SCIENCE CHINA
Information Sciences

• **Supplementary File** •

A novel frequency-dependent hysteresis model based on improved neural Turing machine

Yinan WU[1,2], Yongchun FANG[1,2*], Zhi FAN[1,2] & Cunhuan LIU[1,2]

[1]*Institute of Robotics and Automatic Information System, College of Artificial Intelligence, Nankai University, Tianjin* 300350*, China;*
[2]*Tianjin Key Laboratory of Intelligent Robotics, Tianjin* 300350*, China*

Appendix A Model parameter identification

Parameter identification for the frequency module

The relationship between frequency and hysteresis area is established in the frequency module as follows:

$$\hat{a} = \tanh\Big(\sum_{m=1}^{M}(\theta_m h_m) + \lambda\Big) = \tanh\Big(\sum_{m=1}^{M}(\theta_m \tanh(\hat{\theta}_m f + \hat{\lambda}_m)) + \lambda\Big). \tag{A1}$$

Furthermore, to determine the specific values of the weight and bias in the BPNN, a training data set, consisting of the input voltages and hysteresis area at different frequencies, is constructed as follows:

$$F = \{f_1, f_2, \ldots, f_N\},\ A = \{a_1, a_2, \ldots, a_N\}, \tag{A2}$$

where $f_1 < f_2 < \cdots < f_N$ denote different frequencies, $a_1 < a_2 < \cdots < a_N$ represent the corresponding hysteresis area of a PEA measured by a displacement sensor.

For parameter identification, a cost function is first defined as follows:

$$E_a = \frac{1}{2}\sum_{n=1}^{N}(\hat{a}_n - a_n)^2, \tag{A3}$$

where $\hat{a}_n$ denotes the hysteresis area produced by the BPNN model.

Moreover, the weight and bias in the BPNN are updated as follows:

$$\begin{aligned}\theta_m^{(z+1)} &= \theta_m^{(z)} - \alpha\frac{\partial E_a^{(z)}}{\partial \theta_m^{(z)}} = \theta_m^{(z)} - \alpha\sum_{n=1}^{N}\Big[(\hat{a}_n^{(z)} - a_n)\frac{\partial \hat{a}_n^{(z)}}{\partial \theta_m^{(z)}}\Big] \\ &= \theta_m^{(z)} - \alpha\sum_{n=1}^{N}\Big[(\hat{a}_n^{(z)} - a_n)(1 - {\hat{a}_n^{(z)}}^2)\tanh(\hat{\theta}_m^{(z)} f_n + \hat{\lambda}_m^{(z)})\Big],\end{aligned} \tag{A4}$$

$$\begin{aligned}\lambda^{(z+1)} &= \lambda^{(z)} - \alpha\frac{\partial E_a^{(z)}}{\partial \lambda^{(z)}} = \lambda^{(z)} - \alpha\sum_{n=1}^{N}\Big[(\hat{a}_n^{(z)} - a_n)\frac{\partial \hat{a}_n^{(z)}}{\partial \lambda^{(z)}}\Big] \\ &= \lambda^{(z)} - \alpha\sum_{n=1}^{N}\Big[(\hat{a}_n^{(z)} - a_n)(1 - {\hat{a}_n^{(z)}}^2)\Big],\end{aligned} \tag{A5}$$

$$\begin{aligned}\hat{\theta}_m^{(z+1)} &= \hat{\theta}_m^{(z)} - \alpha\frac{\partial E_a^{(z)}}{\partial \hat{\theta}_m^{(z)}} = \hat{\theta}_m^{(z)} - \alpha\frac{\partial E_a^{(z)}}{\partial h_m^{(z)}}\frac{\partial h_m^{(z)}}{\partial \hat{\theta}_m^{(z)}} \\ &= \hat{\theta}_m^{(z)} - \alpha\sum_{n=1}^{N}\Big[(\hat{a}_n^{(z)} - a_n)(1 - {\hat{a}_n^{(z)}}^2)\theta_m^{(z)}\big(1 - \tanh^2(\hat{\theta}_m^{(z)} f_n + \hat{\lambda}_m^{(z)})\big) f_n\Big],\end{aligned} \tag{A6}$$

