Plan of Jilin Province of China under Grants 20180101052JC and 20190303020SF.

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



## References

1. Zhang, X.; Chen, X.; Zhu, G.; Su, C.Y. Output feedback adaptive motion control and its experimental verification for time-delay nonlinear systems with asymmetric hysteresis. *IEEE T. Ind. Electron.* **2020**, *67*, 6824–6834. [\[CrossRef\]](#)
2. Zhang, X.; Wang, Y.; Chen, X.; Su, C.Y.; Li, Z.; Wang, C.; Peng, Y. Decentralized adaptive neural approximated inverse control for a class of large-scale nonlinear hysteretic systems with time delays. *IEEE T. Syst. Man Cy.-S.* **2018**, *49*, 2424–2437. [\[CrossRef\]](#)
3. Xu, R.; Zhang, X.; Guo, H.; Zhou, M. Sliding mode tracking control with perturbation estimation for hysteresis nonlinearity of piezo-actuated stages. *IEEE Access* **2018**, *6*, 2840538. [\[CrossRef\]](#)
4. Wong, P.K.; Xu, Q.; Vong, C.M.; Wong, H.C. Rate-dependent hysteresis modeling and control of a piezostage using online support vector machine and relevance vector machine. *IEEE T. Ind. Electron.* **2012**, *59*, 1988–2001. [\[CrossRef\]](#)
5. Cheng, L.; Liu, W.; Hou, Z.G.; Huang, T.; Yu, J.; Tan, M. An adaptive Takagi–Sugeno fuzzy model based predictive controller for piezoelectric actuators. *IEEE T. Ind. Electron.* **2017**, *64*, 3048–3058. [\[CrossRef\]](#)
6. Leang, K.K.; Devasia, S. Design of hysteresis-compensating iterative learning control for piezo-positioners: Application to atomic force microscopes. *Mechatronics* **2006**, *16*, 141–158. [\[CrossRef\]](#)
7. Ho, S.T.; Jan, S.J. A piezoelectric motor for precision positioning applications. *Precis. Eng.* **2016**, *43*, 285–293. [\[CrossRef\]](#)
8. Avci, E.; Hattori, T.; Kamiyama, K.; Kojima, M.; Horade, M.; Mae, Y.; Arai, T. Piezo-actuated parallel mechanism for biological cell release at high speed. *Biomed. Microdevices* **2015**, *17*, 98. [\[CrossRef\]](#)
9. Xiao, S.; Li, Y. Modeling and high dynamic compensating the rate-dependent hysteresis of piezoelectric actuators via a novel modified inverse Preisach model. *IEEE T. Contr. Syst. T.* **2013**, *21*, 1549–1557. [\[CrossRef\]](#)
10. Zhang, X.; Wang, Y.; Wang, C.; Su, C.Y.; Li, Z.; Chen, X. Adaptive estimated inverse output-feedback quantized control for piezoelectric positioning stage. *IEEE T. Cybernetics* **2019**, *49*, 2106–2118. [\[CrossRef\]](#)
11. Ge, P.; Jouaneh, M. Tracking control of a piezoceramic actuator. *IEEE T. Contr. Syst. T.* **1996**, *4*, 209–216.
12. Gu, G.Y.; Zhu, L.M.; Su, C.Y.; Ding, H.; Fatikow, S. Modeling and control of piezo-actuated nanopositioning stages: A survey. *IEEE T. Autom. Sci. Eng.* **2014**, *13*, 313–332. [\[CrossRef\]](#)
13. Yu, Y.; Zhang, C.; Zhou, M. NARMAX model based hysteresis modeling of magnetic shape memory alloy actuators. *IEEE T. Nanotechnol.* **2020**, *19*, 1–4. [\[CrossRef\]](#)
14. Xu, R.; Zhou, M. A self-adaption compensation control for hysteresis nonlinearity in piezo-actuated stages based on Pi-sigma fuzzy neural network. *Smart Mater. Struct.* **2018**, *27*, 045002. [\[CrossRef\]](#)
15. Nguyen, P.B.; Choi, S.B.; Song, B.K. A new approach to hysteresis modelling for a piezoelectric actuator using Preisach model and recursive method with an application to open-loop position tracking control. *Sensor. Ac. A-Phys.* **2018**, *270*, 136–152. [\[CrossRef\]](#)
16. Gu, G.Y.; Zhu, L.M.; Su, C.Y. Modeling and compensation of asymmetric hysteresis nonlinearity for piezoceramic actuators with a modified Prandtl–Ishlinskii model. *IEEE T. Ind. Electron.* **2014**, *61*, 1583–1595. [\[CrossRef\]](#)
17. Yang, M.J.; Li, C.X.; Gu, G.Y.; Zhu, L.M. Modeling and compensating the dynamic hysteresis of piezoelectric actuators via a modified rate-dependent Prandtl–Ishlinskii model. *Smart Mater. Struct.* **2015**, *24*, 125006. [\[CrossRef\]](#)
18. Zhou, M.; Xu, R.; Zhang, Q.; Wang, Z.; Zhao, Y. Hybrid Control Method of Magnetically Controlled Shape Memory Alloy Actuator Based on Inverse Prandtl–Ishlinskii Model. *J. Electr. Eng. Technol.* **2016**, *11*, 1457–1465. [\[CrossRef\]](#)
19. Li, Z.; Shan, J.; Gabbert, U. Inverse compensation of hysteresis using Krasnoselskii–Pokrovskii model. *IEEE-ASME T. Mech.* **2018**, *23*, 966–971. [\[CrossRef\]](#)
20. Rakotondrabe, M. Bouc–Wen modeling and inverse multiplicative structure to compensate hysteresis nonlinearity in piezoelectric actuators. *IEEE T. Autom. Sci. Eng.* **2011**, *8*, 428–431. [\[CrossRef\]](#)
21. Wang, Y.; Zhang, C.; Wu, Z.; Gao, W.; Zhou, M. A hopfield neural network based Bouc–Wen model for magnetic shape memory alloy actuator. *AIP Adv.* **2020**, *10*, 015212. [\[CrossRef\]](#)
22. Lin, C.J.; Lin, P.T. Tracking control of a biaxial piezo-actuated positioning stage using generalized Duhem model. *Comput. Math. Appl.* **2012**, *64*, 766–787. [\[CrossRef\]](#)

