

Article

Classical Preisach Model Based on Polynomial Approximation and Applied to Micro-Piezoelectric Actuators

Van-Tsai Liu ^{1,*}  and Home-Young Wing ²¹ Department of Electrical Engineering, National Formosa University, Huwei 632, Taiwan² Department of Electrical and Control Engineering, National Chiao Tung University, Hsinchu 300, Taiwan; 10965109@gm.nfu.edu.tw

* Correspondence: vtliu@nfu.edu.tw

Abstract: In engineering applications, where we demand more and more precision, the modeling of systems with hysteretic nonlinearity has received considerable attention. The classical Preisach model (CPM) is currently the most popular for characterizing systems with hysteresis, and this model can represent the hysteresis with an infinite but countable first-order inversion curve (FORC). The table method is a method used to realize CPM in practice. The data in the table corresponds to a limited number of FORC samples. There are two problems with this approach: First, in order to reflect the timing effects of elements with hysteresis, it needs to consume a lot of memory space to obtain accurate data table. Second, it is difficult to come up with an efficient way to modify the data table to reflect the timing effects of elements with hysteresis. To overcome these shortcomings, this paper proposes to use a set of polynomials instead of the table method to implement the CPM. The proposed method only needs to store a small number of polynomial coefficients, and thus it reduces the required memory usage. In addition, to obtain polynomial coefficients, we can use least squares approximation or adaptive identification algorithms, which can track hysteresis model parameters. We developed an adaptive algorithm for the identification of polynomial coefficients of micro-piezoelectric actuators by applying the least mean method, which not only reduces the required memory size compared to the table method implementation, but also achieves a significantly improved model accuracy, and the proposed method was successfully verified for displacement prediction and tracking control of micro-piezoelectric actuators.

Keywords: hysteresis; classical Preisach model (CPM); polynomial approximation (PA); micro-piezoelectric actuator



Citation: Liu, V.-T.; Wing, H.-Y. Classical Preisach Model Based on Polynomial Approximation and Applied to Micro-Piezoelectric Actuators. *Symmetry* **2022**, *14*, 1008. <https://doi.org/10.3390/sym14051008>

Academic Editor: Sergei D. Odintsov

Received: 17 April 2022

Accepted: 12 May 2022

Published: 16 May 2022

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



Copyright: © 2022 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 (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the development of computer and numerical techniques, the use of hysteresis models has become more and more popular in finite element (FE) method (FEM)-based computer-aided design (CAD) simulations for calculating magnetization properties and iron losses [1]. Hysteresis appears in many engineering devices or smart materials, such as micro-piezoelectric actuators (PZA), piezoelectric ceramics, shape memory alloys (SMA), and ferromagnetic components [2–6], which is a nonlinear phenomenon. Systems with hysteresis are often difficult to describe accurately, and without feedback control, may lead to unstable behavior of the system. When a device exhibits nonlinearity, the system often exhibits inaccuracies or oscillations, even instability caused by the non-differentiable hysteresis and non-memoryless properties of magnetic hysteresis [7–9]. Therefore, accurate hysteresis models are crucial for developing suitable control algorithms or applications using these systems [10]. Several models have been developed to characterize systems with hysteresis; commonly used models include the Bouc–Wen, Preisach-like Krasnosel'skii–Pokrovskii (KP), Prandtl–Ishlinskii model (PIM), and the classical Preisach model (CPM) [11]. CPM has long been known as the most familiar model for characterizing

hysteretic behavior. It uses double integrals with relay operators and weighting parameters to describe the system input/output relation, which can be further represented by an infinite but countable first-order inversion curve (FORC), which can be experimentally measured in many applications [12]. The traditional method to implement CPM for systems with hysteresis approximate by sampling a finite number of FORCs and storing the sample data in a table [13]. The model output is then evaluated from the table data and the given input by simple linear interpolation. However, this approach has two drawbacks: One is that when the number and sampling resolution of FORC should be increased, it requires a large amount of memory space to accurately predict the hysteresis behavior. The other is that it is difficult to find an efficient way to adjust the data table to reflect changes in parameters with aging or time. As a result, execution accuracy may degrade over time and even require rebuilding of the table.

To overcome these drawbacks, we recommend implementing CPM using a set of polynomials rather than data tables. It only needs to store a small number of polynomial coefficients to reduce the required storage space. Specifically, polynomial coefficients can be obtained using a least squares approximation or an adaptive identification algorithm [14], allowing tracking of hysteresis model parameters. The proposed method has been applied to computer simulations of CPM output predictions. We apply the least mean squares (LMS) adaptation algorithm to develop an adaptive algorithm for identifying the polynomial coefficients of FORC. Compared to the implementation of the table method, the results show that it has higher model accuracy than the table method and requires significantly less memory. Furthermore, the proposed method combined with an adaptive algorithm can achieve dynamic modification of CPM parameters instead of rebuilding the table. According to [15], the proposed method has been applied to simulate the displacement of micro-piezo actuators. The modeled RMS error changes from 1.0098 μm for the table method to 0.4539 μm .

The remainder of this paper is divided to five sections including conclusions. Section 2 reviews the hysteresis and the Preisach model. Section 3 introduces the basic definition and the numerical implementation on the CPM. Section 4 present the computer simulations to verify capabilities of different realizations of the CPM. Section 5 demonstrates the applications of the micro-piezoelectric actuators. The final section presents the conclusions.

2. Realization Numerical of CPM

The classical Preisach model (CPM) [16,17] represents the input and output relations of a system with hysteresis by a double integral formula as below:

$$f(t) = \iint_{\alpha \geq \beta} \mu(\alpha, \beta) \gamma_{\alpha\beta}[u(\cdot)](t) d\alpha d\beta \quad (1)$$

where $u(t)$ denotes the CPM input, $f(t)$ is the CPM output, $\mu(\alpha, \beta)$ is the Preisach density function, and $\gamma_{\alpha\beta}[u(\cdot)](t)$ is an operator with two distinct outputs. The idea of the CPM is to regard a hysteresis transducer as the superposition of infinitely weighted hysteresis operators as shown in Figure 1.

The hysteresis operator is the most important core in the Preisach model. The values α and β of the hysteresis operator correspond to “up” and “down” switching values of input, respectively. In this paper, we assume that $\alpha \geq \beta$. The operator is characterized by its switching values α and β , which are represented by a rectangular loop on its input and output phase diagrams, as shown in Figure 2. As shown in the figure, the operator output of $\gamma_{\alpha\beta}[u(\cdot)](t)$ remains -1 until the input is higher than the value of α , and the output remains $+1$ until the input is lower than the value of β . A practical expectation for practical applications is the limit of the integration, which requires us to set the saturation value of the input, that is, $u_{set} \geq \alpha \geq \beta \geq -u_{set}$.

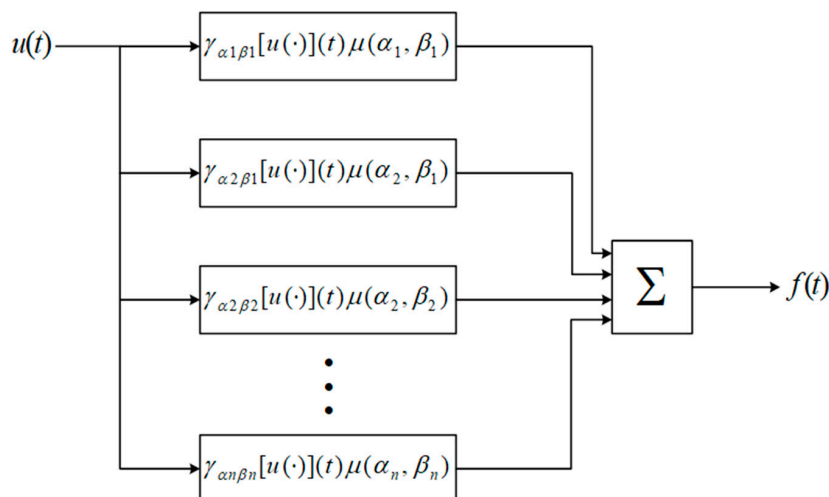


Figure 1. Architecture of the Preisach model.

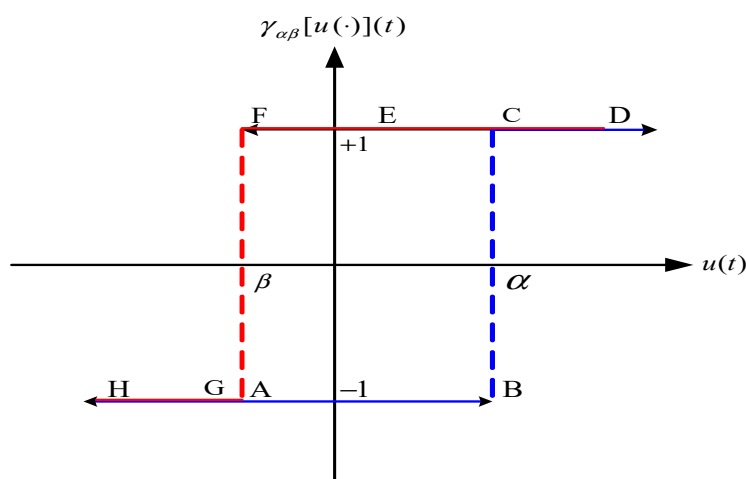


Figure 2. The hysteresis operator: the output follows trajectory ABCD when the input is increasing and follows EFGH when the input is decreasing.

A progressive evaluation of the CPM is demonstrated by the following example, depicted in Figure 3. In the example, we calculate a first order reversal curve (FORC), a curve is formed after the first reversal of the input, and a second order reversal curve. Any higher order reversal curve can be constructed by using the analogical method.

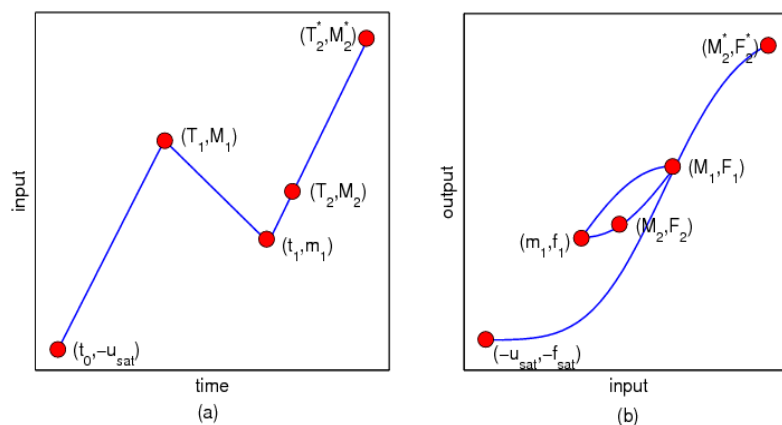


Figure 3. A progressive evaluation of the CPM: (a) the input diagram; (b) the input–output phase diagram.