$$\begin{aligned}\hat{\lambda}_m^{(z+1)} &= \hat{\lambda}_m^{(z)} - \alpha\frac{\partial E_a^{(z)}}{\partial \hat{\lambda}_m^{(z)}} = \hat{\lambda}_m^{(z)} - \alpha\frac{\partial E_a^{(z)}}{\partial h_m^{(z)}}\frac{\partial h_m^{(z)}}{\partial \hat{\lambda}_m^{(z)}} \\ &= \hat{\lambda}_m^{(z)} - \alpha\sum_{n=1}^{N}\Big[(\hat{a}_n^{(z)} - a_n)(1 - {\hat{a}_n^{(z)}}^2)\theta_m^{(z)}\big(1 - \tanh^2(\hat{\theta}_m^{(z)} f_n + \hat{\lambda}_m^{(z)})\big)\Big],\end{aligned} \tag{A7}$$

where α denotes the training step size, z stands for the iteration number, the weight and bias are initialized to 0. The update procedure stops when the value of the cost function is less than the specified threshold ϵ, where ϵ is a positive number close to 0.

Parameter identification for the nonlinear module

Compared with a BPNN, the modeling efficiency of an ELM is significantly improved by only training $\hat{\delta}_{\hat{m}}$, since the hidden nodes are determined by randomly setting the values of $\delta_{p\hat{m}}$ and $\gamma_{\hat{m}}$. To be specific, for a set of training data $\{i_k, o_k\}$, $i_k =$

* Corresponding author (email: fangyc@nankai.edu.cn)

$[i_{1k}, i_{2k}, i_{3k}]^T$, $k \in \{1, 2, \ldots, K\}$, o_k is the actual displacement measured by a capacitive sensor, K denotes the number of training samples. The calculation of the ELM output is transformed into the form of matrix operation as follows:

$$\hat{o}_k = \sum_{\hat{m}=1}^{\hat{M}} \Big(\hat{\delta}_{\hat{m}} \sin(\Delta_{\hat{m}} i_k + \gamma_{\hat{m}}) \Big), \tag{A8}$$

where $\Delta_{\hat{m}} = [\delta_{1\hat{m}}, \delta_{2\hat{m}}, \delta_{3\hat{m}}]$, $\hat{o}_k$ is initialized to 0 since the initial value of $\hat{\delta}_{\hat{m}}$ is set to 0.

To determine the value of $\hat{\delta}_{\hat{m}}$, the goal of model training is to minimize the error between $\hat{o}_k$ and o_k as follows:

$$\hat{\delta}_{\hat{m}} = \operatorname{argmin} \sum_{k=1}^{K} \Big\| \hat{o}_k - o_k \Big\|. \tag{A9}$$

Therefore, it can be obtained that

$$\sum_{\hat{m}=1}^{\hat{M}} \Big(\hat{\delta}_{\hat{m}} \sin(\Delta_{\hat{m}} i_k + \gamma_{\hat{m}}) \Big) = o_k, \tag{A10}$$

which is then transformed into matrix operations as follows:

