



University of Kentucky
UKnowledge

University of Kentucky Master's Theses

Graduate School

2005

MODELING AND CONTROL OF MAGNETOSTRICTIVE ACTUATORS

Wei Zhang
University of Kentucky

[Right click to open a feedback form in a new tab to let us know how this document benefits you.](#)

Recommended Citation

Zhang, Wei, "MODELING AND CONTROL OF MAGNETOSTRICTIVE ACTUATORS" (2005). *University of Kentucky Master's Theses*. 257.
https://uknowledge.uky.edu/gradschool_theses/257

This Thesis is brought to you for free and open access by the Graduate School at UKnowledge. It has been accepted for inclusion in University of Kentucky Master's Theses by an authorized administrator of UKnowledge. For more information, please contact UKnowledge@lsv.uky.edu.

ABSTRACT OF THESIS

Wei Zhang

The Graduate School
University of Kentucky

2005

MODELING AND CONTROL OF MAGNETOSTRICTIVE ACTUATORS

ABSTRACT OF THESIS

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science in the
College of Engineering
at the University of Kentucky

By
Wei Zhang

Lexington, Kentucky

Director: Dr. YuMing Zhang, Professor of Electrical Engineering

Lexington, Kentucky

2005

Copyright © Wei Zhang, 2005

ABSTRACT OF THESIS

MODELING AND CONTROL OF MAGNETOSTRICTIVE ACTUATORS

Most smart actuators exhibit rate-dependant hysteresis when the working frequency is higher than 5Hz. Although the Preisach model has been a very powerful tool to model the static hysteresis, it cannot be directly used to model the dynamic hysteresis. Some researchers have proposed various generalizations of the Preisach operator to model the rate-dependant hysteresis, however, most of them are application-dependant and only valid for low frequency range. In this thesis, a first-order dynamic relay operator is proposed. It is then used to build a novel dynamic Preisach model. It can be used to model general dynamic hysteresis and is valid for a large frequency range. Real experiment data of magnetostrictive actuator is used to test the proposed model. Experiments have shown that the proposed model can predict all the static major and minor loops very well and at the same time give an accurate prediction for the dynamic hysteresis loops.

The controller design using the proposed model is also studied. An inversion algorithm is developed and a PID controller with inverse hysteresis compensation is proposed and tested through simulations. The results show that the PID controller with inverse compensation is good at regulating control; its tracking performance is really limited (average error is 10 micron), especially for high frequency signals. Hence, a simplified predictive control scheme is developed to improve the tracking performance. It is proved through experiments that the proposed predictive controller can reduce the average tracking error to 2 micron while preserve a good regulating performance.

KEYWORDS: Dynamic hysteresis, Preisach, Magnetostriction, Smart Actuators,
Modeling and Control

MODELING AND CONTROL OF
MAGNETOSTRICTIVE ACTUATORS

By
Wei Zhang

Director of Dissertation

Director of Graduate Studies

RULES FOR THE USE OF THESES

Unpublished dissertations submitted for the Master's degree and deposited in the University of Kentucky Library are as a rule open for inspection, but are to be used only with due regard to the rights of the authors. Bibliographical references may be noted, but quotations or summaries of parts may be published only with the permission of the author, and with the usual scholarly acknowledgements.

Extensive copying or publication of the dissertation in whole or in part also requires the consent of the Dean of the Graduate School of the University of Kentucky.

A library that borrows this dissertation for use by its patrons is expected to secure the signature of each user.

Name

Date

THESIS

Wei Zhang

The Graduate School
University of Kentucky

2005

MODELING AND CONTROL OF
MAGNETOSTRICTIVE ACTUATORS

THESIS

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science in the
College of Engineering
at the University of Kentucky

By
Wei Zhang

Lexington, Kentucky

Director: Dr. YuMing Zhang, Professor of Electrical Engineering

Lexington, Kentucky

2005

Copyright © Wei zhang, 2005

MASTER'S THESIS RELEASE

I authorize the University of Kentucky
Libraries to reproduce this thesis in
whole or in part for purposes of research.

Signed: _____

Date: _____

ACKNOWLEDGEMENTS

This research is funded by the Energen Inc company and the University of Kentucky Center for Manufacturing. I would like to thank Dr. YuMing Zhang, and Dr. Bruce L. Walcott for their guidance, encouragement and support. I would also like to thank Wei Lu and Yu Chi for their helpful advice throughout my research. In addition, I want to thank my parents for their love and support and lastly my wife, Rong Zhang, for being the most wonderful person in my life.