Assume the system is first at negative saturation; that is, the CPM input is $-u_{sat}$ and all hysteresis operator outputs are -1 . Then, the initial CPM output is

$$f(t) = \iint_{a \geq \beta} \mu(\alpha, \beta) \gamma_{\alpha\beta}[u(\cdot)](t) d\alpha d\beta \quad (2)$$

As the CPM input increases from $-u_{max}$ to M_1 , the hysteresis operator output will be switched to the -1 if the switching value α is less than the current input value M_1 . Then, the CPM output is

$$f(T_1) = -f_{sat} + 2 \iint_{M_1 \geq a \geq \beta \geq -u_{sat}} \mu(\alpha, \beta) d\alpha d\beta \quad (3)$$

When the CPM input changes direction and decreases from M_1 to a value m_1 , the hysteresis operator output will be switched to the -1 state for the switching value β greater than m_1 . Then, the FORC is

$$f(t_1) = -f(T_1) - 2 \iint_{M_1 \geq a \geq \beta \geq -m_1} \mu(\alpha, \beta) d\alpha d\beta \quad (4)$$

Evaluation of $f(T_2)$ is the same with $f(T_1)$, that is,

$$f(T_2) = -f(t_1) - 2 \iint_{M_2 \geq a \geq \beta \geq -m_1} \mu(\alpha, \beta) d\alpha d\beta \quad (5)$$

The wiping-out property appears as the input $u(t)$ reaching M_1 and then reaches M_2^* . Hysteresis operators with a switching value α less than M_2^* but greater than M_1 stay at the $+1$ state after T_2 , so they have no influence on the current output. Additionally, hysteresis operators with α greater than M_1 but less than M_2^* are originally at the -1 state and would be transitioned to $+1$. Summarizing all of the above,

$$\begin{aligned} f(T_2^*) &= f(T_1) + 2 \iint_{M_2^* \geq a \geq \beta \geq -u_{sat}} \mu(\alpha, \beta) d\alpha d\beta \\ &= -f_{sat} + 2 \iint_{M_2^* \geq a \geq \beta \geq -u_{sat}} \mu(\alpha, \beta) d\alpha d\beta \end{aligned} \quad (6)$$

It is clear that (M_1, F_1) and (m_1, f_1) , which were wiped out, are not necessary to determine $f(T_2^*)$ in (5).

In the CPM, hysteresis operator is a concern associated with the impact of the past inputs for such scheme. The wiping-out property, however, clarifies that to determine the current CPM the output does not have to memorize all of the history of CPM but only parts of the extremum of the past inputs and outputs [18]. The CPM can be rewritten in such a way to dichotomize a CPM output into two terms: the shifted term, depending on the past outputs, and the integrated term, depending on the past inputs and the current input. We can define alternating series, which are kept to determine the limits of integration and the storage part of a CPM output, by generalizing the above-mentioned example as below. Consider a particular input $u(t)$ for a CPM in the time interval $[t_0, t]$ and $u(t_0) = -u_{sat}$. We use the notations T_k for the time the global maximum M_k of $u(t)$ is reached and t_k for the time the global minimum m_k of $u(t)$ is reached, where M_k and m_k satisfy that

$$M_k = \max_{[t_{k-1}, t]} u(t) = u(T_k), m_k = \min_{[T_k, t]} u(t) \quad (7)$$

The alternating series of the input in $[t_0, t]$ is defined as a tuple $s_i = \{-u_{sat}, M_1, m_1, \dots, u(t)\}$, and the alternating series of the output in $[t_0, t]$ is $s_o = \{-f_{sat}, F_1, f_1, \dots, f(t)\}$, where $F_k = f(T_k)$ and $f_k = f(t_k)$. Furthermore, for a given alternating series s_i and s_o , except $f(t)$ at some point, the CPM output is obtained by

$$f(t) = f_n + 2 \iint_{u(t) \geq a \geq \beta \geq m_n} \mu(\alpha, \beta) d\alpha d\beta \quad (8)$$

If $u(t)$ is increasing, and

$$f(t) = F_n - 2 \iint_{M_n \geq \alpha \geq \beta \geq -u(t)} \mu(\alpha, \beta) d\alpha d\beta \tag{9}$$

If $u(t)$ is decreasing. In the ensuing discussion, the alternating series except $f(t)$ are assumed to be recorded.

So far, we developed forthright mathematical Formulas (8) and (9) to simplify the CPM with the known Preisach density function and given alternating series. However, the identification of $\mu(\alpha, \beta)$ requires the differentiation of experimental data, which may amplify errors in the experimental data. There are some algorithms [19,20] published to remove sensitivity of identification of $\mu(\alpha, \beta)$, but the approach using $\mu(\alpha, \beta)$ to evaluate CPM outputs is still unappealing. Although we could find an accurate $\mu(\alpha, \beta)$ of CPM, the double integration would be very time consuming and mathematically intractable. The congruency property of CPM supplies another way of representing a CPM through use of the FORC to replace integration with experimental data [16]. We introduce the property and how to use FORC to numerically implement a CPM by examining (8) and (9). It is expected that the integrated terms in (8) or (9) for a particular CPM input will be congruent with other CPM input if their limits of integration are the same. Ground on this fact, we can store integrated terms with variational limits of integration beforehand. The FORC measuring experiment is used for this notion. In order to connect FORC and the algebraic calculation of the CPM output, a function $f_a(m, M)$ is defined as the output of a CPM with $s_i = \{u_{sat}, m, M\}$ and a function $f_d(M, m)$ is the output of a CPM with $s_i = \{-u_{sat}, M, m\}$. Hence, $f_a(m, M)$ denotes an ascending FORC and $f_d(M, m)$ denotes a descending FORC. Thus, (8) is equivalent to

$$f(t) = f_n + f_a(m_n, u(t)) - f_a(m_n, m_n) \tag{10}$$

Additionally, (9) is equivalent to

$$f(t) = F_n + f_d(M_n, u(t)) - f_d(M_n, M_n) \tag{11}$$

From the above discussion, the numerical implementation of the CPM can be obtained by combining the alternating series and the FORC, which can be used for evaluation of the CPM output; this approach avoids the trouble of differentiation and integration shown in Figure 4.

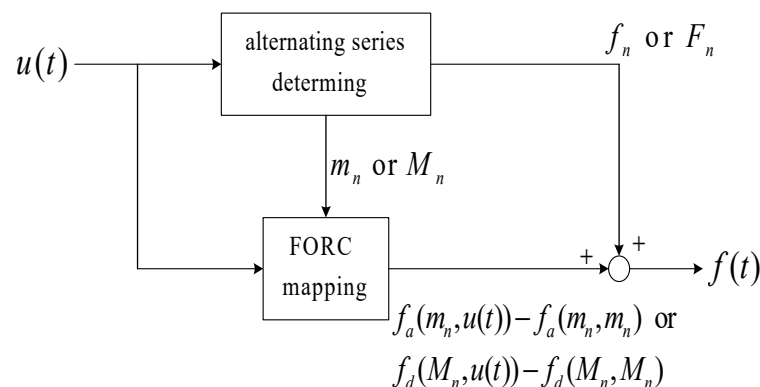


Figure 4. Block diagram of the numerical implementation of the CPM.

3. Realizations of the CPM

In the previous section, we introduced the basic definition and the numerical implementation on the CPM. We also developed $f_a(m, M)$ and $f_d(M, m)$ as an algebraic representation of a CPM. Because the $f_a(m, M)$ and $f_d(M, m)$ can be explicitly obtained by experiments of the FORC-measuring experiments, they play the central roles in the realization of a CPM. In practice, however, it is impossible to obtain all values of FORC

which requires infinite and uncountable experiments. One realizable method to approximate values of FORC is to find finite data of $f_a(m, M)$ and $f_d(M, m)$, store them as tables, and we can obtain values of $f_a(m, M)$ or $f_d(M, m)$ by the interpolation using its neighbors stored in tables. In this section, we take an introductory look at different realizations of a CPM using finite-FORC, including the table method [21,22], polynomial approximation, and adaptive polynomial approximation.

Linear interpolation is a very intuitive way to estimate values between known values and we adopt this method to approximate $f_a(m, M)$ or $f_d(M, m)$. If a point (M, m) belongs to a rectangular cell formed by (M_1, m_1) , (M_1, m_2) , (M_2, m_1) , (M_2, m_2) , and we have $f_d(M_1, m_1) = f_{M_1m_1}$, $f_d(M_1, m_2) = f_{M_1m_2}$, $f_d(M_2, m_1) = f_{M_2m_1}$, $f_d(M_2, m_2) = f_{M_2m_2}$. Then, $f_d(M, m)$ is approximated by

$$\begin{aligned} f_L &= \frac{(M_2 - M)f_{M_1m_2} + (M - M_1)f_{M_2m_2}}{M_2 - M_1}, \\ f_R &= \frac{(M_2 - M)f_{M_1m_1} + (M - M_1)f_{M_2m_1}}{M_2 - M_1}, \\ f_d(M, m) &= \frac{(m - m_1)f_L + (m_2 - m)f_R}{m_2 - m_1}. \end{aligned} \quad (12)$$

If a point (M, m) belongs to a triangular cell formed by (M_1, m_1) , (M_2, m_1) , (M_2, m_2) , then $f_d(M, m)$ is approximated by

$$\begin{aligned} f_R &= f_{M_2m_2}, f_L = \frac{(M_2 - M)f_{M_1m_2} + (M - M_1)f_{M_2m_2}}{M_2 - M_1}, \\ f_d(M, m) &= \frac{(m - m_1)f_L + (m_2 - m)f_R}{m_2 - m_1}. \end{aligned} \quad (13)$$

The Formulas (12) and (13) also exist for $f_a(m, M)$.

It is convenient to investigate the measured data using an analyzable function called curve fitting and the polynomial function is commonly used. Our goal is to fit the samples of a finite number of FORC using polynomial functions. For instance, we can use a quadratic polynomial function of single variable M to fit $f_a(m, M)$ with a particular m . That is,

$$f_a(m, M) = a_0(m) + a_1(m)M + a_1(m)M^2 \quad (14)$$

Additionally, $f_d(M, m)$ also can be approximated as

$$f_d(M, m) = d_0(M) + d_1(M)m + d_2(M)m^2 \quad (15)$$

where $a_i(m)$ and $d_i(m)$ are notations to indicate the polynomial coefficients corresponding with m and M , respectively. The coefficients of the curve fitting method vary with m or M , so to know all of them is also impossible. Hence, we also use the interpolation to approximate coefficients that are not derived from experiments. For $m_k < m < m_{k+1}$,

$$a_i(m) = \frac{m_{k+1}a_i(m_k) + m_k a_i(m_{k+1})}{m_{k+1} - m_k}, \text{ where } i = 0, 1, 2. \quad (16)$$

For $M_k < M < M_{k+1}$,

$$d_i(M) = \frac{M_{k+1}d_i(M_k) + M_k d_i(M_{k+1})}{M_{k+1} - M_k} \quad (17)$$

According to (10) and (11), a CPM output can be determined by a sifted term and a combination of FORC. Now, we substitute for FORC by using (14) and (15). If the input $u(t)$ is ascending,

$$f(t) = f_n + a_2(m_n)(u^2(t) - m_n^2) + a_1(m_n)(u(t) - m_n) \quad (18)$$

Define $a_i(p\Delta) = a_{si}[p]$, $i = 0, 1, 2$; for $m_n = p\Delta + r = u$, $0 \leq r < \Delta$, then

$$a_i(m_n) = \left(1 - \frac{r}{\Delta}\right) a_{si}[p] + \frac{r}{\Delta} a_{si}[p+1] \quad (19)$$

Thus, (18) becomes

$$f(t) = f_n + \theta_1^T x_1 \quad (20)$$

where

$$\theta_1 = \begin{bmatrix} a_{s2}[p] \\ a_{s2}[p+1] \\ a_{s1}[p] \\ a_{s1}[p+1] \end{bmatrix}, x_1 = (u - m_n) \begin{bmatrix} \left(1 - \frac{r}{\Delta}\right)(u + m_n) \\ \frac{r}{\Delta}(u + m_n) \\ \left(1 - \frac{r}{\Delta}\right) \\ \frac{r}{\Delta} \end{bmatrix} \quad (21)$$

Additionally, u indicates $u(t)$. Similarly, if the input $u(t)$ is descending,

$$f(t) = F_n + \theta_2^T x_2 \quad (22)$$

where

$$\theta_2 = \begin{bmatrix} d_{s2}[p] \\ d_{s2}[p+1] \\ d_{s1}[p] \\ d_{s1}[p+1] \end{bmatrix}, x_2 = (u - M_n) \begin{bmatrix} \left(1 - \frac{r}{\Delta}\right)(u + M_n) \\ \frac{r}{\Delta}(u + M_n) \\ \left(1 - \frac{r}{\Delta}\right) \\ \frac{r}{\Delta} \end{bmatrix} \quad (23)$$

Additionally, $M_n = p\Delta + r$, $0 \leq r < \Delta$. The polynomial coefficients represent the parameters of the CPM and are suitable for model identification because they are adjustable.