$$\hat{H}\hat{\Delta} = O, \tag{A11}$$

where

$$\begin{aligned} \hat{H} &= \begin{bmatrix} \sin(\Delta_1 i_1 + \gamma_1) & \sin(\Delta_2 i_1 + \gamma_2) & \cdots & \sin(\Delta_{\hat{M}} i_1 + \gamma_{\hat{M}}) \\ \sin(\Delta_1 i_2 + \gamma_1) & \sin(\Delta_2 i_2 + \gamma_2) & \cdots & \sin(\Delta_{\hat{M}} i_2 + \gamma_{\hat{M}}) \\ \vdots & \vdots & \cdots & \vdots \\ \sin(\Delta_1 i_K + \gamma_1) & \sin(\Delta_2 i_K + \gamma_2) & \cdots & \sin(\Delta_{\hat{M}} i_K + \gamma_{\hat{M}}) \end{bmatrix}, \\ \hat{\Delta} &= \begin{bmatrix} \hat{\delta}_1 & \hat{\delta}_2 & \cdots & \hat{\delta}_{\hat{M}} \end{bmatrix}^T, \quad O = \begin{bmatrix} o_1 & o_2 & \cdots & o_K \end{bmatrix}^T. \end{aligned} \tag{A12}$$

Furthermore, $\hat{\delta}_{\hat{m}}$ is calculated as follows:

$$\hat{\Delta} = \hat{H}^{\dagger} O, \tag{A13}$$

where $\hat{H}^{\dagger}$ denotes the Moore-Penrose generalized inverse matrix of $\hat{H}$.

For the BPNN in the nonlinear module, $\varphi_{\check{m}}$, $\omega_{\check{m}}$, $\breve{\varphi}_{\check{m}}$ and $\breve{\omega}$ are determined by the error back propagation algorithm. The training data are constructed as $\{\hat{o}_k, o_k\}$, $k \in \{1, 2, \ldots, K\}$, where $\hat{o}_k$ is the output displacement of the ELM, o_k is the actual displacement measured by a capacitive sensor. Hence, the cost function is defined as follows:

$$E_o = \frac{1}{2} \sum_{k=1}^{K} (\check{o}_k - o_k)^2, \tag{A14}$$

where $\check{o}_k$ denotes the output displacement produced by the nonlinear module, which is calculated as follows:

$$\check{o} = \tanh\Big(\sum_{\check{m}=1}^{\check{M}} (\breve{\varphi}_{\check{m}} \check{h}_{\check{m}}) + \breve{\omega} \Big) = \tanh\Big(\sum_{\check{m}=1}^{\check{M}} \big(\breve{\varphi}_{\check{m}} \tanh(\varphi_{\check{m}} \hat{o} + \omega_{\check{m}}) \big) + \breve{\omega} \Big). \tag{A15}$$

The weight and bias are then updated as follows:

$$\begin{aligned} \breve{\varphi}_{\check{m}}^{(z+1)} &= \breve{\varphi}_{\check{m}}^{(z)} - \beta \frac{\partial E_o^{(z)}}{\partial \breve{\varphi}_{\check{m}}^{(z)}} = \breve{\varphi}_{\check{m}}^{(z)} - \beta \sum_{k=1}^{K} \Big[(\check{o}_k^{(z)} - o_k) \frac{\partial \check{o}_k^{(z)}}{\partial \breve{\varphi}_{\check{m}}^{(z)}} \Big] \\ &= \breve{\varphi}_{\check{m}}^{(z)} - \beta \sum_{k=1}^{K} \Big[(\check{o}_k^{(z)} - o_k)(1 - {\check{o}_k^{(z)}}^2) \tanh(\varphi_{\check{m}}^{(z)} \hat{o}_k + \omega_{\check{m}}^{(z)}) \Big], \end{aligned} \tag{A16}$$

$$\begin{aligned} \breve{\omega}^{(z+1)} &= \breve{\omega}^{(z)} - \beta \frac{\partial E_o^{(z)}}{\partial \breve{\omega}^{(z)}} = \breve{\omega}^{(z)} - \beta \sum_{k=1}^{K} \Big[(\check{o}_k^{(z)} - o_k) \frac{\partial \check{o}_k^{(z)}}{\partial \breve{\omega}^{(z)}} \Big] \\ &= \breve{\omega}^{(z)} - \beta \sum_{k=1}^{K} \Big[(\check{o}_k^{(z)} - o_k)(1 - {\check{o}_k^{(z)}}^2) \Big], \end{aligned} \tag{A17}$$

