Control of Hysteresis: Theory and Experimental Results

by Xiaobo Tan, Ram Venkataraman, P.S. Krishnaprasad

TR 2001-30
### Control of Hysteresis: Theory and Experimental Results

**Title and Subtitle:** Control of Hysteresis: Theory and Experimental Results

**Author:**

**Performing Organization:** Army Research Office, PO Box 12211, Research Triangle Park, NC, 27709

**Distribution/Availability Statement:** Approved for public release; distribution unlimited

**Supplementary Notes:** The original document contains color images.

### Abstract

See report

### Security Classification

- **a. Report:** Unclassified
- **b. Abstract:** Unclassified
- **c. This Page:** Unclassified

### Limitation of Abstract

13 pages

### Sponsor/Monitor’s Report Number

- **10. Sponsor/Monitor’s Acronym(s):**
- **11. Sponsor/Monitor’s Report Number(s):**

---

The original document contains color images.
Control of hysteresis: theory and experimental results

Xiaobo Tan†, Ram Venkataraman‡, and P. S. Krishnaprasad†

†Institute for Systems Research and
Department of Electrical and Computer Engineering
University of Maryland, College Park, MD 20742 USA
‡Veridian Engineering and
Wright-Patterson AFB, OH 45433-7531 USA

ABSTRACT
Hysteresis in smart materials hinders the wider applicability of such materials in actuators. In this paper, a systematic approach for coping with hysteresis is presented. The method is illustrated through the example of controlling a commercially available magnetostrictive actuator.

We utilize the low-dimensional model for the magnetostrictive actuator that was developed in earlier work. For low frequency inputs, the model approximates to a rate-independent hysteresis operator, with current as its input and magnetization as its output. Magnetostrictive strain is proportional to the square of the magnetization. In this paper, we use a classical Preisach operator for the rate-independent hysteresis operator.

In this paper, we present the results of experiments conducted on a commercial magnetostrictive actuator, the purpose of which was the control of the displacement/strain output. A constrained least-squares algorithm is employed to identify a discrete approximation to the Preisach measure. We then discuss a nonlinear inversion algorithm for the resulting Preisach operator, based on the theory of strictly-increasing operators. This algorithm yields a control input signal to produce a desired magnetostrictive response. The effectiveness of the inversion scheme is demonstrated via an open-loop trajectory tracking experiment.

Keywords: Control, hysteresis, Preisach, identification, inversion, magnetostriction, smart actuator

1. INTRODUCTION
Hysteresis in smart materials, e.g., magnetostrictives, piezoceramics, and Shape Memory Alloys (SMAs), hinders the wider applicability of such materials in actuators. A fundamental idea in coping with hysteresis is to formulate the mathematical model of hysteresis and use inverse compensation to cancel out the hysteretic effect. This idea can be found in the works of Tao and Kokotović,1 Smith,2 and Galinaitis and Rogers,3 to name a few.

Hysteresis models for smart materials can be classified into physics based models and phenomenological models. An example of physics based model is the Jiles-Atherton model for ferromagnetic hysteresis,4 where hysteresis is considered to arise from pinning of domain walls on defect sites. The most popular phenomenological hysteresis model used in control of smart actuators has been the Preisach model.5–8 A similar type of operator, called Krasnosel’skii-Pokrovskii (KP) operator was used by Galinaitis and Rogers in modeling a piezoelectric actuator.3 Though the Preisach model does not provide any physical insight into the problem, it provides a means of developing phenomenological models that are capable of producing behaviors similar to physical systems (see Mayergoyz9 for an excellent exposition).

In this paper we present a systematic approach for control of smart actuators. The method is illustrated through the example of controlling a commercially available magnetostrictive actuator. The model we use for the magnetostrictive actuator is based on earlier work of our group.10,11 It was shown that a key component of a low-order model is as shown in Figure 4. The rate-independent hysteresis operator is a classical Preisach operator followed by a squaring operator. The input of the hysteresis operator is the current input to the actuator, and the output of the squaring operator is a quantity with the dimension of force. In this paper, we deal with low frequency inputs...
of less than 5 Hz, as we wish to focus on the hysteresis operator only. For this case, the linear system in Figure 4 approximates to a constant with dimension of length/force.