TABLE OF CONTENT

ACKNOWLEDGEMENTS	III
------------------------	-----

TABLE OF CONTENT.....	IV
-----------------------	----

TABLE OF FIGURES.....	VII
-----------------------	-----

CHAPTER 1 INTRODUCTION.....	1
-----------------------------	---

1.1 BACKGROUNDS	1
-----------------------	---

1.2 STATIC HYSTERESIS MODELING.....	4
-------------------------------------	---

1.3 DYNAMIC HYSTERESIS MODELING	8
---------------------------------------	---

1.4 CONTRIBUTIONS.....	10
------------------------	----

1.5 OUTLINE OF DISSERTATION	11
-----------------------------------	----

CHAPTER 2 BACKGROUND	13
----------------------------	----

2.1 MAGNETOSTRICTION.....	13
---------------------------	----

2.2 GENERAL HYSTERESIS PHENOMENON.....	15
--	----

2.3. CLASSICAL PREISACH MODEL.....	17
------------------------------------	----

2.3.1 Introduction.....	18
-------------------------	----

2.3.2 Definition	19
------------------------	----

2.3.3 Geometric Interpretation	21
--------------------------------------	----

2.3.4 Discrete Preisach Model.....	27
------------------------------------	----

CHAPTER 3 A NOVEL DYNAMIC PREISACH MODEL WITH DYNAMIC RELAY	29
--	----

3.1 FIRST-ORDER DYNAMIC RELAY (FDR)	29
---	----

3.2 DYNAMIC PREISACH MODEL WITH FDR	33
3.3 DISCUSSION	33
3.4 MODEL VERIFICATION	35
3.4.1 Interpretation of dynamic hysteresis using the new model	35
3.4.2 Simulation Results.....	36
3.5 IDENTIFICATION METHODS	40
3.5.1 System Identification	40
3.5.2 Weights Identification	47
3.5.3 Time constants Identification	49
CHAPTER 4 MODELING EXPERIMENT RESULTS.....	51
4.1 SYSTEM SETUP	51
4.2 EXPERIMENTS DESCRIPTION	53
4.2.1 Low Frequency Experiments.....	53
4.1.2 HIGH FREQUENCY EXPERIMENTS.....	55
4.2 WEIGHTS IDENTIFICATION.....	56
4.2.1 Preprocess the data.....	56
4.2.2 Data selection for the identification.....	57
4.2.3 Identification.....	58
4.3 TIME CONSTANTS IDENTIFICATION	65
4.4 SENSITIVITY OF TIME CONSTANTS	70
CHAPTER 5 CONTROL OF HYSTERETIC SYSTEM	73
5.1 OBJECTIVES AND PREPARATION	73
5.2 INVERSE CONTROL	76
5.3 PID CONTROL WITH INVERSE HYSTERESIS COMPENSATION	85

5.4 PREDICTIVE CONTROL	90
5.4.1 Nonlinear Predictive Control [46,47,48].....	90
5.4.2 Proposed Predictive Control	93
CHAPTER 6 CONCLUSIONS	99
APPENDIX:.....	101
I PROGRAM MANUAL.....	101
II CODES	103
REFERENCE:.....	117
VITA.....	123

TABLE OF FIGURES

Figure 1-1: Magnetostrictive Actuator and Test Apparatus	1
Figure 1-2: Hysteresis Observed in Magnetostrictive Actuators	2
Figure 1-3: Inverse Hysteresis Compensation	3
Figure 1-4: Non-ideal Relay Operator	6
Figure 2-1: Domain Walls and Their Movements [49]	14
Figure 2-2: Elongation Principle [49]	14
Figure 2-3: Typical Hysteresis Input-output Diagram	15
Figure 2-4: Hysteresis Transducer	16
Figure 2-5: Hysteresis branching principle	17
Figure 2-6: Nonideal Relay Hysteron	19
Figure 2-7: Parallel Relay Connection: (a) Connection. (b) Response	20
Figure 2-8: Limiting Triangular in $\alpha - \beta$ plane	22
Figure 2-9 Geometric Interpretation of the Preisach Model	23
Figure 2-10: Discretization of $\alpha - \beta$ Plane	27
Figure 3-1 First-order Dynamic Relay	30
Figure 3-2: Hysteretic Loops of Magnetostrictive Actuators under Different Frequencies [2]	37
Figure 3-3: Response of single FDR under different frequencies	38
Figure 4-1 Experiment System setup (Cross-section view)	51
Figure 4-2: Experiment System Setup (Angle view)	51
Figure 4-3: Uneven Discretization scheme	53

Figure 4-4: Smoothing the experiment data.....	57
Figure 4-5: Sampled data for identification	59
Figure 4-6: Identified weights.....	60
Figure 4-7: Model Performance for 1Hz identification data	60
Figure 4-8: Model Performance for the test data (Output).	61
Figure 4-9: Model Performance for the test data (Input-Output Diagram)	62
Figure 4-10: Classical Preisach Model Performance for Dynamic Hysteresis data	63
Figure 4-11: Classical Preisach Model Performance for Dynamic Hysteresis data	64
Figure 4-12: Dynamic Modeling Results for the identification data	66
Figure 4-13: Dynamic Modeling result for identification data.....	67
Figure 4-14: Dynamic Modeling Results for test data.....	68
Figure 4-15: Dynamic Modeling Results for test data.....	69
Figure 4-16: Model Performance under Different Time Constants Variations	71
Figure 4-17: Hysteresis Loops under Different Time Constant Variations.....	72
Figure 5-1: Dynamic hysteresis loops of Virtual Actuator (Model A).....	77
Figure 5-2: Static Modeling Results for Virtual Actuator (Identification Data)	78
Figure 5-3: Dynamic Modeling Results for Virtual Actuator (Identification Data).....	78
Figure 5-4: Dynamic Modeling Results for Virtual Actuator (Test Data).....	79
Figure 5-5: Dynamic Modeling Results for Virtual Actuator (Test Data).....	80
Figure 5-6: Settling Time Estimation of Virtual Actuator (Model A).....	83
Figure 5-7: Inverse Control System.....	84
Figure 5-8: Displacement Control with Inverse Controller Only	84
Figure 5-9: PID+Inversion Control Scheme	85

Figure 5-10: Displacement Control Results with PID+inverse Controller	87
Figure 5-11: PID+Inverse Control with Noisy Measurement	87
Figure 5-12: Displacement Control with PID+Inverse Controller and Noisy Measurement	88
Figure 5-13: 1 Hz Tracking Result with PID+Inverse Controller and Noisy Measurement	88
Figure 5-14: Tracking Results with PID+Inverse Controller (50Hz)	89
Figure 5-15: Displacement Control with Predictive Controller.....	96
Figure 5-16: Tracking Control with Predictive Controller	97

CHAPTER 1

INTRODUCTION

1.1 Backgrounds

Smart materials, whose characteristics may alter due to the change of environments or exogenous inputs, are being more and more employed in measurement and control systems. Magnetostrictive materials, a type of the most widely used smart materials, are good at providing giant forces, strains and high energy densities, and thus are very promising in noise or vibration control, especially for heavy structures. They rely on the magnetostrictive effect, which is inherent to ferromagnetic materials such as nickel and Terfenol-D, to achieve high performance as actuators or sensors.