$$\begin{aligned} \varphi_{\check{m}}^{(z+1)} &= \varphi_{\check{m}}^{(z)} - \beta \frac{\partial E_o^{(z)}}{\partial \varphi_{\check{m}}^{(z)}} = \varphi_{\check{m}}^{(z)} - \beta \frac{\partial E_o^{(z)}}{\partial \check{h}_{\check{m}}^{(z)}} \frac{\partial \check{h}_{\check{m}}^{(z)}}{\partial \varphi_{\check{m}}^{(z)}} \\ &= \varphi_{\check{m}}^{(z)} - \beta \sum_{k=1}^{K} \Big[(\check{o}_k^{(z)} - o_k)(1 - {\check{o}_k^{(z)}}^2) \breve{\varphi}_{\check{m}}^{(z)} \big(1 - \tanh^2(\varphi_{\check{m}}^{(z)} \hat{o}_k + \omega_{\check{m}}^{(z)})\big) \hat{o}_k \Big], \end{aligned} \tag{A18}$$

$$\omega_{\check{m}}^{(z+1)} = \omega_{\check{m}}^{(z)} - \beta \frac{\partial E_o^{(z)}}{\partial \omega_{\check{m}}^{(z)}} = \omega_{\check{m}}^{(z)} - \beta \frac{\partial E_o^{(z)}}{\partial \check{h}_{\check{m}}^{(z)}} \frac{\partial \check{h}_{\check{m}}^{(z)}}{\partial \omega_{\check{m}}^{(z)}}$$

$$= \omega_{\check{m}}^{(z)} - \beta \sum_{k=1}^{K} \Big[(\check{o}_k^{(z)} - o_k)(1 - \check{o}_k^{(z)^2}) \check{\varphi}_{\check{m}}^{(z)} \big(1 - \tanh^2(\varphi_{\check{m}}^{(z)} \hat{o}_k + \omega_{\check{m}}^{(z)})\big) \Big], \tag{A19}$$

where β stands for the training step size, the weight and bias are initialized to 0. The update procedure stops when E_o is less than a positive threshold $\check{\epsilon}$, where $\check{\epsilon}$ is close to 0.

Appendix B Performance analysis of the nonlinear module

As an improved BPNN with a single hidden layer, an ELM is characterized in that the weight and bias of the hidden layer are given randomly or artificially without the need to be updated through training, and only output weights need to be calculated in the training process, thus powerfully improving modeling efficiency when compared with a BPNN. However, one of the difficulties in using an ELM is to achieve the desired high-precision since the model parameters are not obtained through iterative training. Therefore, in the nonlinear module, a novel neural network structure is designed by combining the respective advantages of a BPNN and an ELM, thereby ensuring modeling accuracy while shortening time-consumption for model training.

To demonstrate the advantage of the proposed structure compared with the nonlinear module containing a solitary ELM or a solitary BPNN, the comparative experiments between the proposed model and the solitary ELM/BPNN structure are implemented with detailed analysis regarding computational cost and modeling accuracy. For the training data with the maximum displacement of 10 μm, the goal of model training is to make the mean square error between the model output and the true displacement converge to 1×10^{-4} μm. For the nonlinear module in the proposed model, the hidden layers of the ELM and the BPNN both include 10 nodes.

Specifically, when adopting a solitary BPNN to serve as the nonlinear module, the structure of the BPNN is set to be the same as that in the proposed model, which contains 10 hidden layer nodes. The mean square error curves of the proposed model and the solitary BPNN structure are plotted in Figure B1, which demonstrate that the proposed model can converge to the set goal in fewer iterations than the solitary BPNN structure. Since the structure of the BPNN in the two models is exactly the same, it can be approximately considered that the training time of each iteration is the same for the two models, which verifies that the proposed model has less computational cost than the solitary BPNN structure.