A constrained least-squares algorithm is employed to identify a discrete approximation to the Preisach measure (also called weighting function in literature). We then discuss a nonlinear inversion algorithm for the resulting Preisach operator, based on the theory of strictly-increasing operators. This algorithm, called closest match algorithm, yields a control input signal producing the closest match to a desired magnetostrictive response.

Previous work closest to this paper was by Hughes and Wen.\(^6,\!^7\) Although both their papers and this paper deal with the themes of (a) identification of a Preisach operator; (b) numerical inverse computation of the Preisach operator and (c) experimental implementation, there are significant differences both in the method and complexity. First, Hughes and Wen use a first-order reversal curve and polynomial fitting to identify a continuous Preisach density function, while we use constrained least-squares method to identify a discrete Preisach measure. Second, Hughes and Wen use an implicit function theorem approach to exactly invert the Preisach operator, whereas we only try to find the input yielding the closest match to the desired output. This difference is significant both mathematically and in computational savings. In Venkataraman and Krishnaprasad,\(^1,^2\) we have shown that in general, one can only hope to approximately invert the Preisach operator with a continuous density function, and trying to exactly invert it is fraught with numerical ill-conditioning. The implementation of the closest match algorithm is both numerically robust and time-saving.

The remainder of the paper is organized as follows. Section 2 provides an introduction to the Preisach operator, where the emphasis is on how to evaluate the output of the Preisach operator given an initial memory curve and the input. Discretization scheme and identification algorithm will be discussed in Section 3. Section 4 is devoted to the closest match inverse algorithm, which fully exploits the strictly-increasing property of the Preisach operator and the discrete structure of the problem. Experimental results are given in Section 5. Concluding remarks and discussions on possible future work are provided in Section 6.

2. THE PREISACH MODEL

Consider a simple hysteretic element (relay) shown in Figure 1. The relationship between the “input” variable \(u\) and the “output” variable \(v\) at each instant of time \(t\) can be described by:

\[
\begin{align*}
  v &= +1 \quad \text{if} \quad u > \alpha, \\
  v &= -1 \quad \text{if} \quad u < \beta, \\
  v &= \text{remains unchanged if} \quad \beta \leq u \leq \alpha.
\end{align*}
\]  

(1)

Call the operator relating \(u(\cdot)\) to \(v(\cdot)\) as \(\tilde{\gamma}_{\beta,\alpha}[u(\cdot)]\), where we now view the input and output variables as functions of time. This operator is sometimes referred to as an elementary Preisach hysterons since it is a basic block from which the Preisach operator \(\Gamma[\cdot]\) will be constructed. We now outline this construction. Suppose \(u(\cdot) \in C[0, T]\) is the input to the elementary hysterons. The output of the Preisach operator is defined as:

\[
\omega(t) = \Gamma[u](t) = \int \int_{\alpha \geq \beta} \mu(\beta, \alpha) \tilde{\gamma}_{\beta,\alpha}[u](t) d\beta d\alpha,
\]

(2)

where \(\mu(\cdot, \cdot)\) is a density function (also called weighting function or the Preisach measure). This representation is the most natural one for the Preisach operator and is closest to Preisach’s original definition.\(^9\) The Preisach operator has non-local memory and it “remembers” the dominant maximum and minimum values of the past input. For a review of this and other basic properties of the Preisach operator, please refer to Mayergoyz and Brokate and Sprekels.\(^1,^{13}\)

The memory effect of the Preisach operator can be captured by curves in the Preisach \((\beta, \alpha)\) plane. To simplify the discussion, we assume the Preisach measure \(\mu(\cdot, \cdot)\) has a compact support, i.e., \(\mu(\beta, \alpha) = 0\) if \(\beta < \beta_0\) or \(\alpha > \alpha_0\) for some \(\beta_0, \alpha_0\). Then the Preisach plane \(P \triangleq \{(\beta, \alpha) | \alpha \geq \beta, \beta \geq \beta_0, \alpha \leq \alpha_0\}\), as shown in Figure 2(a). Each \((\beta, \alpha) \in P\) is identified with the hysterons \(\tilde{\gamma}_{\beta,\alpha}\). At each time instant \(t\), \(P\) can be divided into two regions:

\[
\begin{align*}
  P_-(t) &\triangleq \{(\beta, \alpha) \in P | \text{output of } \tilde{\gamma}_{\beta,\alpha} \text{ at } t = -1\}, \\
  P_+(t) &\triangleq \{(\beta, \alpha) \in P | \text{output of } \tilde{\gamma}_{\beta,\alpha} \text{ at } t = +1\},
\end{align*}
\]
Figure 1. Illustration of the elementary Preisach hysteron

so that $P = P_-(t) \cup P_+(t)$ at all times. Equation (2) can be rewritten as:

$$\omega(t) = \int \int_{P_+(t)} \mu(\beta, \alpha)d\beta d\alpha - \int \int_{P_-(t)} \mu(\beta, \alpha)d\beta d\alpha.$$  \hspace{1cm} (3)

Now assume that at some initial time $t_0$, the input $u(t_0) = u_0 < \beta_0$. Then the output of every hysteron operator is -1. Therefore $P_-(t_0) = P$, $P_+(t_0) = \emptyset$ and it corresponds to the “negative saturation” (Figure 2(b)). Next we assume that the input is monotonically increased to some maximum value at $t_1$ with $u(t_1) = u_1$. The output of $\hat{\gamma}_{\beta, \alpha}$ is switched to +1 as the input $u(t)$ increases past $\alpha$. Thus at time $t_1$, the boundary between $P_-(t_1)$ and $P_+(t_1)$ is the horizontal line $\alpha = u_1$ (Figure 2(c)). Next we assume that the input starts to decrease monotonically until it stops at $t_2$ with $u(t_2) = u_2$. It’s easy to see that the output of $\hat{\gamma}_{\beta, \alpha}$ becomes -1 as $u(t)$ sweeps past $\beta$, and correspondingly, a vertical line segment $\beta = u_2$ is generated as part of the boundary (Figure 2(d)). Further input reversals generate additional horizontal or vertical boundary segments.

Figure 2. Memory curve in the Preisach plane

From the above illustration, we can see that each of $P_-$ and $P_+$ is a connected set, and the output of the Preisach operator is determined by the boundary between $P_-$ and $P_+$. The boundary is also called the memory curve, since it
provides information about the state of any hysteron. At any time instant, the memory curve is a piecewise constant, nonincreasing function of \( \beta \). We also note that due to its staircase structure, the memory curve is fully captured by its corner points, which correspond exactly to the past dominant maximum and minimum input values. Motivated by this observation, we store and update only these corner points in our numerical implementation of the Preisach model.

3. IDENTIFICATION OF THE PREISACH MEASURE

3.1. Review of identification methods

A classical method for identifying the Preisach measure is using the so called first order reversal curves, detailed in Mayergoyz.\(^9\) A first order reversal curve can be generated by first bringing the input to the negative saturation, followed by a monotonic increase to \( \alpha \), then a monotonic decrease to \( \beta \). The term “first order reversal” comes from the fact that each of these curves is formed after the first reversal of the input. Denote the output value as \( f(\beta, \alpha) \) when the input reaches \( \beta \). Then the measure \( \mu(\beta, \alpha) \) can be obtained as

\[
\mu(\beta, \alpha) = \frac{1}{2} \frac{\partial^2 f(\beta, \alpha)}{\partial \beta \partial \alpha}.
\]  

Equation (4) is useful only when the two-dimensional surface \( f(\beta, \alpha) \) is twice differentiable, which is not the case for measured curves in experiments. To overcome this difficulty, a smooth approximation surface \( \bar{f}(\beta, \alpha) \) is fit to the data points.\(^6\)–\(^8\) Hughes and Wen\(^6,7\) approximated the surface by polynomials using a least squares method. Gorbet, Wang and Morris employed functions with specific forms, and the parameters were obtained via a weighted least square algorithm.\(^8\) As pointed out in Gorbet, Wang and Morris,\(^8\) deriving the measure by differentiating a fitted surface is inherently imprecise, since different type of approximating functions lead to quite different measure distributions.