Based on the polynomial approximation, we obtain a set of coefficients that can be used to describe FORC. However, these coefficients may not be accurate. For a system model with inaccurate parameters, we can use adaptive algorithms such as LMS or RLS [18] to update the parameters. Compared with the adaptive model output and plant output, this error is used to update adaptive system parameters. Assume the adaptive model output $f_m = (\theta, x) = \theta^T x$ is a function of parameters θ and the input x , and the plant output $f_p(x)$ is a function of the input x . Define the error as

$$e = f_p(x) - f_m(\theta, x) \quad (24)$$

The steepest descent algorithm is:

$$\theta^{(k+1)} = \theta^{(k)} + 2\eta ex \quad (25)$$

where η is a step size. Using (20), (22), (25), we obtain following updated formulations:

If the input $u(t)$ is ascending,

$$\theta_1^{(k+1)} = \theta_1^{(k)} + 2\eta ex_1 = \begin{bmatrix} a_{s2}^{(k)}[p] \\ a_{s2}^{(k)}[p+1] \\ a_{s1}^{(k)}[p] \\ a_{s1}^{(k)}[p+1] \end{bmatrix} + 2\eta e(u - m_n) \begin{bmatrix} \left(1 - \frac{r}{\Delta}\right)(u + m_n) \\ \frac{r}{\Delta}(u + m_n) \\ \left(1 - \frac{r}{\Delta}\right) \\ \frac{r}{\Delta} \end{bmatrix} \quad (26)$$

If the input $u(t)$ is descending,

$$\theta_2^{(k+1)} = \theta_2^{(k)} + 2\eta ex_2 = \begin{bmatrix} d_{s2}^{(k)}[p] \\ d_{s2}^{(k)}[p+1] \\ d_{s1}^{(k)}[p] \\ d_{s1}^{(k)}[p+1] \end{bmatrix} + 2\eta e(u - M_n) \begin{bmatrix} \left(1 - \frac{r}{\Delta}\right)(u + M_n) \\ \frac{r}{\Delta}(u + M_n) \\ \left(1 - \frac{r}{\Delta}\right) \\ \frac{r}{\Delta} \end{bmatrix} \quad (27)$$

Then, we can obtain CPM outputs by applying polynomial coefficients adapted by (26) or (27) to (20) or (22).

4. CPM Simulation and Discussion

In this section, we demonstrate the computer simulations to verify capabilities of different realizations of the CPM proposed in Section 3. Consider a special CPM with Preisach density function,

$$\mu(\alpha, \beta) = k \exp\left(-\frac{1}{2} \left\{ \left(\frac{\alpha - \alpha_c}{\sigma_1}\right)^2 + \left(\frac{\beta - \beta_c}{\sigma_2}\right)^2 \right\}\right) \quad (28)$$

We chose $k = 50$, $\alpha_c = \beta_c = 0$, $\sigma_1 = 1.5$, $\sigma_2 = 1.8$, and $u_{sat} = 3$ in our simulations. The exact CPM outputs are calculated by numerical integral of Matlab. The FORC are measured and sampled in the tables first; these sampled data are used to obtain polynomial coefficients. The CPM outputs are predicted via different realizations and the root mean square error (RMS) is adopted as the judgment of how close the prediction is to the CPM. We can also observe the tracking ability of the adaptive polynomial approximation by disturbing $\mu(\alpha, \beta)$ slightly.

To implement the table method, the samples of the ascending FORC and descending FORC are measured first. By dividing $-3 \sim +3$ into 6 segments, the CPM input is decreased from $+3$ to some divided point and then increased to $+3$. After the input begins decreasing from some divided point to $+3$, we sample the CPM outputs every unit change in the CPM input and store these sampling data in Table 1 which presents the samples of ascending FORC. The first column denotes the value of m of $f_a(m, M)$, and the first row denotes the value of M . The samples of descending FORC are stored in Table 2 using a similar procedure and both ascending and descending FORC are plotted in Figure 5.

Table 1. Sampling data of ascending FORC.

m/M	−3	−2	−1	0	1	2	3
−3	−21.58	−13.70	−1.09	11.60	19.02	21.31	21.58
−2		−13.70	−12.40	−5.90	5.75	16.29	21.58
−1			−1.10	1.54	9.34	17.35	21.58
0				11.60	14.18	18.08	21.58
1					19.02	20.24	21.58
2						21.31	21.58
3							21.58

Table 2. Sampling data of descending FORC.

M/m	−3	−2	−1	0	1	2	3
−3	−21.58						
−2	−21.58	−21.28					
−1	−21.58	−19.89	−18.59				
0	−21.58	−17.78	−12.62	−9.98			
1	−21.58	−15.67	−6.64	1.21	3.79		
2	−21.58	−14.29	−2.75	8.50	14.48	15.70	
3	−21.58	−13.70	−1.09	11.60	19.02	21.31	21.58

We apply the least-square approximation to sampling data of Tables 1 and 2 to identify the polynomial coefficients directly in this simulation. Because of the establishment of saturated boundaries, FORC are much more gradual and cannot be fit by polynomials adequately. Hence, $f_a(m, 3)$ and $f_d(M, -3)$ are rejected in the procedure of the least-square approximation and the table method is used if they are needed. The polynomial coefficients are stored in Tables 3 and 4. Based on (20) and (22), the polynomial coefficients a_0 and d_0 are not needed for the polynomial approximation. The polynomial coefficients of $f_a(m, M)$ and $f_d(M, m)$ are plotted in Figure 6.

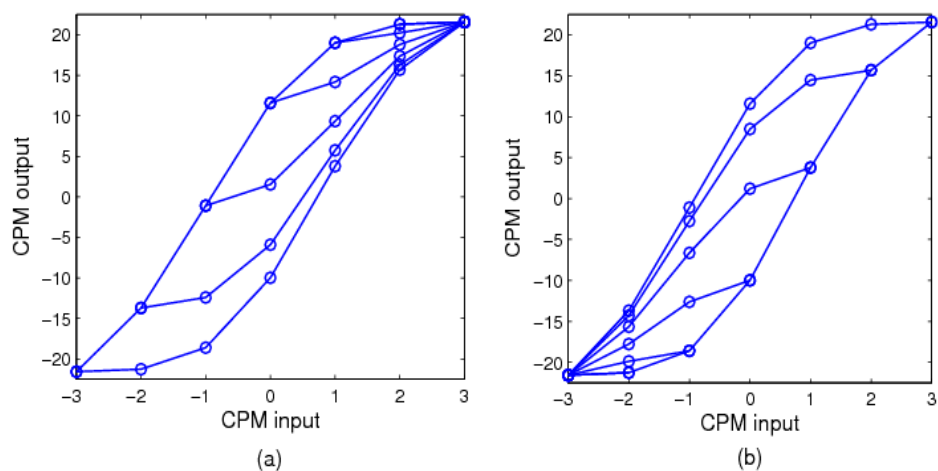


Figure 5. FORC of the CPM: (a) ascending FORC; (b) descending FORC.

Table 3. Polynomial coefficients of $f_a(m, M)$.

m	a_2	a_1	a_0
-3	1.83	9.55	-9.67
-2	1.69	7.81	-5.37
-1	1.34	4.97	2.28
0	1.02	1.56	11.60
1	0	-1.22	21.46
2	0	0	21.31

Table 4. Polynomial coefficients of $f_d(M, m)$.

M	d_2	d_1	d_0
-2	0	0	-21.28
-1	0	1.30	-17.30
0	-1.26	1.38	-9.98
1	-1.61	5.01	0.59
2	-1.85	7.72	8.03
3	-1.84	9.02	11.12

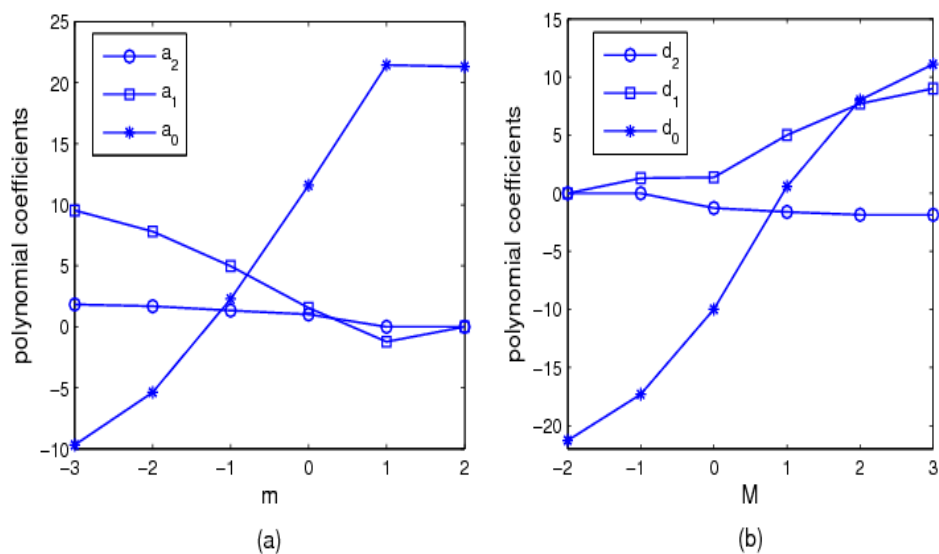


Figure 6. The coefficients of the PA: (a) of $f_a(m, M)$; (b) of $f_d(M, m)$.

In our simulations, we use a sinusoidal CPM input that with a frequency of 1 Hz has a sample time of 0.01 s, and a varying amplitude. The waveform of CPM input and the input–output phase diagram are shown in Figure 7. The simulation results are shown in Figures 8–11. The CPM output and CPM output realized via the table method are plotted together in Figure 8a. The difference between these two outputs, defined as the prediction error, are shown in Figure 8b. The RMS of the prediction error is 0.5256. The result of the simulation of the polynomial approximation is shown in Figure 9, and its RMS is 1.0717. We use a step size $\eta = 0.005$ and Tables 3 and 4 as initial parameters of the adaptive polynomial approximation and the simulation result is shown in Figure 10. The RMS of the adaptive polynomial approximation is 0.5909. We also plotted these three prediction errors together in Figure 11 for comparison. We can find that the performance of the adaptive polynomial approximation and the table method are in the same range roughly and are better than the polynomial approximation. This is because polynomial approximation uses a single function to characterize a FORC but the table method uses a number of linear functions. The adaptive polynomial approximation, however, adjusts the coefficients of the polynomial so it can be viewed using several functions to fit FORC equivalently and has a better effect than the polynomial approximation.