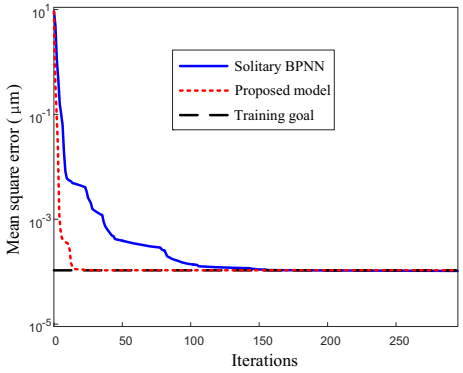

Figure B1 The mean square error of the model with a solitary BPNN and the proposed model.

Furthermore, for the nonlinear module containing a solitary ELM, the number of the hidden layer nodes is set to increase from 10 to 100. The mean square error is calculated for different numbers of the nodes with the results shown in Figure B2, which indicates that the mean square error of the solitary ELM structure cannot converge to the set goal (1×10^{-4} μm) as the number of nodes increases. The reason for this problem is the lack of training mechanism to improve modeling accuracy for the ELM.

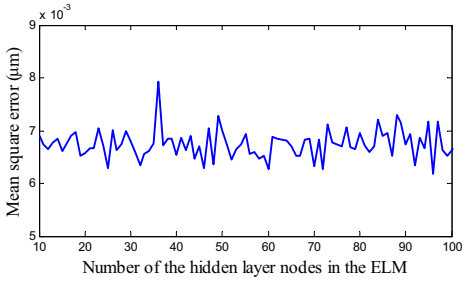

Figure B2 The mean square error of the solitary ELM structure containing different hidden layer nodes.

Through the above analysis, the nonlinear module containing an ELM and a BPNN is more helpful to balance computational cost and modeling accuracy.

Appendix C Experiment and analysis

In this paper, a single-tube piezoelectric actuator, provided by Benyuan company with the movement range of 20 μm$\times$20 μm$\times$4 μm in x-y-z direction, is adopted to establish the hysteresis model by the proposed method, and the output displacements are measured by a capacitive sensor supplied by ADE Technologies with the bandwidth of 10 kHz and the sub-nanometer resolution. Figure C1 exhibits the actual experimental platform, where the capacitive sensor consists of a measuring head, a measuring unit, a holder and a calibration block. The calibration block is fixed on the PEA to reflect its displacement, which is detected by the measuring head

and recorded by the measuring unit. The configuration of the computer used for the experiment is Intel(R) Core(TM) i7-10510U CPU @ 2.30 GHz with the memory of 8 GB.

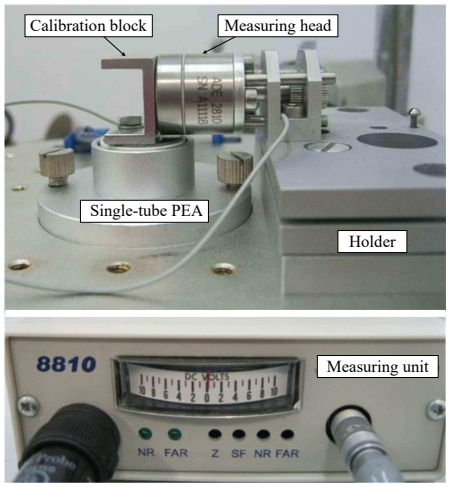

Figure C1 The adopted single-tube PEA and the displacement acquisition system.

Model Training Results

To demonstrate the performance of the proposed modeling method, multiple measured data of the adopted PEA, including input voltages and the corresponding displacements, are utilized to train the model and thus determine the model parameters. The training data cover the voltages at different frequencies in the range of $\{5\text{ Hz}, 15\text{ Hz}, 25\text{ Hz}, \ldots, 95\text{ Hz}\}$. [1)]

In addition, a widely used frequency-independent Preisach hysteresis model [1], and an advanced frequency-dependent fuzzy hysteresis model reported in [2], are built by using the same training data for comparative experiments, so as to convincingly verify the performance of the proposed method. For the Preisach model, it can accurately describe hysteresis by increasing the number of superposed operators. Compared with the data-driven hysteresis model, the Preisach model has better interpretability and expansibility. However, it is unable to directly simulate frequency-dependent hysteresis due to lack of frequency information in the model, and its integral term and analytical irreversibility is difficult to process. For the fuzzy model in [2], it adopts a Takagi-Sugeno fuzzy system based approach to simulate hysteresis, which has generalization ability since it is able to describe the change tendency and extract the dynamic property of frequency-dependent hysteresis.