Hoffmann and Sprekels\(^14\) proposed a scheme to identify the Preisach measure directly. By devising the input sequence carefully, they set up independent blocks of linear equations involving the output measurements and the measure masses in the discretized Preisach plane. Each block of equations can be solved successively to obtain the measure. This scheme is very sensitive to experimental errors as one can easily see. Using the identified discrete measure,\(^14\) Hoffmann and Meyer\(^15\) approximated the (continuous) Preisach measure in terms of a set of basis functions. A least squares method was applied to compute the coefficients.

Another way to obtain the measure is driving the system with a “reasonably” rich input signal, measuring the output and then estimating the Preisach measure by a least squares method. This idea appeared in the work of Banks and his colleagues,\(^16,17\) where they investigated the identification problem of the KP operator. Galinaitis and Rogers\(^3\) used the same idea to identify the weights for a discretized KP operator. We will also adopt the least squares method for measure identification in this paper.

3.2. Identification scheme

Magnetostrictive actuators, due to the capacity of the windings or other practical reasons, have to be operated with their inputs within a specific range. As a consequence, we will not be able to visit the whole Preisach plane and identify the measure everywhere during the identification process. We assume the input range is \([u_{\text{min}}, u_{\text{max}}]\). In Figure 3, the big triangle represents the whole Preisach plane, while the smaller triangle is the region we can visit and we denote it by \( \Omega_1 \). The region outside \( \Omega_1 \) in the Preisach plane is denoted by \( \Omega_0 \). Since the input \( u(t) \) never goes beyond the limits, states of hysteron in \( \Omega_0 \) remain unchanged. Thus the bulk contribution to the output from \( \Omega_0 \) is a constant and we denote it by \( \omega_0 \).

The input is discretized into \( L + 1 \) levels uniformly and we label the cells in the grid as illustrated in Figure 3 for \( L = 9 \). The measure mass inside each cell is assumed to concentrate at the cell center. The quantities we want to identify include measure masses \( \mu_{ij}, i = 1, \cdots, L, j = 1, \cdots, i \) and \( \omega_0 \). To simplify the discussion, with a little bit notation abuse, we write \( \mu_{ij}, i = 1, \cdots, L, j = 1, \cdots, i \) into a column vector \( \mu_k, k = 1, \cdots, K \), where \( K = \frac{L(L+1)}{2} \).

To initialize the states of hysteron, we first increase the input to \( u_{\text{max}} \) and then reduce it to \( u_{\text{min}} \). This sets the state of each hysteron in \( \Omega_1 \) to -1. We may also initialize each hysteron in \( \Omega_1 \) to +1 by decreasing the input to \( u_{\text{min}} \) followed by bringing it to \( u_{\text{max}} \). We then apply some piecewise monotone continuous input \( u(t), t \in [0, T] \) which contains sufficiently rich information (by this we mean \( \Omega_1 \) should be visited completely), and measure the
Figure 3. Discretization of the Preisach plane \((L = 9)\)

output \(\omega(t)\). Signal \(u(t), \omega(t)\) are sampled into sequences \(u[n], \omega[n], n = 1, \cdots, N\). The input sequence \(u[n]\) (after discretization) is fed into the discretized Preisach operator and the state of each hysteron, \(\hat{\gamma}_k[n], k = 1, \cdots, K\) is computed. The output of the Preisach model at time instant \(n\) can be expressed as:

\[
\tilde{\omega}[n] = \omega_0 + \sum_{k=1}^{K} \mu_k \hat{\gamma}_k[n],
\]  

where \(\mu_k\)'s are yet to be found.