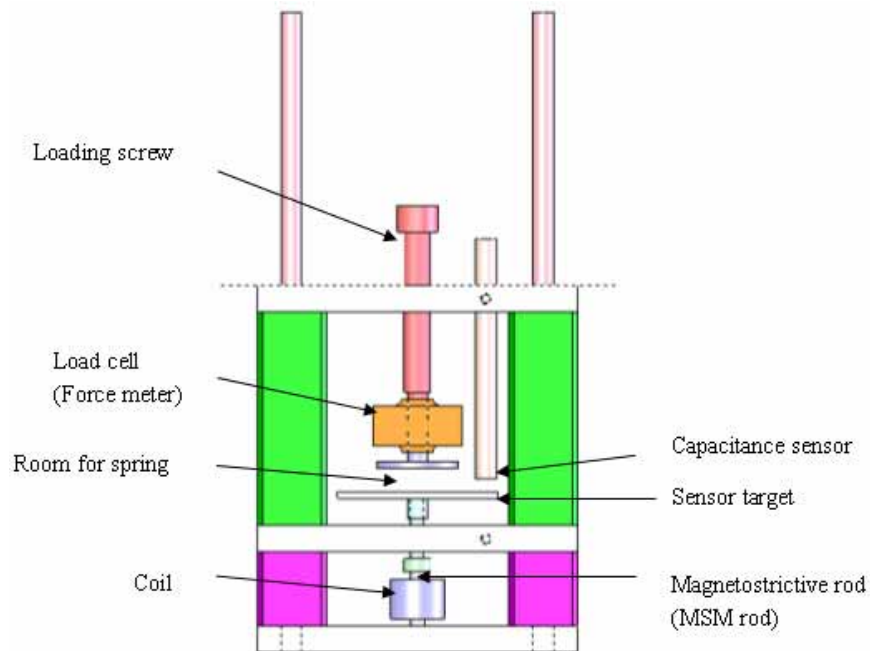


Figure 1-1: Magnetostrictive Actuator and Test Apparatus

Figure 1-1 is the actuator under study in this project installed in the test apparatus used to obtain the experimental data cited in this work. When the input current is given, the coil generates a magnetic field along the central axis and causes an elongation (displacement) in the MSM rod. The expanded MSM rod pushes the plunger and the target up to change the gap between the eddy current sensor and the target. This movement causes a voltage change of the sensor and can be converted to displacement data.

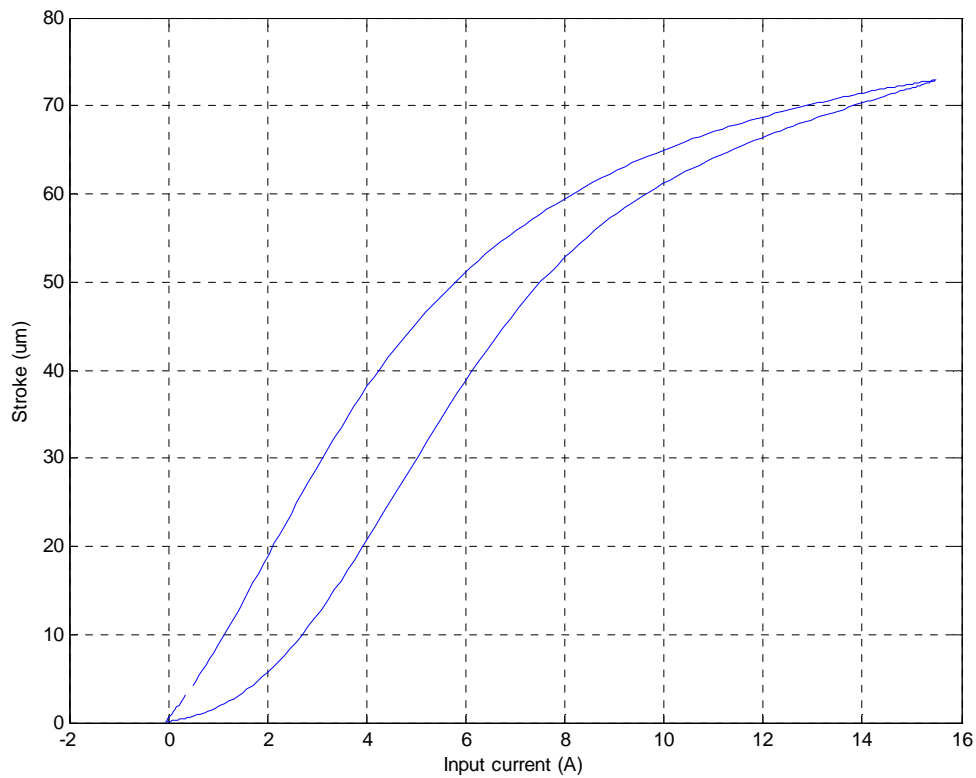


Figure 1-2: Hysteresis Observed in Magnetostrictive Actuators

By adjusting the input current, the actuator is able to provide forces as well as accurate displacements. However, the strong hysteresis behavior between the input

current and output displacement as shown in Fig 1-2 makes the actuator really difficult to control. This hysteresis is believed to be caused primarily by the hysteresis between the magnetic field H and magnetization M , which is inherent in ferromagnetic materials. In fact, magnetostrictive actuators exhibit significant hysteretic nonlinearities to a degree that other smart materials, such as electrostrictive and piezoelectric, do not. Hence, the strong hysteresis nonlinearity becomes the major obstacle to further applications of the magnetostrictive actuators.