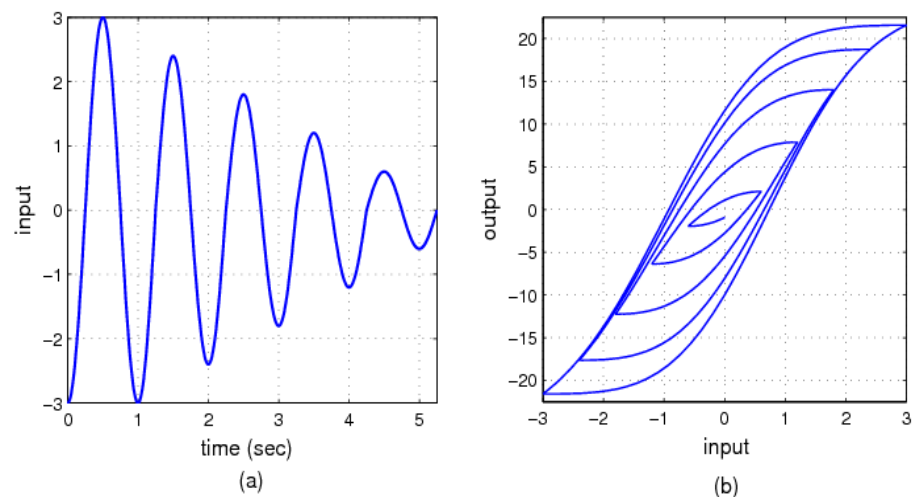


Figure 7. The CPM input and output of simulations: (a) the waveform of the CPM input; (b) the input–output phase diagram.

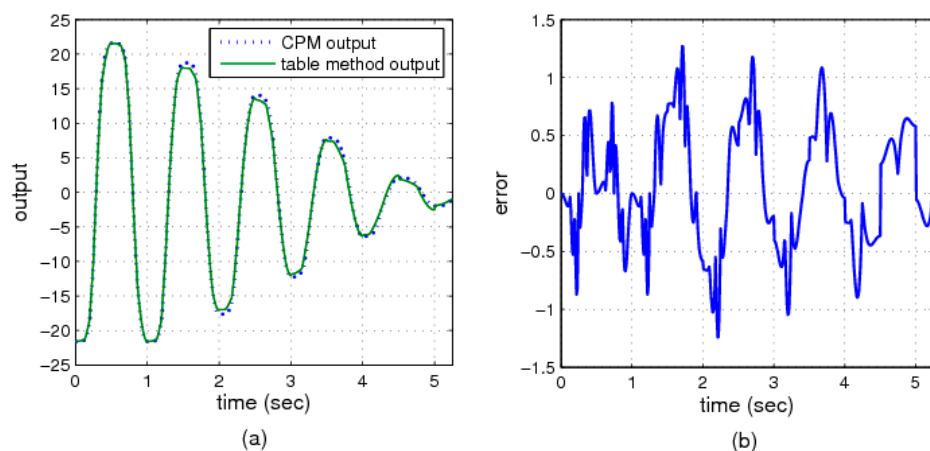


Figure 8. Prediction of the CPM output using the table method: (a) the CPM output compared with prediction via the table method; (b) the prediction error of the table method.

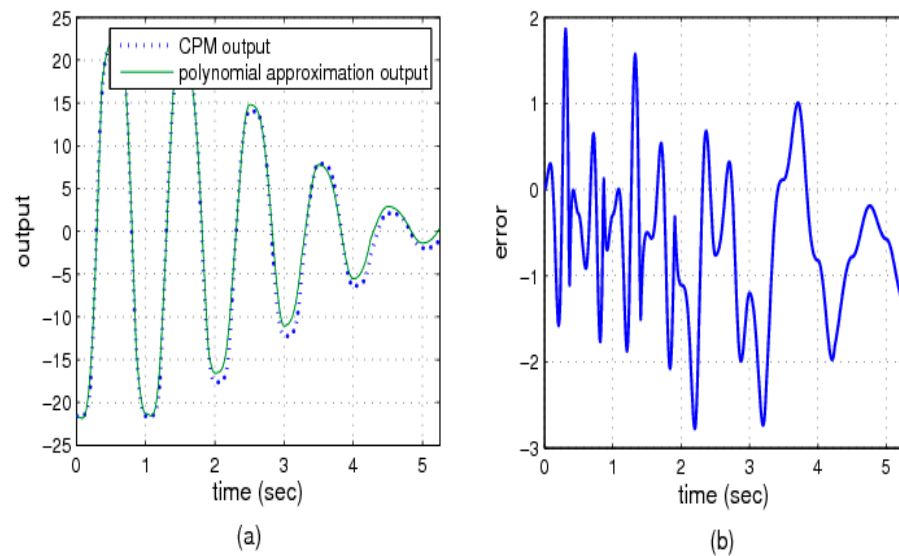


Figure 9. Prediction of the CPM output using PA: (a) the CPM output compared with prediction via PA; (b) the prediction error of PA.

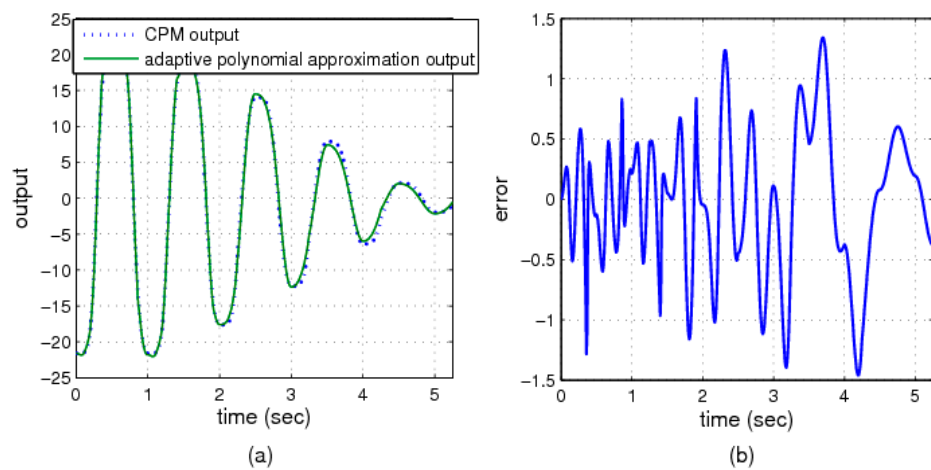


Figure 10. Prediction of the CPM output using the adaptive PA: (a) the CPM output compared with prediction; (b) the prediction error.

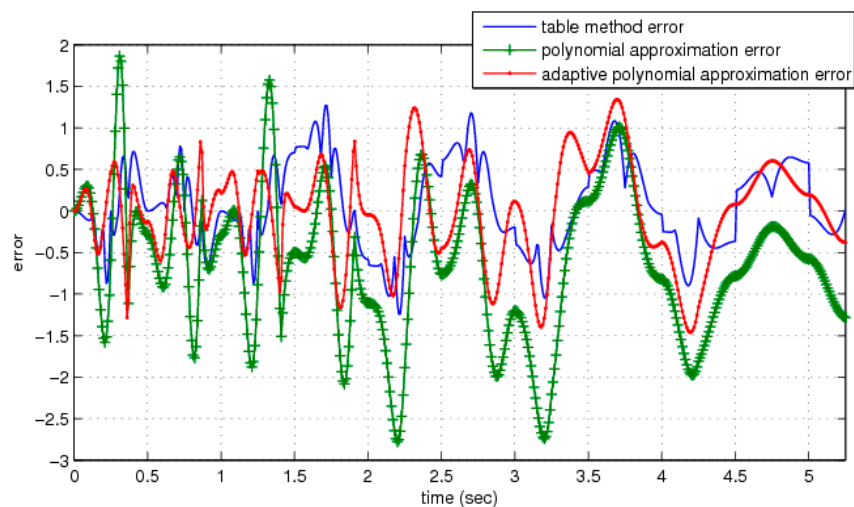


Figure 11. Prediction error of the CPM via different realizations.

We saw the performance of dissimilar realizations of the CPM with a fixed Preisach density function. That is, we assume that the quality of the CPM is abiding for all situations. This assumption is not coherent with systems in real applications. It is broad that the table method and the polynomial approximation are static models and hard to be modified in real time. An advantage of the adaptive polynomial approximation is that the parameters of such scheme are obtained on-line; this means that even if the CPM changed, the adaptive polynomial approximation will trace the CPM by adjusting its parameters. To witness this, we disturb $\mu(\alpha, \beta)$ such that σ_1 becomes from 1.5 to 1.3 and σ_2 becomes from 1.8 to 2.1. Tables 1 and 2 are still used to implement the table method; Tables 3 and 4 are still used to implement the polynomial approximation and are the initial parameters of the adaptive polynomial approximation. We apply the same CPM input of previous simulations and the new CPM output compared with the original CPM output is plotted in Figure 12. Then, we use the different realizations to predict the new CPM output and plot the prediction errors together in Figure 13. The RMS of each realization is 1.9707 for the table method, 1.0746 for the polynomial approximation, and 0.6349 for the adaptive polynomial approximation. It is clear that the adaptive polynomial approximation has the best performance in this simulation.

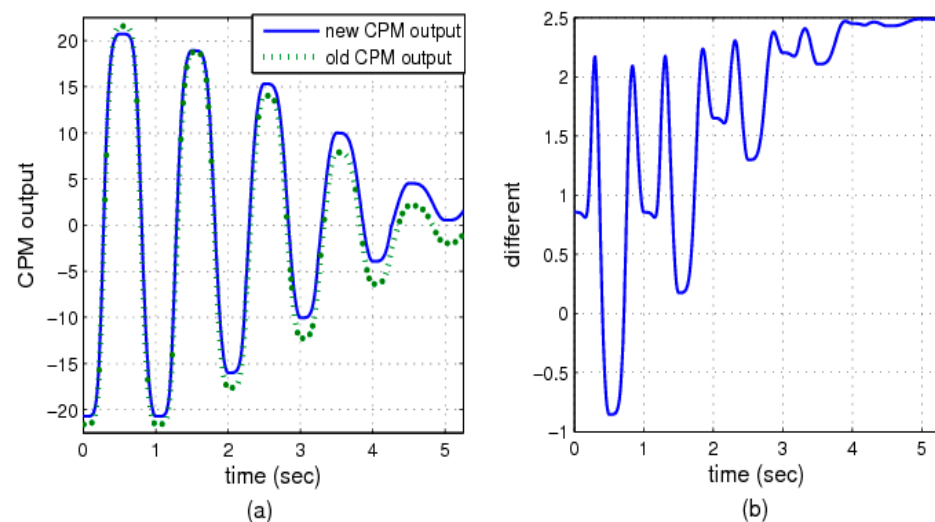


Figure 12. The CPM output of a changed Preisach density function: (a) the new CPM output compared with the original CPM output; (b) the difference between the new and original CPM output.