We use the least squares method to estimate the parameters, i.e. the parameters are determined in such a way that

\[
\sum_{n=1}^{N} |\omega[n] - \tilde{\omega}[n]|^2
\]

is minimized. This can be written in a more compact form. Let

\[
\theta = \begin{bmatrix}
\mu_1 \\
\mu_2 \\
\vdots \\
\mu_K \\
\omega_0
\end{bmatrix}, \quad A = \begin{bmatrix}
\hat{\gamma}_1[1] & \hat{\gamma}_2[1] & \cdots & \hat{\gamma}_K[1] & 1 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
\hat{\gamma}_1[N] & \hat{\gamma}_2[N] & \cdots & \hat{\gamma}_K[N] & 1
\end{bmatrix}, \quad b = \begin{bmatrix}
\omega[1] \\
\omega[2] \\
\vdots \\
\omega[N]
\end{bmatrix}.
\]

Then the problem becomes finding the parameter vector \(\theta\), such that

\[
\|A\theta - b\|
\]

is minimized, where \(\|\cdot\|\) stands for the Euclidean norm in \(\mathbb{R}^N\). Since we require \(\mu_k \geq 0, k = 1, \cdots, K\), it is a least square error optimization problem with constraints.

What we have identified is a discrete approximation to the Preisach measure. Sometimes a continuous measure function is of more interest, e.g., when we want to study the analytic properties of the system. In that case, we may fit a smooth function from the identified discrete measure.
4. INVERSION OF THE PREISACH OPERATOR

The general structure of models for smart actuators that capture both hysteresis and dynamic behaviour is shown in Figure 4. In the figure, $G(s)$ represents the transfer function of the linear part in the actuator, while $W$ denotes a rate-independent hysteretic nonlinearity. Venkataraman\textsuperscript{10} has shown that a key component of a low-order model for magnetostriction in Terfenol-D has a structure resembling Figure 4.

\begin{figure}[h]
\centering
\includegraphics[width=0.8\textwidth]{structure_models}
\caption{Structure of models for smart actuators}
\end{figure}

A basic idea for controller synthesis for such systems is to design a right inverse operator $W^{-1}$ for $W$ as shown in Figure 5. Then $\omega(\cdot) = \bar{u}(\cdot)$ and the controller design problem is reduced to designing a linear controller $K(s)$ for the linear system $G(s)$.

\begin{figure}[h]
\centering
\includegraphics[width=0.8\textwidth]{controller}
\caption{Controller design schematic}
\end{figure}

In the context of this paper, we consider $W$ as a Preisach operator. The Preisach operator is highly nonlinear, and in general, we cannot find closed form formulas for the inverse operator, unless the measure function is of some special form, like in the work of Galinaitis and Rogers.\textsuperscript{18} Hughes and Wen\textsuperscript{6,7} utilized the first order reversal curves $f(\beta, \alpha)$ in computing the numerical inverse of the Preisach operator. They defined $F(\beta, \alpha) = \frac{1}{2}[f(\alpha, \alpha) - f(\beta, \alpha)]$ (called the Everett surface). For output change $\Delta\omega$,

$$\Delta\omega = 2F(\beta, \alpha)$$

defines implicitly two inverse maps $G_\alpha(\cdot, \cdot)$ and $G_\beta(\cdot, \cdot)$:

$$\alpha = G_\alpha(\Delta\omega, \beta),$$

$$\beta = G_\beta(\beta, \Delta\omega).$$

$G_\alpha(\cdot, \cdot)$ and $G_\beta(\cdot, \cdot)$ is then used to compute the desired input given the desired output and past input history. This method relies on measurement of all first order reversal curves and therefore is subject to experimental errors. Also computing $G_\alpha, G_\beta$ involves solving nonlinear equations and therefore is not a trivial task.

4.1. Contraction mapping algorithm for inversion

Under mild assumptions, the Preisach operator is Lipshitz continuous and incrementally strictly increasing.\textsuperscript{13} Venkataraman and Krishnaprasad\textsuperscript{19,12} exploited these properties and proposed an inversion algorithm for the Preisach operator based on the contraction mapping principle. The algorithm is summarized in the following theorem:
Theorem 4.1: [Contraction Mapping Algorithm for Inversion] \(^{19,12}\) Let \(X = C_I[0,T]\), where \(I = [a,b]\). 
\(C_I[0,T]\) denotes the space of continuous functions defined on \([0,T]\) taking values in \(I\). Let \(\Gamma : X \rightarrow Y\), be an incrementally strictly increasing, strongly Lipshitz continuous Preisach operator (with Lipshitz constant \(k_2\)) with some initial memory curve \(\psi_{-1}\), and \(Y\) is the range of \(\Gamma\). Given \(\epsilon > 0\) and the operator equation
\[
\Gamma(x) = y, \tag{10}
\]
where \(y \in Y\) is known, consider the algorithm:

- pick any \(x_0 \in X\);
- while \(\|x_n - x_{n-1}\| \geq \epsilon\):

\[
x_{n+1} = \frac{1}{k_2}(y + (k_2x_n - \Gamma(x_n))). \tag{11}
\]

The sequence \(\{x_n\}\) terminates at \(z\) which satisfies \(\|z - x\| \leq \epsilon\) where \(x\) is the solution of (10). The rate of convergence is linear.