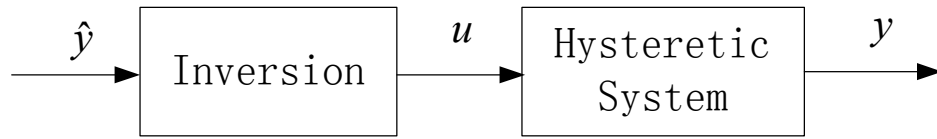


Figure 1-3: Inverse Hysteresis Compensation

A common and easy way to deal with actuator hysteresis is to use the inverse compensation [2, 10] as demonstrated in Fig. 1-3 where the inversion of a certain hysteretic model of interest is used to compute an appropriate input to the actuator. That is, to produce an expected output \hat{y} , the inverse model is used to calculate an input u and the calculated input u is applied to the actuator. Of course, the produced output y may not be exactly the same as \hat{y} . The accuracy of this method depends on the accuracy of the hysteresis model and is sensitive to the noise. For this reason, various advanced control algorithms [37~42] are employed to improve the actuator performance. These methods use feedback information to adjust the actuator input in order to accommodate the model uncertainty and the noise disturbance.

Although the advanced control algorithm can usually achieve a better performance than the inverse compensation, this performance is inevitably limited by the accuracy of the actuator model employed in the controller. Hence, a better actuator model is the key problem for the effective use of magnetostrictive actuators.

Usually, the actuator model is constructed in two steps. First, a hysteretic model is used to model the hysteresis between the magnetic field H and magnetization M . Then another model (usually a linear dynamic model) is employed to capture various dynamics of rod in the actuator. In some applications, it is not necessary to know what really happens inside the actuator, so a direct input-output model is sufficient and more desirable, especially for control applications. For example, in [25] a Preisach-based dynamic hysteresis model is used to directly model the voltage-to-displacement dynamics in piezoceramic actuators. The model is very promising in controller design using piezoceramic actuators. No matter the actuator is treated as a whole or as several cascaded parts, the hysteresis model is the key part that usually determines the overall performance of the entire actuator model.

1.2 Static Hysteresis Modeling

Typically, hysteresis models are classified into physics-based models and phenomenon based models.

The Jiles-Atherton model of ferromagnetic hysteresis [16] is one of the most well known physics-based hysteresis models. It is a quantitative model that is based on a macromagnetic formulation. The model describes isotropic polycrystalline materials

with domain wall motion as the major magnetization process. Five physical parameters are used to describe magnetization curves, which are:

M_s → the spontaneous magnetization

k → pinning coefficient

α → interactions between domains

a → thermal aspect (domain wall density)

c → reversible magnetization component

A fitting procedure can then be easily proposed to enable the user to determine values for each of the parameters above. These are related to measurable characteristics of the material, specifically the differential initial susceptibility, the coercivity, the slope at the coercive field and anhysteretic susceptibility.

Since physics-based hysteresis models are usually derived rigorously from some basic physics assumptions, they seem more reasonable and convictive. However, most of them require substantial physics knowledge and are specific to particular system, so they are not as common as phenomenon-based models, especially in the area other than material science and physics, such as mathematics, mechanical and electrical engineering. In contrary, phenomenon-based models do not provide insight into the behavior of the material, therefore they cannot be used to obtain new physical insight. However, they are application independent and can describe or predict the behavior of a consistent and well-controlled material very well without requiring too much background in material science. Hence phenomenon-based hysteresis models are widely used in modeling and control of hysteretic systems.

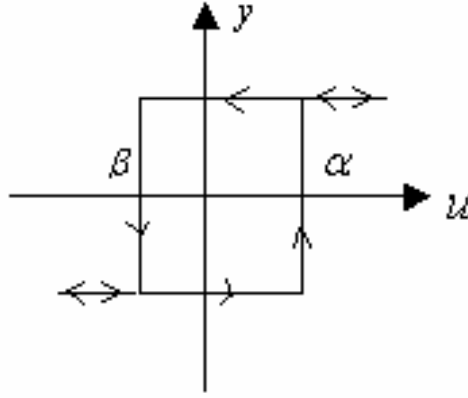


Figure 1-4: Non-ideal Relay Operator

The classical Preisach model [1,2,6,8,9] is the most widely used phenomenon-based model for hysteresis. It models hysteresis as the weighted sum of an infinite set of relays (Fig 1-4). Each relay in the model can be uniquely represented by its ‘up’ and ‘down’ switching thresholds α and β . Given the weight function of the relays $\mu(\alpha, \beta)$, the output of the model can be mathematically calculated by an integral of $\mu(\alpha, \beta) \gamma_{\alpha\beta}(u(t), \xi(t))$ over the set E of all the possible thresholds pairs $E = \{(\alpha, \beta) \in R^2 : \beta \leq \alpha\}$, where $\gamma_{\alpha\beta}$ is the relay operator, $\xi(t)$ is the state of the relay (‘on’ or ‘off’). The detailed exposition of the Preisach model is given in next chapter.

Since the classical Preisach model is application-independent, does not require substantial physics background and is capable to predict the hysteresis behavior very well, it has become the focus of research for a long time and is regarded as the most popular tool in modeling various static hysteretic systems.

The classical Preisach model is just the linear combination of the elementary nonlinearities—non-ideal relay, which is also called the kernel of the operator. To generalize this idea, the Krasnoselskii-Pokrovskii (KP) operator [36] allows the kernels to be any reasonable functions. This generalization finally separates the Preisach model from its physics meaning and ends up with a purely mathematical and phenomenological operator. This generalized Preisach model has been further studied and applied in [10,11], where kernels other than non-ideal relay operators are employed to achieve some mathematical properties.

There are also other phenomenon-based models being used in the literature. In [7, 34], J. Takacs proposed a purely mathematical model of hysteresis that takes advantage of the fundamental similarities between the Langevin function (the specified $T(x)$ function) and the sigmoid shape to operationally describe the hysteresis loops. It describes not only the regular hysteresis loops but also the biased and other minor loops like the ones produced by the interrupted and reversed magnetization process and the open loops created by a piecewise monotonic magnetizing field input of diminishing amplitude. Although it is also a phenomenon-based model as the classical Preisach model, it is purely operational and is not based on any physics principles. While this model often provides accurate model fits with a small number of required parameters, its capabilities for general applications involving symmetric and asymmetric minor loops appears limited [35].