It might be noted that, since the Preisach model is frequency-independent, the parameter identification needs to be repeatedly implemented for the data with different frequencies. Therefore, ten specific Preisach models with different parameters are obtained in the model training process due to the ten sets of training data with different frequencies. On the contrary, the proposed model, together with the fuzzy model, obtained by training has only a specific set of parameters owing to the frequency-dependence. The training time for the proposed model is 0.03 seconds, which is much less than 36.06 seconds of the Preisach model and 0.21 seconds of the fuzzy model.

The hysteresis modeling results are presented in Figure C2(a), where the blue solid curve denotes the measured displacements corresponding to the voltages with different frequencies, the black dot dash curve represents the displacements of the trained Preisach model, the green dotted curve stands for the output displacements of the fuzzy model, and the red dashed curve corresponds to the training results of the proposed model. Moreover, Figure C2(b) shows the absolute value of the relative error between the measurements and the displacements produced by the three models. In particular, the maximum absolute value of the relative error for the proposed model is 0.46%, less than 0.77% of the Preisach model and 0.61% of the fuzzy model. In addition, the mean absolute value of the relative error is 0.09% for the proposed method, which is also less than that of the Preisach model (0.22%) and the fuzzy model (0.18%). It can be seen from the results that, compared with the Preisach model and the fuzzy model, the output displacements of the proposed model are closer to the measurements with less training time, thereby indicating the satisfactory performance of the proposed modeling method.

Furthermore, the voltage-strain representations of the training results are shown in Figure C3, where Figure C3(a)-(j) present the hysteresis loops at different frequencies, the blue solid loops correspond to the actual measurements, the black dot dash loops and the green dotted loops are respectively plotted based on the training results of the Preisach model and the fuzzy model, the red dashed curves denote the hysteresis loops produced by the proposed model. Meanwhile, to quantitatively reflect the voltage-strain representations of the different modeling methods, the hysteresis area is calculated for different loops with the results provided in Table C1. It can be seen from the voltage-strain representations, together with the hysteresis area, that the proposed method is able to simulate frequency-dependent hysteresis more accurately.

Model Testing Results

After completing the model training, model testing is carried out to demonstrate the applicability of the trained models for other untrained data. The frequencies of the testing voltages cover the range of $\{10\text{ Hz}, 20\text{ Hz}, 30\text{ Hz}, \ldots, 90\text{ Hz}\}$. The Preisach model, trained by the data with frequency of 5 Hz, is selected to perform comparative experiments. For the Preisach model, the fuzzy model and the proposed model, the testing voltages at different frequencies are input to the trained models to generate the output displacements, which are compared with the actual measurements to verify the model generalization ability.

The testing results are presented in Figure C4(a), where the blue solid curve corresponds to the actual measurements, the black dot dash curve denotes the output displacements of the Preisach model built by the data with the frequency of 5 Hz, the green dotted curve stands for the output displacements of the fuzzy model, and the red dashed curve represents the output displacements of the proposed model. For the Preisach model, since it does not contain frequency information of the voltage, the model is

1) The training data and the testing data are available in the attachment.