When the Preisach measure is discretized, the Preisach operator is no longer Lipschitz continuous and therefore the contraction mapping algorithm for inversion does not work efficiently. Indeed, since the output of \(\Gamma\) at any time instant can take only finite number of possible values (due to the finite number of measure atoms), from (11), we see that for almost all \(y\),
\[
\|x_n - x_{n-1}\| = \frac{1}{k_2}\|y - \Gamma(x_{n-1})\| \geq \delta_y > 0.
\]
In addition, the discrete measure masses are, in general, not uniform. These factors make it difficult to choose an appropriate stopping criterion \(\epsilon\): picking \(\epsilon\) big we lose accuracy; picking \(\epsilon\) small we will get stuck if \(\epsilon < \delta_y\). Note that these remarks don’t invalidate Theorem 4.1: Theorem 4.1 works perfectly for continuous measure distribution. However, we need a more practical inversion algorithm for the discrete measure case.

4.2. Closest match algorithm for inversion

We propose a new inversion algorithm in this subsection. This algorithm, like the contraction mapping algorithm, is also based on the strictly increasing property of the Preisach operator. It fully utilizes the discrete structure of the problem. We name it closest match algorithm because it always generates input whose output matches the desired output most closely among all possible inputs.

Note that due to discretization, the input can only take values from a finite set \(U \triangleq \{u_l, 1 \leq l \leq L + 1\}\) with each \(u_l, 1 \leq l \leq L + 1\), representing an input level. To be precise, let
\[
\Delta u = \frac{u_{\max} - u_{\min}}{L},
\]
then \(u_l = u_{\min} + (l-1) \Delta u\). Thus \(u_1 = u_{\min}\) and \(u_{L+1} = u_{\max}\). The inversion problem is: given an initial memory curve \(\psi^{(0)}\) (from which the initial input \(u^{(0)}\) and output \(\omega^{(0)}\) can be derived) and a desired output \(\bar{\omega}\), find \(u^* \in U\), such that
\[
|\Gamma(u^*; \psi^{(0)}) - \bar{\omega}| = \min_{u \in U} |\Gamma(u; \psi^{(0)}) - \bar{\omega}|. \tag{12}
\]
Also the algorithm should return the resulting memory curve \(\psi^*\) for later use. Note in (12) we explicitly put \(\psi^{(0)}\) as argument of \(\Gamma\) to emphasize the effect of the memory curve on the output.

The intuitive idea is as follows. Consider the case where the current output \(\omega^{(0)}\) is less than the desired value \(\bar{\omega}\) (the case \(\omega^{(0)} > \bar{\omega}\) is treated in exactly the same way with some obvious modification). We keep increasing the input by one level in each iteration until, say at iteration \(n\), the input \(u^{(n)}\) reaches \(u_{\max}\), or the output \(\omega^{(n)}\) corresponding to \(u^{(n)}\) exceeds \(\bar{\omega}\). For the first case, the optimal input is clearly \(u_{\max}\); for the second case, two candidates for the optimal input \(u^*\) are \(u^{(n-1)}\) and \(u^{(n)}\). We then take \(u^*\) to be the one with smaller output error. Note that we need back up the memory curve whenever we increase input, so that we can always retrieve the consistent memory curve with \(u^*\).

We now describe the algorithm in detail.
Closest Match Algorithm:

- **Step 0** [Initialization]. Set \( n = 0 \). Compare \( \omega^{(0)} \) and \( \bar{\omega} \): if \( \omega^{(0)} = \bar{\omega} \), let \( u^* = u^{(0)} \), \( \psi^* = \psi^{(0)} \), go to Step 3; if \( \omega^{(0)} < \bar{\omega} \), go to Step 1.1; otherwise go to Step 2.1.