One of the major advantages of the phenomenon-based model is its flexibility for practical applications, where controller design takes priority over physical accuracy. In some applications, the actuator can be assumed to work in some region where its

hysteresis is not that significant. Then some simplified hysteretic model could be used to ease the modeling and controller design process. This idea is frequently used during adaptive controller design, where fast inverse algorithm is performed online to update the model parameters. In [38, 39, 40], a piece-wise linear model is used to approximate the unknown simple hysteresis in the actuators and adaptive algorithms are developed correspondingly. Although this kind of model describes the hysteresis loops in a rough manner, it is good enough for many applications and can make the online adaptive controller design much easier.

1.3 Dynamic Hysteresis Modeling

The above physics-based and phenomenon-based models or their variations can predict the static hysteresis behavior very well. However, they are all rate-independent, thus can not be directly used to model the dynamic hysteresis. Since most applications of smart actuators are not static, effective dynamic hysteresis models are in great need.

The simplest and most straightforward way for dynamic hysteresis modeling is to assume the dynamic hysteresis loop for each frequency of interest as a static loop of a new hypothetical material which is free of dynamic losses. Then use the static hysteresis model to fit the loops to obtain a frequency-by-frequency dynamic loop description [this is not a complete sentence.]. This scheme is proposed and applied in [23]. Obviously, this kind of method requires the working frequencies be selected from several discrete values and known ahead of time, which is quite impractical and limits its applications.

Another simple idea of dynamic modeling is to configure a model primarily based on close examination of the hysteresis curve, obtained from laboratory tests. For example, in [28], Menemenlis developed an operational model step-by-step based on the major and minor hysteresis loops observed in a transformer. In [32], Carpenter proposed a simple, ad hoc, method to broaden a static hysteresis loop and then configure this model that changes the amount of broadening with frequency as needed to agree with observations. These methods are appropriate in some applications. However, they are purely operational, which can not give too much inspiration and directions to other applications.

The above methods are all operational and ad hoc in nature. To formalize the idea, an application-independent method is needed to systematically describe the dynamic hysteresis behavior. Such examples can be found in [20, 21,22], where a neural network is trained to map the frequency and magnitude of a sinusoidal signal to the Fourier coefficients of the corresponding output. Then given any kind of input, its ‘actual frequency’ within a short time-window is estimated and its corresponding output can be obtained through the coefficients computed by the neural network. This method does not require the knowledge of the physical properties of the material and the geometrical effects of the nucleus (skin depth effect, nature of lamination, etc). However, it requires the pre-processing of the experimental input data by Fourier series and the reconstruction of the output data in the inverse way.

In fact, the majority of the dynamic hysteresis models are built on a static hysteresis model. For example, the dynamic model of Jules [15] is based on his static hysteresis model [16], while Hodgdon’s dynamic model [17] is based on the static model of Coleman and Hodgdon [18], and the dynamic piece-wise linear circuit model of

Cincotti [26] is developed from its static counterpart [27]. All these dynamic hysteresis models inevitably depend on the performance of their corresponding static hysteresis models. For this reason the dynamic generalizations of the classical Preisach model are more attractive. There are many dynamic models [2, 12, 19, 24, 25] that are based on the classical Preisach model. To make the classical Preisach model rate-dependent, Mayergoyz [1,19] introduced the dependence of weight function on the speed of output variations; similarly, Mrad and Hu [25, 24] proposed a input-rates dependence of the weight function. Both methods suppose the weight function is the right place to add in dynamic behaviors. However, in [33] a linear dynamic model is added before the classical Preisach operator and the dynamics are assumed to only happen inside the linear dynamic part. This kind of cascade structure is referred as ‘external dynamic hysteresis model’ in [14]. This structure is modified in [2] and [9], where the classical Preisach operator is coupled to an ODE (ordinary differential equation), which can not be simply decomposed as a cascade of the Preisach operator with a linear system.

1.4 Contributions

All of the above modification of the classical Preisach model can fit dynamic hysteresis loops. However, their physical significance and motivation are really complex. The hysteron is believed to be the fundamental reason of the entire hysteresis. And the Preisach model is just the superposition of all these hysterons. This concept is what makes the Preisach model successful. So there is no reason to add the dynamic term into the weight function or to a separate dynamic system. A straightforward way is to make

the elementary hysteresis operator—hysteron rate-dependent. In this way the entire Preisach model will be inherently dynamic, and its structure, which is believed to effectively reflect the physical nature of hysteresis, is kept unchanged.

Thus in this project, the idea of adding dynamic terms into the relay operator of the Preisach model is proposed and studied. A new Preisach-type dynamic hysteresis model is developed. Identification methods of the new model are designed and analyzed. Experiments on a real magnetostrictive actuator are performed to test the proposed model. It is shown that the new dynamic hysteresis model is not only theoretically plausible, but also exceptionally good at modeling practical dynamic hysteretic systems.

1.5 Outline of dissertation

This dissertation is organized into six chapters.

Chapter 2 introduces the background knowledge that is necessary for understanding the rest of the dissertation. It starts with the explanation of the magnetostriction phenomenon. After that the definition of the general hysteresis phenomenon is provided. Last, the classical Preisach model, based on which the thesis is developed, is reviewed and discussed in great details.

Chapter 3 is about the proposed novel dynamic hysteresis model. The motivations, objectives of introducing the new model are discussed. The formal mathematical definition of the model is given. The properties and features of the new model is pointed out and shown in some preliminary simulations. Basic system identification theory is

reviewed. An identification algorithm based on the particular form of the new model is developed and discussed.