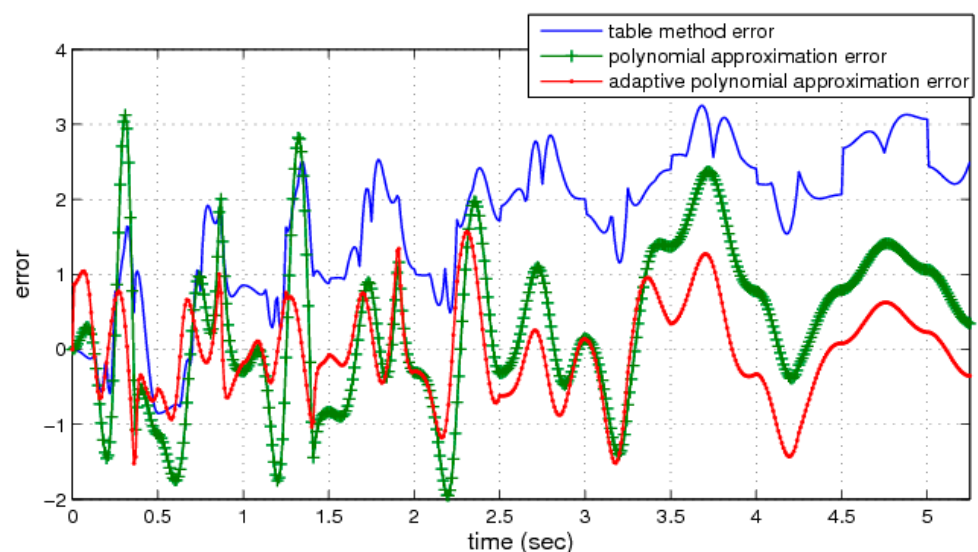


Figure 13. Prediction error of the CPM of a changed Preisach density function via different realizations.

5. Applications for Piezoelectric Actuators

We use different realizations of Preisach model to predict the displacement of a micro-piezoelectric actuator and the inverse Preisach model introduced in this thesis to control it. In these applications, the input of the Preisach model is actuator voltage and the output of the Preisach model is displacement.

To implement the Preisach model, several first order reversal curves are measured for the micro-piezoelectric actuator first. By dividing 0~5 V into 10 segments, the input voltage of micro-piezoelectric actuator is increased from 0 V to some divided point and then decreased to 0 V. After the input begins decreasing from some divided point to 0, we sample the displacements of the micro-piezoelectric actuator every 0.45 V change in input voltage and store these sampling data in Table 5. The first column denotes the value of M , and the first row denotes the value of m . In the predication experiment, we use a sinusoidal and input voltage with a frequency of 1 rad/s, an amplitude of 2.25 V, and a bias of 2.25 V; the input and output waveforms of the micro-piezoelectric actuator of the predication experiment are shown in Figure 14, and the phase transition is shown in Figure 15. In the tracking control experiment, we use a sinusoidal desired output with with a frequency of 1 rad/s, an amplitude of 15 μm , and a bias of 16 μm . The sampling time of the experiment is 0.01 s and the total time is 50 s.

Table 5. Sampling data of FORC.

M/m	0	0.45	0.90	1.35	1.80	2.25	2.70	3.15	3.60	4.05	4.50
0	0										
0.45	0.02	3.43									
0.90	0.09	4.07	7.73								
1.35	0.11	5.28	9.38	12.76							
1.80	0.14	5.36	10.13	14.11	18.15						
2.25	0.23	5.68	70.79	15.32	19.34	23.32					
2.70	0.28	5.94	11.31	16.22	20.58	24.48	28.03				
3.15	0.36	6.17	11.72	16.68	21.27	25.41	29.01	32.65			
3.60	0.45	6.35	12.09	17.29	21.96	26.27	30.13	33.51	36.49		
4.05	0.48	6.61	12.35	17.58	22.40	26.81	30.74	34.35	37.74	40.15	
4.50	0.67	6.87	12.87	18.07	22.98	27.33	31.38	34.93	38.13	40.96	43.18

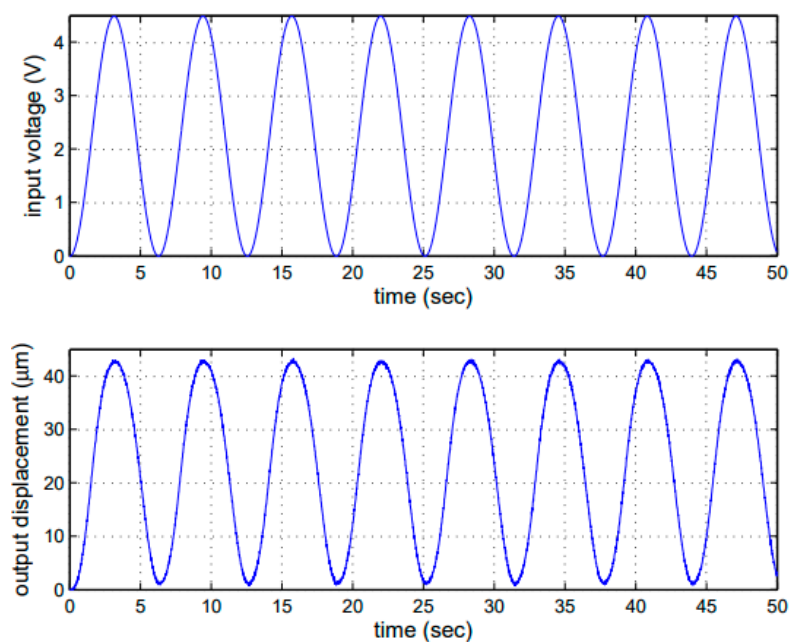


Figure 14. The input and output of the micro-piezoelectric actuator.

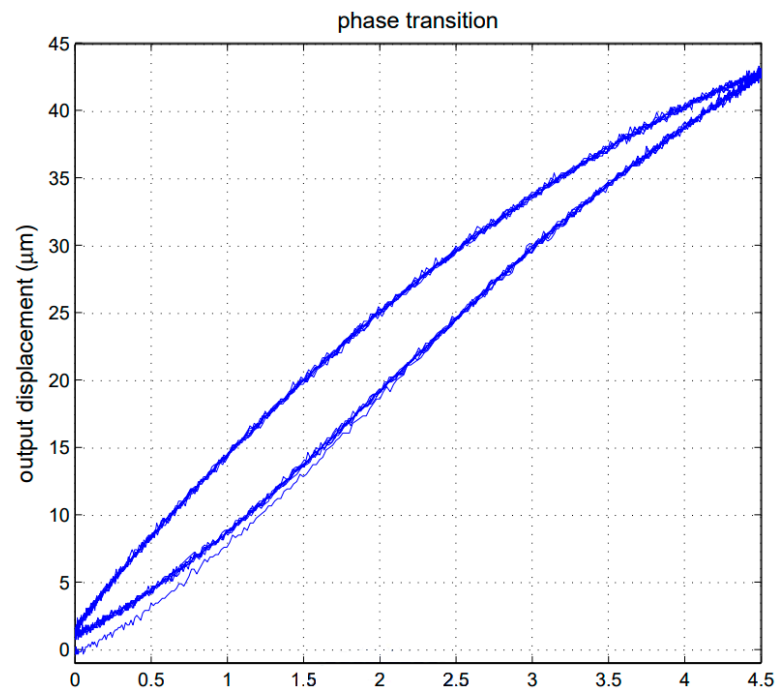


Figure 15. The phase transition of micro-piezoelectric actuator.

The displacement of the micro-piezoelectric actuator and the output of the Preisach model realized via the table method are plotted together in Figure 16a. The difference between these two outputs, defined as the model error, is shown in Figure 16b. The phase transition of model and of plant are plotted together in Figure 17. The RMS of the model error is $1.0098 \mu\text{m}$, and the required amount of memory to store the table is 66.

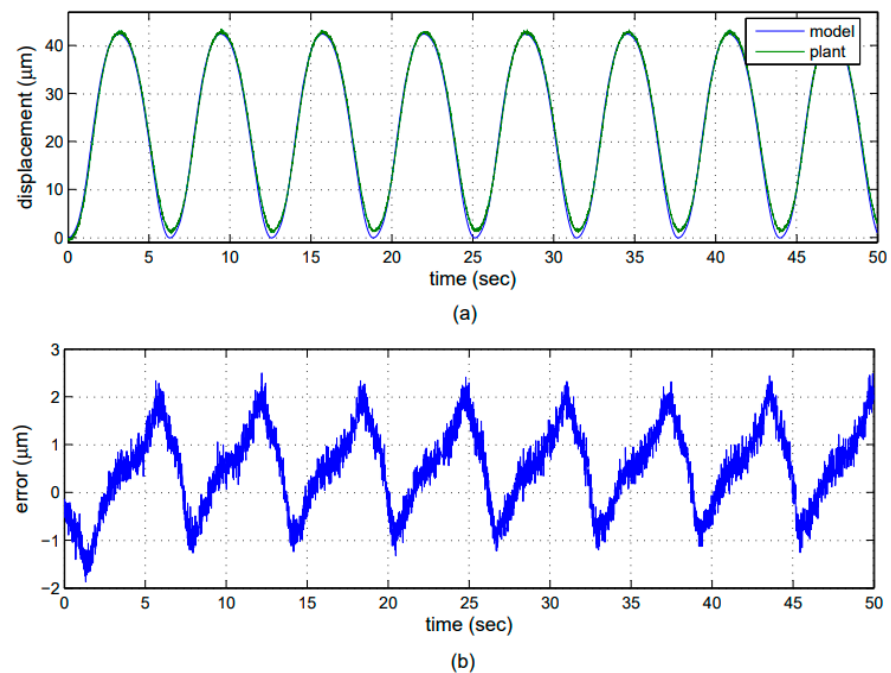


Figure 16. Prediction of the micro-piezoelectric actuator using the Preisach model realization via the table method: (a) the displacement of the micro-piezoelectric actuator and the output of the Preisach model; (b) the error between these two outputs.

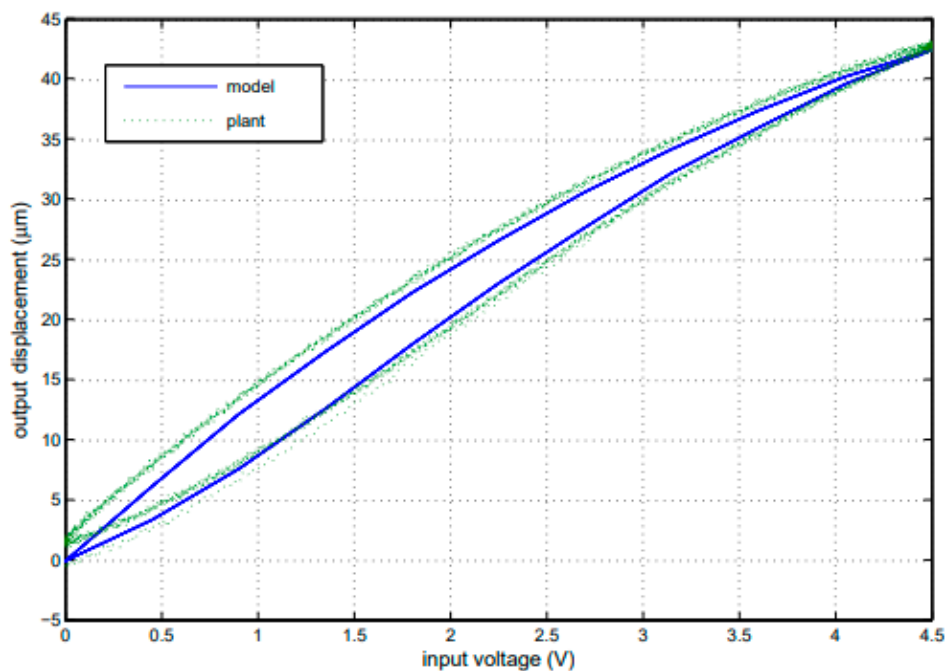


Figure 17. Phase transition of the micro-piezoelectric actuator using the Preisach model realization via the table method.