- **Step 1** [Case \( \omega^{(0)} < \bar{\omega} \)].
  - Step 1.1: If \( u^{(n)} = u_{\text{max}} \), let \( u^* = u^{(n)} \), \( \psi^* = \psi^{(n)} \), go to Step 3; otherwise \( u^{(n+1)} = u^{(n)} + \triangle u, \psi = \psi^{(n)} \)[back up the memory curve], \( n = n + 1 \), go to Step 1.2;
  - Step 1.2: Evaluate \( \omega^{(n)} = \Gamma(u^{(n)}; \psi^{(n-1)}) \), and (at the same time) update the memory curve to \( \psi^{(n)} \). Compare \( \omega^{(n)} \) with \( \bar{\omega} \): if \( \omega^{(n)} = \bar{\omega} \), let \( u^* = u^{(n)} \), \( \psi^* = \psi^{(n)} \), go to Step 3; if \( \omega^{(n)} < \bar{\omega} \), go to Step 1.1; otherwise go to Step 1.3;
  - Step 1.3: If \( |\omega^{(n)} - \bar{\omega}| \leq |\omega^{(n-1)} - \bar{\omega}| \), let \( u^* = u^{(n)} \), \( \psi^* = \psi^{(n)} \), go to Step 3; otherwise \( u^* = u^{(n-1)} \), \( \psi^* = \bar{\psi} \)[restore the memory curve], go to Step 3;

- **Step 2** [Case \( \omega^{(0)} > \bar{\omega} \)].
  - Step 2.1: If \( u^{(n)} = u_{\text{min}} \), let \( u^* = u^{(n)} \), \( \psi^* = \psi^{(n)} \), go to Step 3; otherwise \( u^{(n+1)} = u^{(n)} - \triangle u, \psi = \psi^{(n)} \)[back up the memory curve], \( n = n + 1 \), go to Step 2.2;
  - Step 2.2: Evaluate \( \omega^{(n)} = \Gamma(u^{(n)}; \psi^{(n-1)}) \), and (at the same time) update the memory curve to \( \psi^{(n)} \). Compare \( \omega^{(n)} \) with \( \bar{\omega} \): if \( \omega^{(n)} = \bar{\omega} \), let \( u^* = u^{(n)} \), \( \psi^* = \psi^{(n)} \), go to Step 3; if \( \omega^{(n)} < \bar{\omega} \), go to Step 2.1; otherwise go to Step 2.3;
  - Step 2.3: If \( |\omega^{(n)} - \bar{\omega}| \leq |\omega^{(n-1)} - \bar{\omega}| \), let \( u^* = u^{(n)} \), \( \psi^* = \psi^{(n)} \), go to Step 3; otherwise \( u^* = u^{(n-1)} \), \( \psi^* = \bar{\psi} \)[restore the memory curve], go to Step 3;

- **Step 3**. Exit.

It’s not hard to see the above algorithm yields the best input \( u^* \) in at most \( L \) iterations. And in each iteration, the evaluation of \( \omega^{(n)} \) is very fast since the input has changed by one level and thus we need only update states of hysterons corresponding to that level. In other words, the state of each hysteron needs to be updated at most once during the whole process of finding \( u^* \). These factors combine to make this algorithm simple and efficient.

Other types of search algorithm, like bisection algorithm may also be applied to find \( u^* \). However, bisection algorithm may involve re-evaluation of states of some hysterons for many times, which makes the algorithm slow.

The closest match algorithm generates input in a discrete value set and thus apparently it is discontinuous in time. This is not true because we tacitly assume the input is changed continuously and monotonically in each iteration in evaluating the output of the discretized Preisach operator. This continuity is achieved in real system implementation by linearly interpolating between the computed values.