23. Zhou, M.; Yang, P.; Wang, J.; Gao, W. Adaptive sliding mode control based on Duhem model for piezoelectric actuators. *IETE Tech. Rev.* **2016**, *33*, 557–568. [\[CrossRef\]](#)
24. Xu, Q.; Wong, P.K. Hysteresis modeling and compensation of a piezostage using least squares support vector machines. *Mechatronics* **2011**, *21*, 1239–1251. [\[CrossRef\]](#)
25. Zhou, M.; Wang, Y.; Xu, R.; Zhang, Q.; Zhu, D. Feed-forward control for magnetic shape memory alloy actuators based on the radial basis function neural network model. *J. Appl. Biomater. Func.* **2017**, *15*, 25–30. [\[CrossRef\]](#)
26. Zhou, M.; Zhang, Q. Hysteresis model of magnetically controlled shape memory alloy based on a PID neural network. *IEEE T. Magn.* **2015**, *51*, 1–4.
27. Al-Ghanimi, A.; Zheng, J.; Man, Z. A fast non-singular terminal sliding mode control based on perturbation estimation for piezoelectric actuators systems. *Int. J. Control* **2017**, *90*, 480–491. [\[CrossRef\]](#)
28. Ming, M.; Ling, J.; Feng, Z.; Xiao, X. A model prediction control design for inverse multiplicative structure based feedforward hysteresis compensation of a piezo nanopositioning stage. *Int. J. Precis. Eng. Man.* **2018**, *19*, 1699–1708. [\[CrossRef\]](#)
29. Chen, Z.; Zheng, J.; Zhang, H.T.; Ding, H. Tracking of piezoelectric actuators with hysteresis: A nonlinear robust output regulation approach. *Int. J. Robust Nonlin.* **2017**, *27*, 2610–2626. [\[CrossRef\]](#)
30. Gu, G.Y.; Zhu, L.M.; Su, C.Y. High-precision control of piezoelectric nanopositioning stages using hysteresis compensator and disturbance observer. *Smart Mater. Struct.* **2014**, *23*, 105007. [\[CrossRef\]](#)
31. Ounissi, A.; Yakoub, K.; Kaddouri, A.; Abdessemed, R. Robust adaptive displacement tracking control of a piezo-actuated stage. In Proceedings of the 2017 6th International Conference on Systems and Control (ICSC), Batna, Algeria, North Africa, 7–9 May 2017.
32. Xu, R.; Zhou, M. Sliding mode control with sigmoid function for the motion tracking control of the piezo-actuated stages. *Electron. Lett.* **2016**, *53*, 75–77. [\[CrossRef\]](#)
33. Shan, Y.; Leang, K.K. Accounting for hysteresis in repetitive control design: Nanopositioning example. *Automatica* **2012**, *48*, 1751–1758. [\[CrossRef\]](#)
34. Tsai, M.S.; Chen, J.S. Robust tracking control of a piezoactuator using a new approximate hysteresis model. *J. Dyn. Syst.-T. ASME* **2003**, *125*, 96–102. [\[CrossRef\]](#)
35. Peng, J.Y.; Chen, X.B. Integrated PID based sliding mode state estimation and control for piezoelectric actuators. *IEEE-ASME T. Mech.* **2014**, *19*, 88–99. [\[CrossRef\]](#)
36. Asif, A.R.; Waris, A.; Gilani, S.O.; Jamil, M.; Ashraf, H.; Shafique, M.; Niazi, I.K. Performance Evaluation of Convolutional Neural Network for Hand Gesture Recognition Using EMG. *Sensors* **2020**, *20*, 1642. [\[CrossRef\]](#) [\[PubMed\]](#)
37. Elkatatny, S.; Al-Abduljabbar, A.; Abdelgawad, K. A New Model for Predicting Rate of Penetration Using an Artificial Neural Network. *Sensors* **2020**, *20*, 2058. [\[CrossRef\]](#) [\[PubMed\]](#)
38. Chu, Z.; Zhu, D.; Yang, S.X. Observer based adaptive neural network trajectory tracking control for remotely operated vehicle. *IEEE T. Neur. Net. Lear.* **2017**, *28*, 1633–1645. [\[CrossRef\]](#)
39. Zhang, C.; Yu, Y.; Wang, Y.; Zhou, M. Takagi-Sugeno fuzzy neural network hysteresis modeling for magnetic shape memory alloy actuator based on modified bacteria foraging algorithm. *Int. J. Fuzzy Syst.* **2020**, *22*, 1314–1329. [\[CrossRef\]](#)
40. Svečko, R.; Kusić, D. Feedforward neural network position control of a piezoelectric actuator based on a BAT search algorithm. *Expert Syst. Appl.* **2015**, *42*, 5416–5423. [\[CrossRef\]](#)
41. Lin, F.J.; Shieh, H.J.; Huang, P.K. Adaptive wavelet neural network control with hysteresis estimation for piezo-positioning mechanism. *IEEE T. Neural Networ.* **2006**, *17*, 432–444. [\[CrossRef\]](#)
42. Kauppinen, M.; Rönning, J. Software based neural network assisted movement compensation for nanoresolution piezo actuators. In *Intelligent Robots and Computer Vision XXIX: Algorithms and Techniques*, 2nd ed.; SPIE: Burlingame, CA, USA, 2012; Volume 8301, p. 830102.
43. Tseng, C.L.; Wang, S.Y.; Chien, S.C.; Chang, C.Y. Development of a self-tuning TSK-fuzzy speed control strategy for switched reluctance motor. *IEEE T. Power Electr.* **2011**, *27*, 2141–2152. [\[CrossRef\]](#)
44. Zamani, A.A.; Tavakoli, S.; Etedali, S. Control of piezoelectric friction dampers in smart base-isolated structures using self-tuning and adaptive fuzzy proportional–derivative controllers. *J. Inter. Mat. Syst. Str.* **2017**, *28*, 1287–1302. [\[CrossRef\]](#)