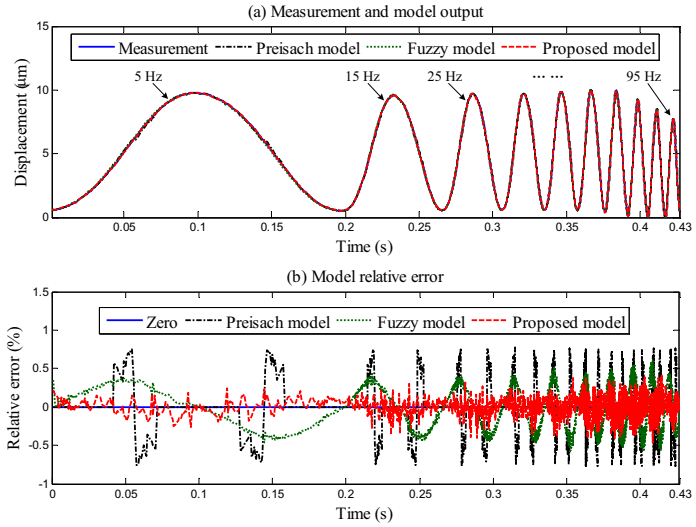

Figure C2 The training results of the proposed model and the reference models.

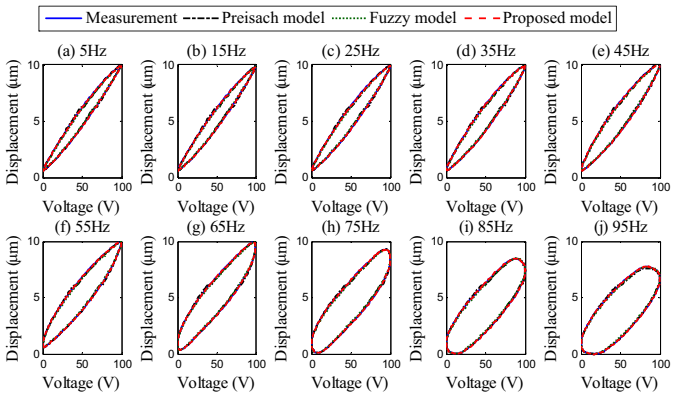

Figure C3 Hysteresis loops of the proposed model and the reference models built by model training.

Table C1 Area of hysteresis loops corresponding to the proposed model and the reference models built by model training.

Frequency (Hz)		5	15	25	35	45
Area (V·μm)	Measurement	114.53	124.75	145.20	165.73	191.04
	Preisach model	113.60	123.77	144.22	164.78	189.82
	Fuzzy model	113.96	124.07	144.71	165.08	189.88
	Proposed model	114.58	124.44	145.07	165.66	191.06
Frequency (Hz)		55	65	75	85	95
Area (V·μm)	Measurement	221.54	279.19	327.55	356.80	373.99
	Preisach model	219.86	277.38	325.40	353.89	369.02
	Fuzzy model	219.97	278.13	325.82	354.14	369.72
	Proposed model	221.61	278.99	327.31	356.64	374.05

not suitable for the untrained data, thus causing obvious deviations between the output displacements and measurements. For the frequency-dependent fuzzy model, its output displacements approach to the actual ones despite a certain degree of errors. Compared with the two reference models, the output displacements of the proposed model are much closer to the measurements due to the frequency-dependence and the ability to store memory.

Besides, the model errors of the testing results are plotted in Figure C4(b). For the Preisach model, the maximum absolute value of the relative error (74.82%) and the mean absolute value of the relative error (23.50%) are pronounced due to its weak generalization ability. For the fuzzy model, the maximum absolute value and the mean absolute value of the relative error are 8.53% and 2.63%. For the proposed model, the maximum absolute value of relative error is 5.81%, and the mean absolute value of the relative error is 0.66%, which are both less than those of the two reference models. Therefore, the testing results verify that the proposed model can still be applied to the new data different from the training set, thus exhibiting the strong generalization ability of the proposed method.

Similarly, the voltage-strain representations of the testing results are provided in Figure C5, and the hysteresis area for different loops is presented in Table C2. It can be seen that, since the Preisach model is frequency-independent, the hysteresis loops are basically the same for different frequencies and differ from the actual measurements, thereby making it unsuitable for simulating untrained hysteresis data. For the fuzzy model, it can roughly simulate the hysteresis loops of the untrained data. Furthermore, the hysteresis loops generated by the proposed model are still close to the actual ones, which further demonstrates the strong applicability of the proposed model.