5. EXPERIMENTAL RESULTS

In this section, we will apply the identification and inversion schemes to open loop control of a magnetostrictive actuator. Magnetostriction is the phenomenon of strong coupling between magnetic properties and mechanical properties of some ferromagnetic materials (e.g., Terfenol-D): strains are generated in response to an applied magnetic field, while conversely, mechanical stresses in the materials produce measurable changes in magnetization. Figure 6 shows a sectional view of a Terfenol-D actuator manufactured by ETREMA Products, Inc. By varying the current in the coil, we vary the magnetic field in the Terfenol-D rod and thus control the motion of the rod head. Figure 7 displays the hysteresis in the magnetostrictive actuator.

As mentioned in Section 1, the magnetostriction \( \lambda \) is connected to the magnetization \( M \) by

\[
\lambda = a_1 M^2.
\]  \hspace{1cm} (13)

We can identify the coefficient \( a_1 \) from the saturation magnetization \( M_s \) and the saturation magnetostriction \( \lambda_s \) provided by the manufacturer:

\[
a_1 = \frac{\lambda_s}{M_s^2}.
\]
Let the length of the magnetostrictive rod be $l_{rod}$. Given a measurement of the displacement $d$, since $d = l_{rod}\lambda$, the underlying magnetization $M$ is determined by,

$$M = \pm \sqrt{\frac{d}{a_1 l_{rod}}}, \quad (14)$$

and the sign of $M$ is determined with further information on the input. The applied magnetic field $H$ is related to the input current through

$$H = NI + H_{bias}, \quad (15)$$

where $N$ is the number of coils per unit length and $H_{bias}$ is the bias magnetic field produced by permanent magnets. $H_{bias}$ is necessary for generating bidirectional strains and it can be identified easily.

We will treat the magnetic field $H$ as input and the bulk magnetization $M$ as output using transformations (14) and (15). And we employ the Preisach operator to model the hysteretic relationship between $M$ and $H$. We will identify the Preisach measure, and then carry out inverse compensation based on the identified measure. An open loop tracking experiment will be done to check the performance of the identification and inversion algorithms.

5.1. Experiment setup

Our experimental setup is as shown in Figure 8. DSpace ControlDesk is a powerful tool for real-time simulation and control. It can generate system models from Simulink of Matlab, download real-time applications into a DSP board, monitor and control a system in real time by collecting data and sending out commands. The data it collects can be displayed or be saved on disk for post-processing. The displacement of the actuator is measured with a LVDT sensor.
5.2. Measure identification and validation

The magnetic field input $H$ is limited in the range $[-40 \text{ Oe}, 480 \text{ Oe}]$ and is discretized into 26 levels. Figure 9 shows distribution of the identified measure. The constant contribution from $\Omega_0$ (see Figure 3) is estimated to be 0.3466.

To verify the identified measure, we apply same input signal (which differs from the input used for identification) to the actuator and the Preisach model. Figure 10 shows the comparison of the actuator output and the output of the Preisach model. We can see they agree reasonably well and therefore the identified Preisach operator provides a good approximation to the actuator.

5.3. Open loop tracking experiment

Finally, we do an open loop tracking experiment to test the overall performance of our identification and inversion scheme. Given a desired trajectory, whose amplitude and frequency are both varying, we compute a desired input using the closest match inversion algorithm. The computed input signal is then sent to the actuator and the displacement trajectory is measured. Figure 11 shows the comparison of the desired trajectory and the actual trajectory. As we can see, the tracking error lies within a small interval $[-3\mu m, 3\mu m]$.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a systematic approach for control of smart actuators. We model the hysteresis by the Preisach operator and identify the Preisach measure using a constrained least square method. We have presented
Figure 10. Validation of the identified measure

Figure 11. Open loop tracking result
an inversion algorithm based on the strictly increasing property of the Preisach operator. The effectiveness of our approach is demonstrated via an open-loop trajectory tracking experiment.

There are a couple of possible and interesting directions to extend this work:

- It is well known that properties of smart actuators may vary with time, temperature, etc. This means we might need to re-identify the model quite often or even on-line identification is necessary. Also off-the-shelf algorithms for solving the constrained least squares problem can be very time-consuming when the discretization gets fine. Therefore a fast and efficient identification algorithm will be very useful.

- In this paper we consider low frequency input signals and thus ignore the dynamic part $G(s)$ in the model (Figure 4). Extending of current work to accommodate wider frequency bandwidth is of practical importance.

REFERENCES