We apply the least-square approximation to sampling numbers of Table 5 to identify the polynomial coefficients directly in this experiment. The polynomial coefficients are stored in Table 6. The polynomial coefficient c_0 is not needed for the Preisach model realization. The displacement of the micro-piezoelectric actuator is compared with the output of the Preisach model which is realized via the table method as shown in Figure 18a, and the model error is shown in Figure 18b. The phase transition of the model and of the plant are plotted together in Figure 19. The RMS of the model error is $1.4817 \mu\text{m}$, and the amount of memory for polynomial coefficients is 22.

Table 6. The polynomial coefficients.

M	c_2	c_1	c_0
0	0	0	0
0.45	0	7.5778	0.02
0.90	-0.79012	9.2	0.09
1.35	-2.2099	12.328	0.1275
1.80	-1.1111	11.949	0.174
2.25	-1.0326	12.543	0.2525
2.70	-1.117	13.302	0.24619
3.15	-1.0256	13.439	0.37625
3.60	-1.091	13.954	0.39576
4.05	-1.058	14.075	0.49745
4.50	-1.0752	14.269	0.73091

We develop the least mean square (LMS) adaptive algorithm to obtain accuracy polynomial coefficients and the data in Table 6 are used as the initial parameters. We test the step sizes $\mu = 10^{-4}$, $\mu = 10^{-3}$, and $\mu = 0.00051 + u(t)$ and the process of each is shown in Figure 20. The polynomial coefficients obtained by the LMS adaptive algorithm with step size $\mu = 0.00051 + u(t)$ are stored in Table 7.

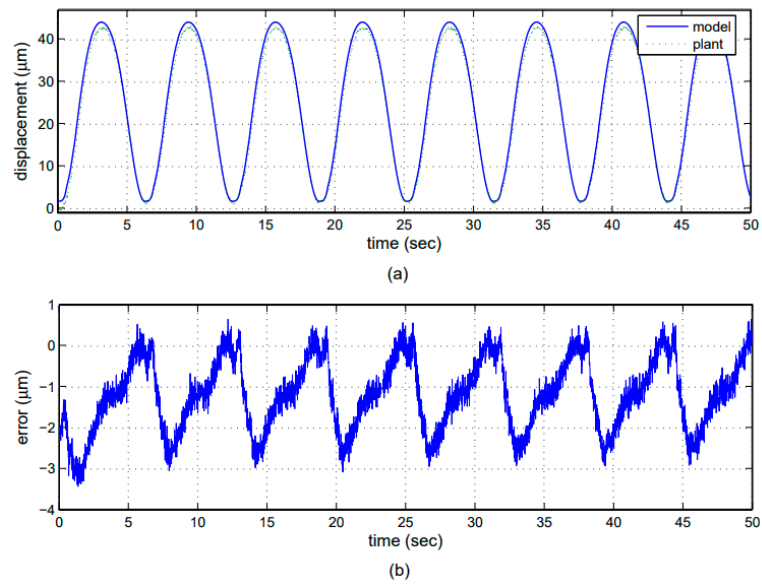


Figure 18. Prediction of the micro-piezoelectric actuator using the Preisach model realization via polynomial approximation: (a) the displacement of the micro-piezoelectric actuator and the output of the Preisach model; (b) the error between these two outputs.

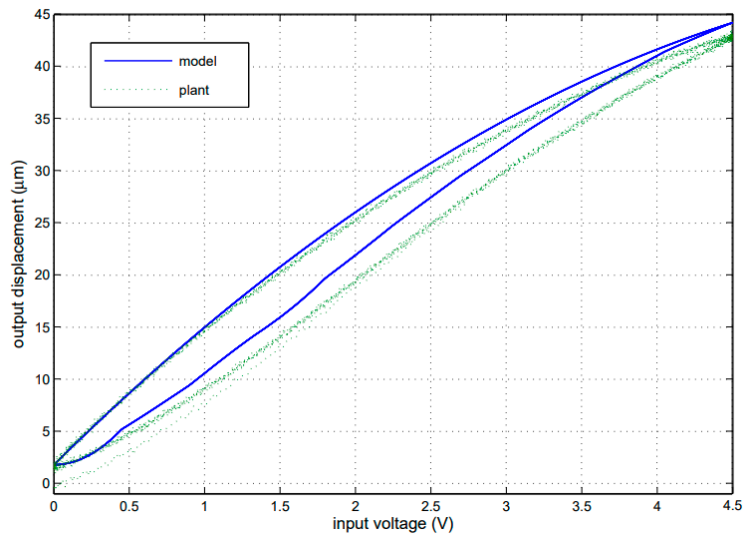


Figure 19. Phase transition of the micro-piezoelectric actuator using the Preisach model realization via polynomial approximation.

Table 7. The polynomial coefficients obtained using the LMS adaptive algorithm.

M	c_2	c_1
0	0.5718	3.8438
0.45	-2.7323	7.3383
0.90	-2.3325	9.2038
1.35	-2.1626	10.8419
1.80	-1.7700	11.8186
2.25	-1.4588	12.3549
2.70	-1.1540	12.4685
3.15	-1.2125	13.2395
3.60	-1.2565	13.9011
4.05	-1.1797	14.0324
4.50	-1.0938	13.9965

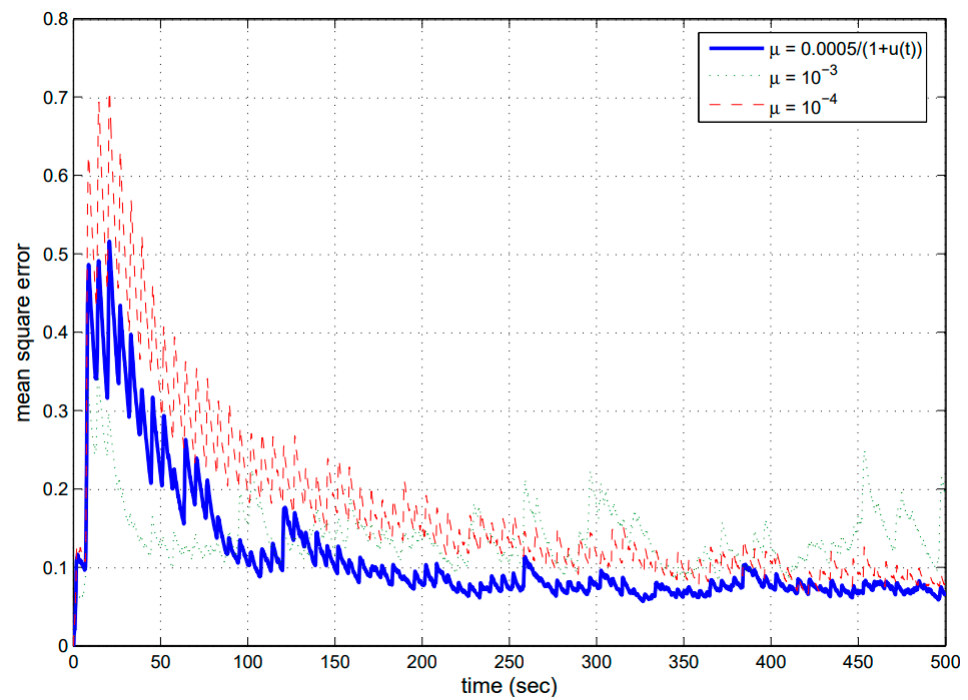


Figure 20. Mean square error of the prediction of the micro-piezoelectric actuator for varying step-size μ using the adaptive polynomial approximation.

The displacement of the micro-piezoelectric actuator is compared with the output of the Preisach model, which is shown in Figure 21a, and the model error is shown in Figure 21b. The phase transitions of the model and the plant are plotted together in Figure 22. The RMS of the model error is $1.5898 \mu\text{m}$.

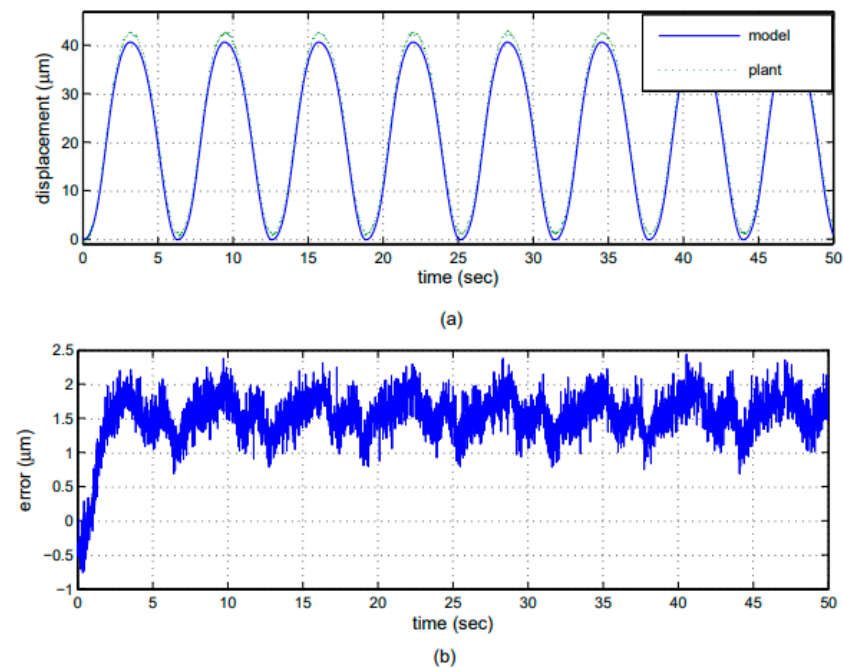


Figure 21. Prediction of the micro-piezoelectric actuator using the Preisach model realization via the adaptive polynomial approximation: (a) the displacement of the micro-piezoelectric actuator is compared with the output of the Preisach model; (b) the model error.