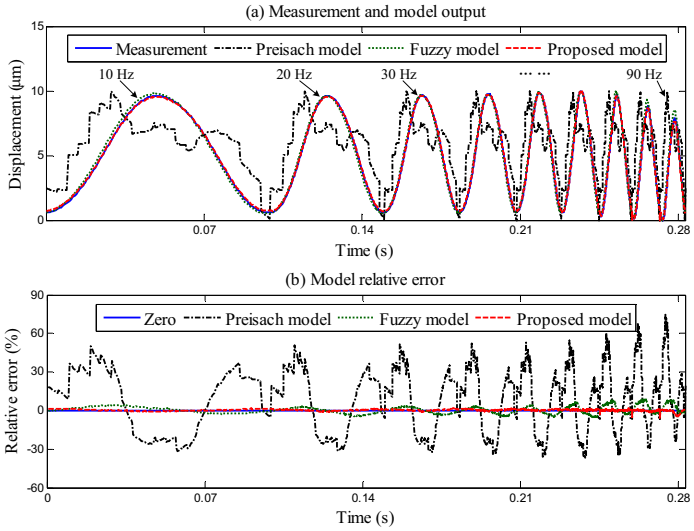

Figure C4 The testing results of the proposed model and the reference models.

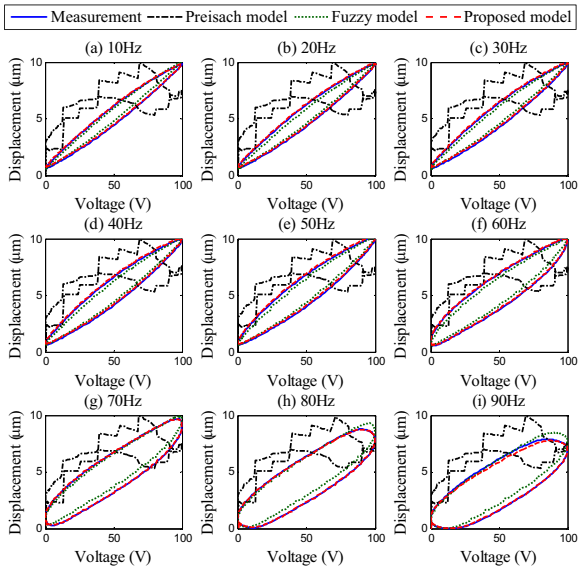

Figure C5 Hysteresis loops of the proposed model and the reference models built by model testing.

Table C2 Area of hysteresis loops corresponding to the proposed model and the reference models built by model testing.

Frequency (Hz)		10	20	30
Area (V·μm)	Measurement	115.68	133.57	153.08
	Preisach model	80.15	79.60	80.15
	Fuzzy model	105.23	121.31	137.18
	Proposed model	114.16	133.48	153.68
Frequency (Hz)		40	50	60
Area (V·μm)	Measurement	175.58	202.92	249.84
	Preisach model	79.60	80.15	79.60
	Fuzzy model	157.44	183.76	230.47
	Proposed model	173.47	203.11	249.12
Frequency (Hz)		70	80	90
Area (V·μm)	Measurement	297.96	340.06	360.19
	Preisach model	79.60	77.99	79.78
	Fuzzy model	281.91	322.78	331.84
	Proposed model	294.94	337.34	355.49

References

1 Li Z, Shan J, Gabbert U. Development of Reduced Preisach Model Using Discrete Empirical Interpolation Method. IEEE Transactions on Industrial Electronics, 2018, 65(10): 8072–8079.

2 Zhao X, Xie H, Pan H. Modeling Rate-Dependent Hysteresis in Piezoelectric Actuators Using T-S Fuzzy System Based on Expanded Input Space Method. Sensors & Actuators A Physical, 2018, 283: 123–127.