- 45. Hernández-Alvarado, R.; García-Valdovinos, L.G.; Salgado-Jiménez, T.; Gómez-Espinosa, A.; Fonseca-Navarro, F. Neural network based self-tuning PID control for underwater vehicles. *Sensors* **2016**, *16*, 1429. [[CrossRef](#)]
- 46. Tran, T.T.; Ha, C. Self-tuning proportional double derivative-like neural network controller for a quadrotor. *Int. J. Aeronaut. Space* **2018**, *19*, 976–985. [[CrossRef](#)]
- 47. Liu, V.T. Self-tuning Neuro-PID controller for piezoelectric actuator. *Adv. Sci. Lett.* **2012**, *14*, 141–145. [[CrossRef](#)]
- 48. Jie, D.G.; Sun, L.N.; Qu, D.S.; Wang, L.; Cai, H.G. Fuzzy-reasoning based self-tuning PID control for piezoelectric micro-displacement system. *J. Harbin Inst. Technol.* **2005**, *37*, 145–147. [[CrossRef](#)]
- 49. Chen, F.C. Back-propagation neural networks for nonlinear self-tuning adaptive control. *IEEE Contr. Syst. Mag.* **1990**, *10*, 44–48. [[CrossRef](#)]



© 2020 by the authors. 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 (<http://creativecommons.org/licenses/by/4.0/>).