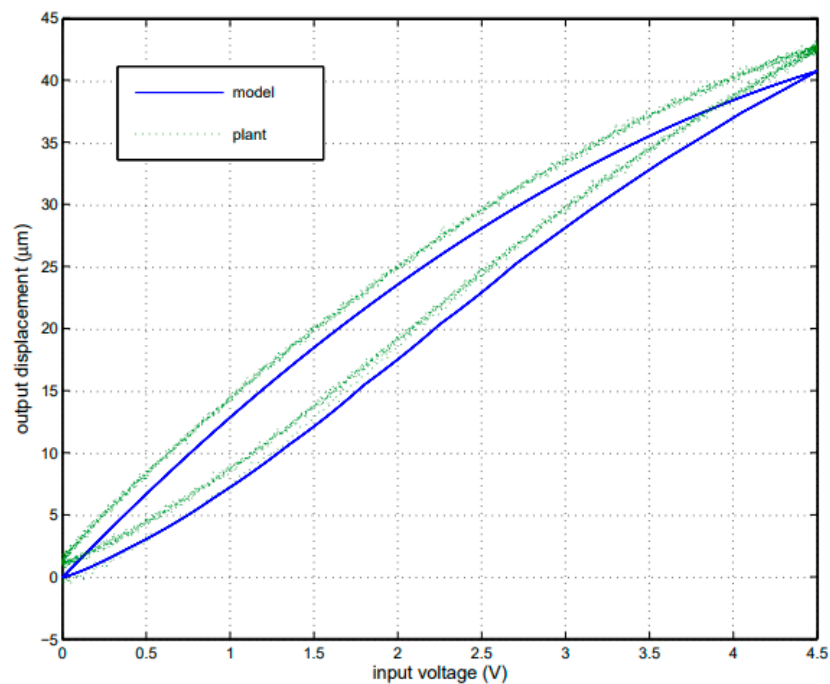


Figure 22. Phase transition of the micro-piezoelectric actuator using the Preisach model realization via the adaptive polynomial approximation.

Obviously, a bias exists in Figure 21b. This means that the Preisach model should be modified in the beginning of the micro-piezoelectric actuator work. Figure 23 shows the phase transition of the model error and the input in the beginning ascending branch.

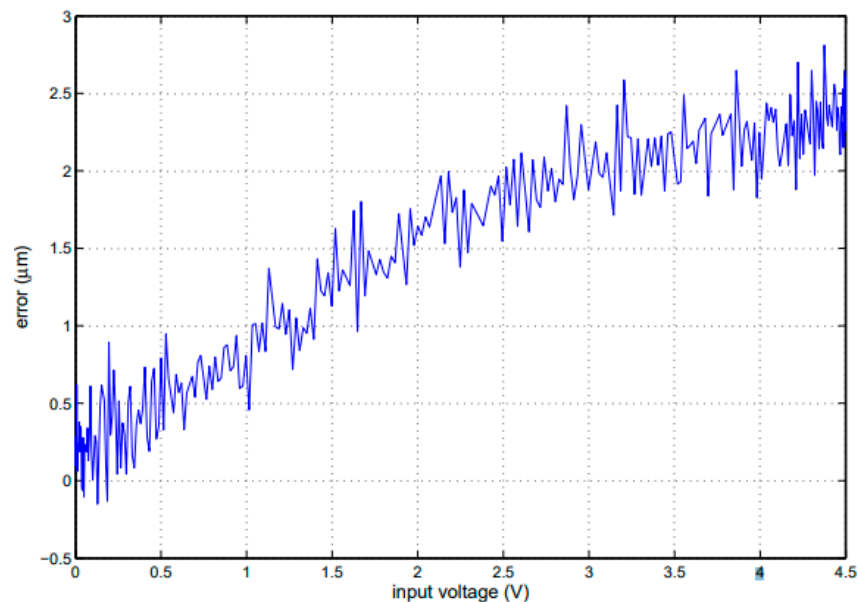


Figure 23. Phase transition of the model error and the input.

We predict the displacement of the micro-piezoelectric actuator again. The displacement of the micro-piezoelectric actuator is compared with output of the Preisach model which is shown in Figure 21a, and the model error is shown in Figure 24b. The phase transition of the model and the plant are plotted together in Figure 25. The RMS of the model error is $0.4539 \mu\text{m}$. The model error of the Preisach model realization via the table method and the modified polynomial approximation are plotted together in Figure 26 for comparison.

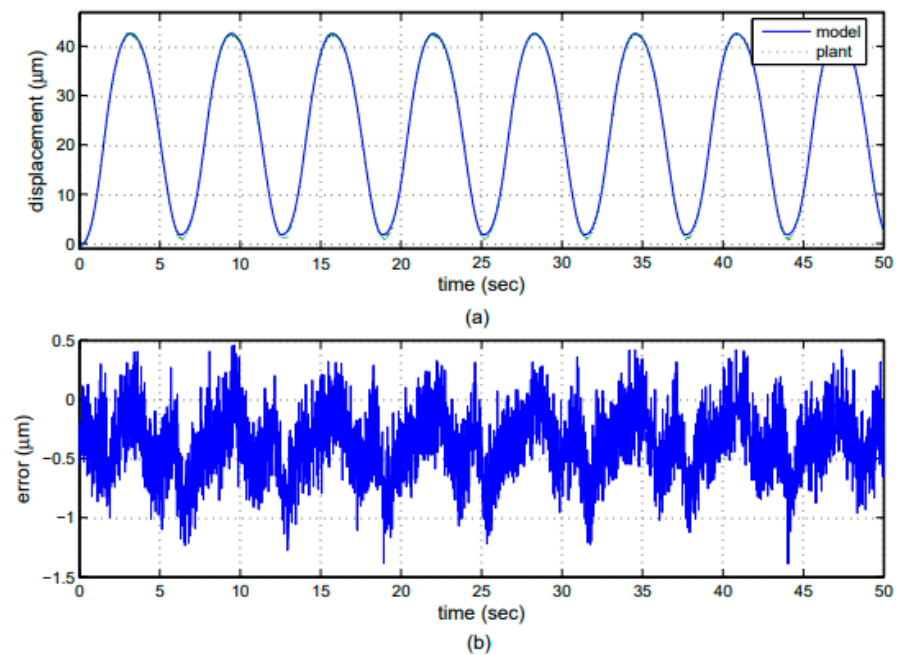


Figure 24. Prediction of the micro-piezoelectric actuator using the Preisach model realization via the modified polynomial approximation: (a) the displacement of the micro-piezoelectric actuator is compared with output of the Preisach model; (b) the model error.

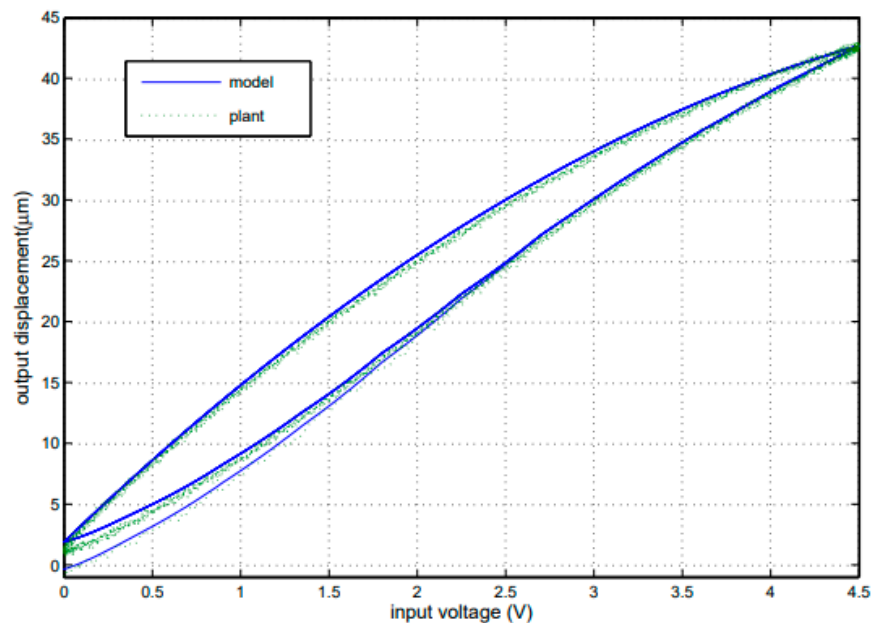


Figure 25. Phase transition of the micro-piezoelectric actuator using the Preisach model realization via the modified polynomial approximation.

An experimental inverse Preisach model for tracking control of micro-piezoelectric actuators was developed. The desired output compared with the displacement of the micro-piezoelectric of this experiment is plotted together in Figure 27a. The difference between these two outputs, defined as the tracking error, is shown in Figure 27a. The RMS of the tracking error is $0.7079 \mu\text{m}$. To reduce tracking error, we combine the inverse Preisach model with a PID controller. The parameters of the PID controller are $K_p = 0.0452$, $K_i = 2.9828$, and $K_d = 0.0024$. The desired output compared with the displacement of the micro-piezoelectric actuator of this experiment are plotted together in Figure 28a. The tracking error is shown in Figure 28b. The RMS of tracking error is $0.2438 \mu\text{m}$.

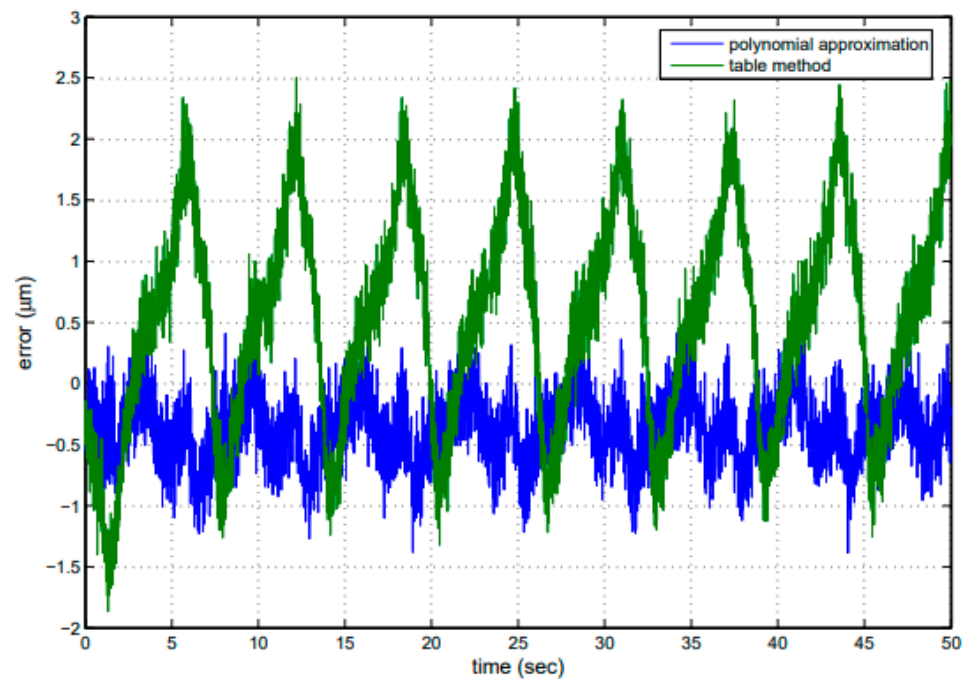


Figure 26. Model error of the Preisach model realization via the table method and the modified polynomial approximation.