Chapter 4 talks about the procedure of modeling a real magnetostrictive actuator using the proposed dynamic hysteresis model. The experiment design and modeling process are both described. The modeling results are shown. The performance of the classical Preisach model using the same experiment data is also given for comparison.

Chapter 5 summarizes the whole dissertation. Conclusions are made. Future research work is suggested.

CHAPTER 2

BACKGROUND

2.1 Magnetostriction

Magnetostriction is the changing of a material's physical dimensions in response to a change in applied magnetic field. All ferromagnetic materials exhibit some measurable magnetostriction, although some rare-earth intermetallics such as Terfenol-D exhibit up to several tenths of a percent elongation. The mechanism of magnetostriction at an atomic level is relatively complex subject matter but on a microscopic level may be explained by the domain wall theorem. A ferromagnetic material is theoretically believed to be composed of many small regions called “domains” (Fig 2.1). Each domain is spontaneously magnetized. However, the whole sample might not appear magnetic if the magnetization of each domain is aligned differently ($\mathbf{M} = 0$). When an external magnetic field \mathbf{H} is applied, the domain walls start moving and each magnetization vector rotates toward the applied field direction. These processes cause dimensional changes of the material called magnetostriction. Increasing the field causes an increase in magnetostriction until saturation is achieved.

When a compressive force is applied to a magnetostrictive material, it “squeezes” the domains perpendicular to the elongation direction (Fig 2.2) [49]. When an external magnetic field \mathbf{H} is applied, all of the domains rotate 90° to align with the field, which enables maximum elongation of the material. Therefore, compressive force is a key factor in magnetostrictor applications.

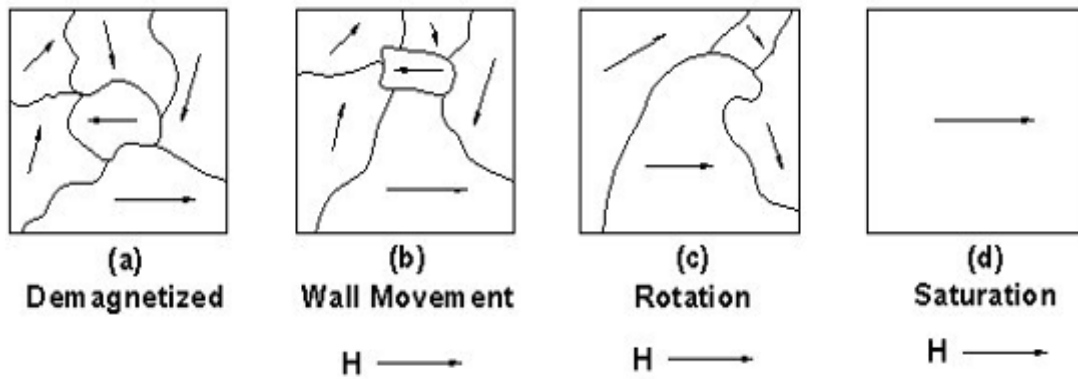


Figure 2-1: Domain Walls and Their Movements [49]

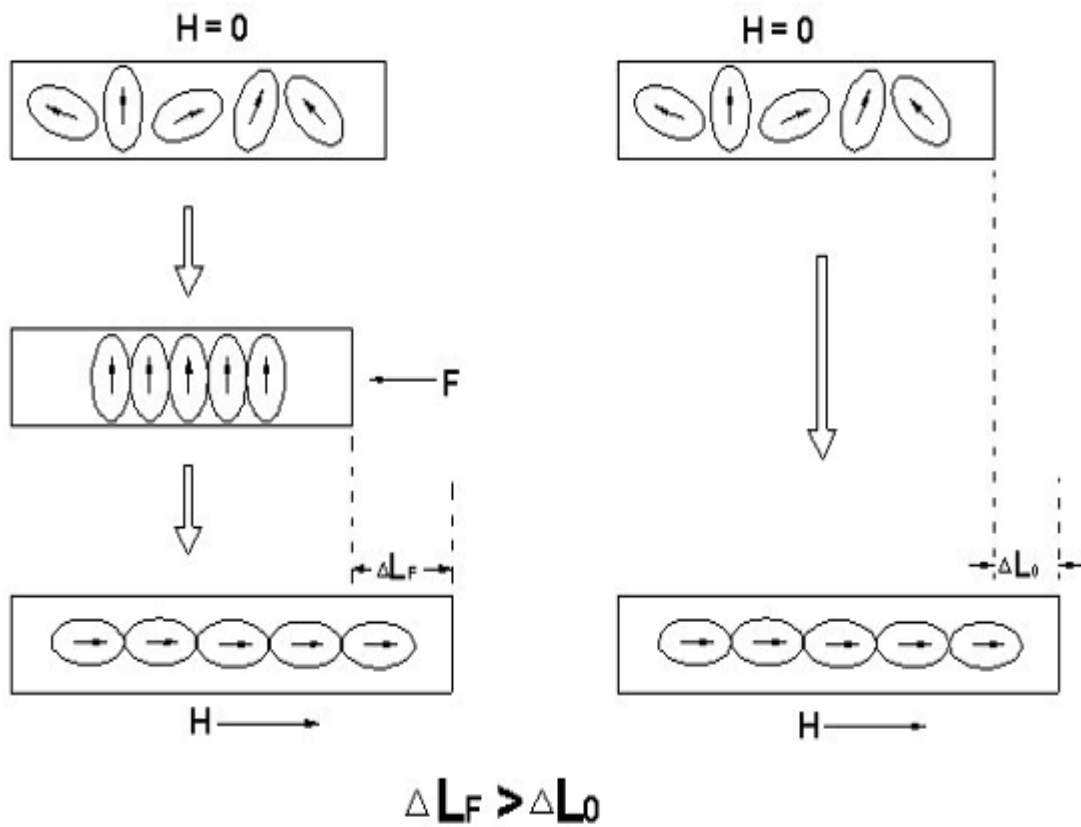


Figure 2-2: Elongation Principle [49]

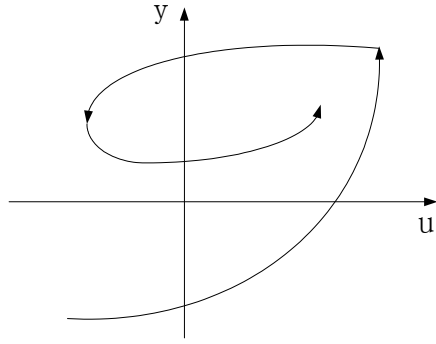


Figure 2-3: Typical Hysteresis Input-output Diagram

2.2 General Hysteresis Phenomenon

The word hysteresis comes from Greece and means etymologically ‘coming behind’. Hysteresis is a strongly nonlinear phenomenon that occurs in many industrial, physical and economic systems.

The phenomenon of hysteresis has been with us for ages and has been attracting the attention of many researchers for a long time. The reason is that hysteresis is ubiquitous. People in different fields may talk about different hysteresis, for example, magnetic hysteresis, ferroelectric hysteresis, mechanical hysteresis, superconducting hysteresis, adsorption hysteresis, optical hysteresis, electron beam hysteresis, etc.

A typical hysteresis system has an input-output diagram as shown in Fig 2-3. The quantities u , y , measured along the horizontal and the vertical axis respectively, can have different physical meanings, such as deformations versus force (plastic hysteresis) or external magnetic field versus magnetization (ferromagnetic hysteresis).

In the literature ([1], [2]), hysteresis is described through a transducer that is characterized by an input $u(t)$ and output $y(t)$ as in Fig 2-4.

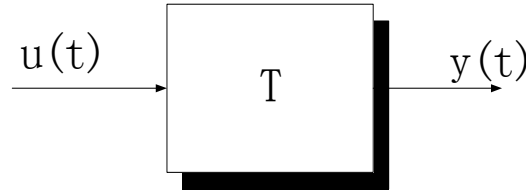


Figure 2-4: Hysteresis Transducer

The transducer T is called a hysteresis transducer whose input-output relationship is a multibranch nonlinearity for which branch-to-branch transitions occur at input extremes. This statement is illustrated in Fig 2-5. From O to A the input $u(t)$ is rising, and the output $y(t)$ is also increasing; from A to B , input is decreasing, the output does not come back along AO , but makes a new branch from A to B . This kind of branching constitutes the essence of hysteresis and makes the hysteresis transducer a very complicated operator.

We can easily see that T is not a function because for the same input value $u(t_0)$, different output values $y(t_0)$ can be observed (graphically some vertical lines may have multiple intersections with the input-output curve in Figure 2-3). In other words, an output $y(t)$, after a certain reference time t_0 , depends not only on the input $u(t)$, $t > t_0$, but also on an internal/initial state x_0 of the transducer T . In this sense the hysteresis transducer is a system with memory and the memory could be infinite. So a hysteretic system is a very complicated nonlinear system, whose behavior is really difficult to predict.

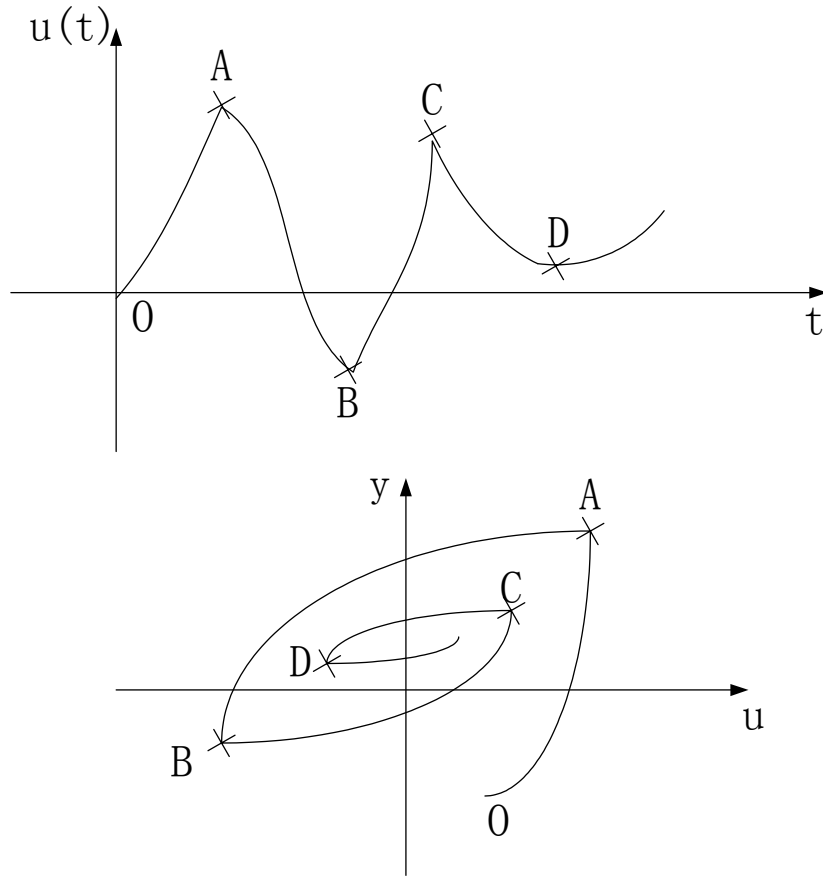


Figure 2-5: Hysteresis branching principle

2.3. Classical Preisach Model

The Preisach model is one of the most remarkable contributions to rate-independent hysteresis modeling [1]. Although it cannot give much insight into the physical nature, in the knowledge of the hysteresis data it can produce similar behaviors to those of real hysteretic physical systems and give reasonable predictions, which are necessary for control applications.