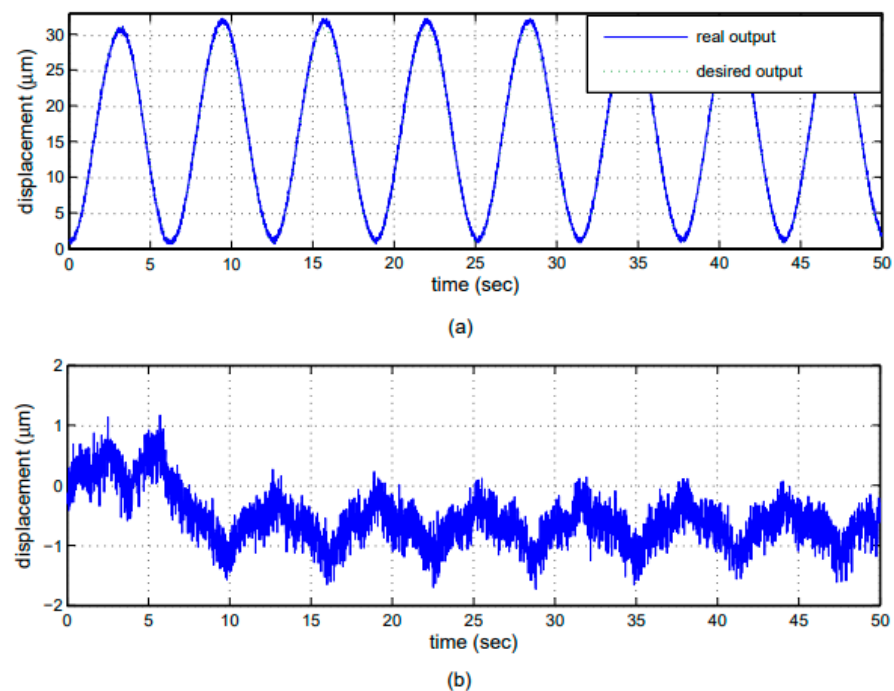


Figure 27. Tracking of the micro-piezoelectric actuator using the inverse Preisach model: (a) the desired output compared with the displacement of the micro-piezoelectric; (b) the difference between these two outputs.

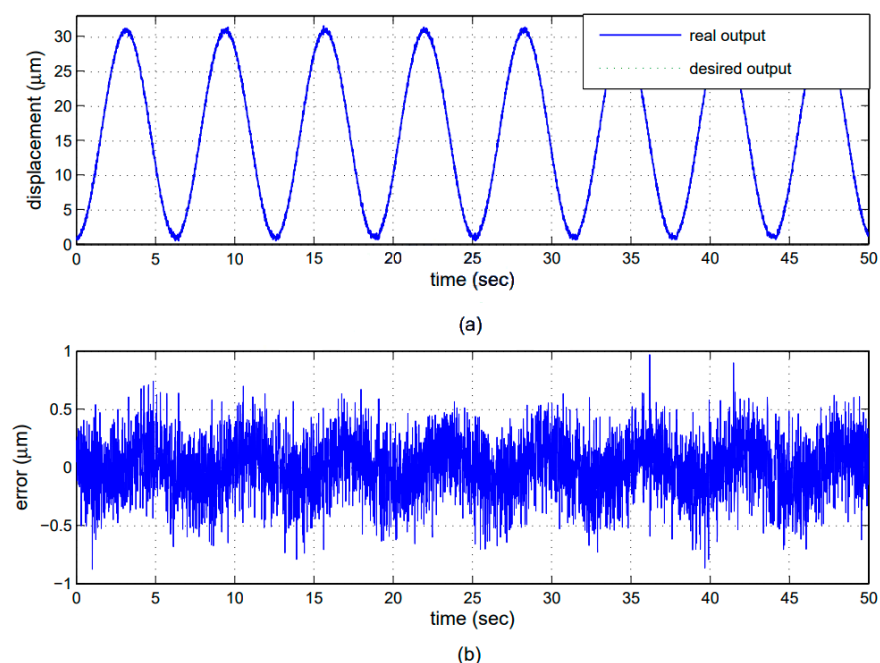


Figure 28. Tracking of the micro-piezoelectric actuator using the inverse Preisach model with a PID controller: (a) the desired output compared with the displacement of the micro-piezoelectric actuator; (b) the tracking error.

6. Conclusions

We implemented the CPM with a polynomial approximation to characterize the hysteresis. Compared with the traditional table method, this method requires less storage space and enables parameter tracking of the hysteresis element through experiments. We successfully obtained polynomial coefficients to model the CPM with a particular Preisach density function in Section 4, and the polynomial coefficients to model the displacement of a micro-piezoelectric actuator in Section 5. In Section 4, the obtained model compared with that via the table method not only requires less memory but also yields a smaller modeling RMS error from 1.0746 via the table method to 0.6349 as the CPM is changed. In Section 5, the modeling RMS error changed from 1.0098 μm via the table method to 0.4539 μm as CPM changes. We also established the inverse Preisach model based on the polynomial approximation; this model was combined with the PID controller used for tracking control and yielded small tracking RMS error of 0.2438 μm .

Author Contributions: All authors adopted the Preisach model to characterize hysteresis behaviors and proposed to use polynomial approximation to realize the Preisach model; this approach was then used to model the displacement of a real micro-piezoelectric actuator. The proposed approach can overcome the drawbacks of the conventional table method. Finally, all authors have given final approval of the version to be published and agree to be accountable for all aspects of the work in ensuring that questions related to the accuracy or integrity of any part of the work are appropriately investigated and resolved. All authors have read and agreed to the published version of the manuscript.

Funding: This research was sponsored by the Ministry of Science and Technology of Taiwan under the Grant MOST 110-2221-E-150-038.

Acknowledgments: This paper is in memory of Mu-Huo Cheng who was a professor in the Department of Electrical and Control Engineering, National Chiao Tung University, Hsinchu, Taiwan.

Conflicts of Interest: The authors declare no conflict of interest. Moreover, the funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Glossary

$u(t)$	signal input
$f(t)$	Preisach model output
$\mu(\alpha, \beta)$	Preisach density function
$\gamma_{\alpha\beta}[u(\cdot)](t)$	hysteresis operator
$\alpha(\beta)$	hysteresis operator corresponds to ‘up’ (‘down’) switching values of input
$\pm u_{sat}$	divides the range of input data
$u(\tau)$	input signal
$\{M_1, M_2, \dots, M_{n-1}\}$	peaks of the previous input signal
$\{m_1, m_2, \dots, m_{n-1}\}$	valleys of the previous input signal
$f_{\alpha\beta}(M, m)$	polynomial function
$C_i(M)$	indicates the coefficient corresponding with M
$C_0(M), C_1(M), C_2(M)$	polynomial coefficients
f_n	represents the effect of previous input
$f_p(t)$	polynomial approximation
$e(t)$	output error
$f(t)$	plant output
θ	update algorithm for the vector
T	transpose operation
$X(t)$	function of the input
η	constant step size
N	normalization factor

References

- Hussain, S.; Lowther, D.A. An Efficient Implementation of the Classical Preisach Model. *IEEE Trans. Magn.* **2018**, *54*, 7300204. [\[CrossRef\]](#)
- MacKenzie, I.; Trumper, D.L. Real-Time Hysteresis Modeling of a Reluctance Actuator Using a Sheared-Hysteresis-Model Observer. *IEEE/ASME Trans. Mechatron.* **2016**, *21*, 4–16. [\[CrossRef\]](#)
- Liang, H.; Zeng, C.; Giraud-Audine, C. Modelling and Simulation of Shape Memory Alloys Micro-Actuator. In Proceedings of the 2009 Fourth International Conference on Innovative Computing, Information and Control (ICICIC), Kaohsiung, Taiwan, 7–9 December 2009; pp. 1405–1408.
- Jia, W.; Luo, X.; Chen, X. Modeling and analysis of hysteresis in piezoceramic actuators excited by dynamic driving frequency. In Proceedings of the 2009 Symposium on Piezoelectricity, Acoustic Waves, and Device Applications (SPAWDA 2009), Wuhan, China, 17–20 December 2009; p. 116.
- Agrawal, B.N.; Elshafei, M.A.; Song, G. Adaptive antenna shape control using piezoelectric actuators. *Act. Astronaut.* **1997**, *40*, 821–826. [\[CrossRef\]](#)
- Mayhan, P.; Srinivasan, K.; Watechagit, S.; Washington, G. Dynamic modeling and controller design for a piezoelectric actuation system used for machine tool control. *J. Intell. Mater. Syst. Struct.* **2000**, *11*, 771–780. [\[CrossRef\]](#)
- Chen, X.; Su, C.; Fukuda, T. Adaptive Control for the Systems Preceded by Hysteresis. *IEEE Trans. Autom. Control* **2008**, *53*, 1019–1025. [\[CrossRef\]](#)
- Tao, G.; Kokotovic, P.V. Adaptive control of plants with unknown hystereses. *IEEE Trans. Autom. Control* **1995**, *40*, 200–212. [\[CrossRef\]](#)
- Zhang, J.; Merced, E.; Sepúlveda, N.; Tan, X. Modeling and Inverse Compensation of Nonmonotonic Hysteresis in VO₂-Coated Microactuators. *IEEE/ASME Trans. Mechatron.* **2014**, *19*, 579–588. [\[CrossRef\]](#)
- Li, Z.; Zhang, X.; Su, C.-Y.; Chai, T. Nonlinear Control of Systems Preceded by Preisach Hysteresis Description: A Prescribed Adaptive Control Approach. *IEEE Trans. Control Syst. Technol.* **2016**, *24*, 451–460. [\[CrossRef\]](#)
- Riccardi, L.; Naso, D.; Turchiano, B.; Janocha, H. Adaptive Control of Positioning Systems with Hysteresis Based on Magnetic Shape Memory Alloys. *IEEE Trans. Control Syst. Technol.* **2013**, *21*, 2011–2023. [\[CrossRef\]](#)
- Stoleriu, L.; Stancu, A. Using Experimental FORC Distribution as Input for a Preisach-Type Model. *IEEE Trans. Magn.* **2006**, *42*, 3159–3161. [\[CrossRef\]](#)
- Ge, P.; Jouaneh, M. Modeling hysteresis in piezoceramic actuators. *Prec. Eng.* **2005**, *17*, 211–221. [\[CrossRef\]](#)
- Liu, V.-T.; Lin, C.-L.; Wing, H.-Y. New realisation of Preisach model using adaptive polynomial approximation. *Int. J. Syst. Sci.* **2012**, *43*, 1642–1649. [\[CrossRef\]](#)
- Liu, V.-T.; Lin, C.-L.; Wing, H.-Y. Realization of Preisach model using adaptive polynomial approximation method. In Proceedings of the 2010 5th IEEE Conference on Industrial Electronics and Applications, Taichung, Taiwan, 15–17 June 2010; pp. 1258–1263.
- Mayergoyz, I.D. *Mathematical Models of Hysteresis*; Springer: New York, NY, USA, 1991.

17. Mayergoyz, I.D. Hysteresis models from the mathematical and control theory points of view. *J. Appl. Phys.* **1985**, *57*, 3803–3805. [[CrossRef](#)]
18. Choi, S.B.; Han, Y.M. Hysteretic behavior of a magnetorheological fluid: Experimental identification. *Acta Mech.* **2005**, *180*, 37–47. [[CrossRef](#)]
19. Iyer, R.V.; Shirley, M.E. Hysteresis Parameter Identification with Limited Experimental Data. *IEEE Trans. Magn.* **2004**, *40*, 3227–3239. [[CrossRef](#)]
20. Tan, X.; Baras, J.S. Adaptive Identification and Control of Hysteresis in Smart Materials. *IEEE Trans. Autom. Control* **2005**, *50*, 827–839.
21. Ge, P.; Jouaneh, M. Tracking control of a piezoceramic actuator. *IEEE Trans. Control Syst. Technol.* **1996**, *4*, 209–216.
22. Tang, Z.-H.; Lv, F.-U.; Xiang, Z.-H. Hysteresis model of magnetostrictive actuators and its numerical realization. *J. Zhejiang Univ. Sci. A* **2007**, *8*, 1059–1064. [[CrossRef](#)]