2.3.1 Introduction

The origin of the Preisach model can be traced back to the landmark paper of F. Preisach published in 1935. Preisach's approach was purely intuitive. It was based on some plausible hypothesis concerning the physical mechanisms of magnetization, and thus was regarded as a physical model of hysteresis at the beginning. Because of its effectiveness and simplicity, the Preisach model has become the most popular tool in hysteresis modeling and considerable research has been done in this field. The most decisive step in the direction of better understanding of the model was made in the 1970s by Russian mathematician M. Krasnoselskii who realized that the Preisach model contained a new general mathematical idea. Krasnoselskii separated this model from its physical meaning and represented it in a purely mathematical form which is similar to a spectral decomposition of operators. As a result, a new mathematical tool has evolved which can now be used for the mathematical description of hysteresis of different physical nature. The new methodology that Krasnoselskii proposed is generally the following:

1. Choosing elementary hysteresis nonlinearities, so called *hysterons* (such as nonideal relay, generalized play, Prandtl or Duhem models, etc).
2. Treating complex hysteresis nonlinearities as *block-diagrams* of hysterons.
3. Establishing *identification principles*.

Nowadays this approach to hysteresis is standard and it contains a wide variety of 'branches', depending on the choice of hysterons in item 1 and/or the basic type of the block-diagrams in item 2.

2.3.2 Definition

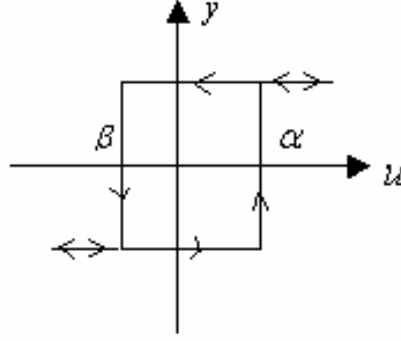


Figure 2-6: Nonideal Relay Hysteron

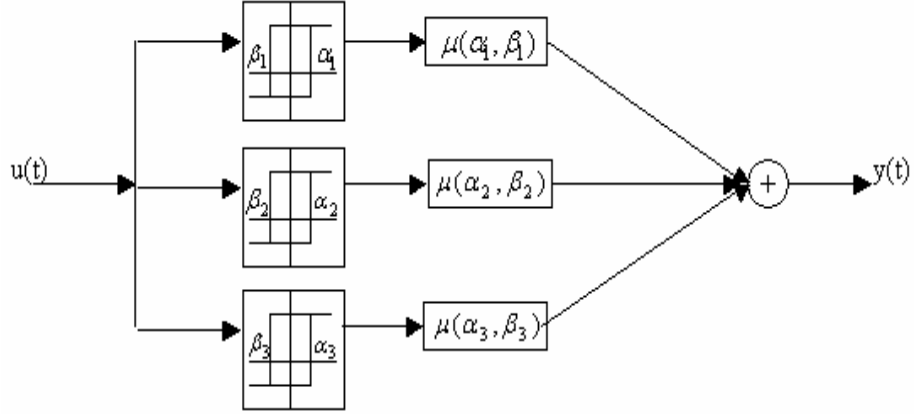
Among hysterons, the most important are probably the *nonideal relay nonlinearities* (Fig 2-6), or, as they are also called, the *thermostat nonlinearities*. It is denoted by $\gamma_{\alpha\beta}(u(t), \xi(t))$, where $u(t)$ is the input and $\xi(t) = +1$ or -1 is the state of the relay.

Its output can take one of two values -1 or 1 , which means that at any moment the relay is either 'switched off' or 'switched on'. It is mathematically defined as:

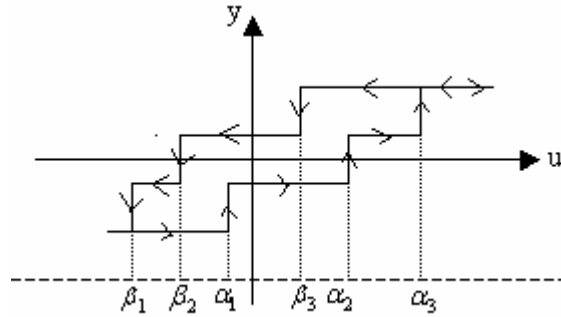
$$w(t) = \gamma_{\alpha\beta}(u(t), \xi(t)) = \begin{cases} -1 & \text{if } u(t) < \beta \\ 1 & \text{if } u(t) > \alpha \\ w(t^-) & \text{if } \beta \leq u(t) \leq \alpha \end{cases} \quad (2-1)$$

where $w(t^-) = -1$ or 1 , and $t^- = \lim_{\varepsilon \rightarrow 0, \varepsilon > 0} t - \varepsilon$.

The simplest block-diagrams are essentially those of standard parallel connections of a number of hysterons, or their continuous analogue. This kind of connections of the nonideal relay hysterons is a realization of the Krasnoselskii's concept, which leads to the so called Preisach operator.



(a)



(b)

Figure 2-7: Parallel Relay Connection: (a) Connection. (b) Response

It is believed that ferromagnets are composed of a large number of elementary magnets (domains), each of which behaves like a relay. Thus the overall response of a hysteresis system is just the weighted superposition of those relevant relays. To understand this idea, let's consider the weighted parallel connection of three hysterons as shown in Fig 2-7 (a). The output of this system can be written as:

$$y(t) = \sum_{n=1}^3 \mu(\alpha_n, \beta_n) \cdot \gamma_n(u(t)) \quad (2.2)$$

We can see its corresponding input-output diagram (Fig 2-7 (b)) is more like a real hysteretic loop than a single relay. If we add more relays with different thresholds, the

response loop will become smoother and easier to fit different kind of shapes by adjusting the weights $\mu(\alpha, \beta)$.

Hence, to generalize this idea, the Preisach model sums the weighted response of an infinite set of relays $\gamma_{\alpha\beta}$ over all possible switching thresholds $\alpha \geq \beta$:

$$y(t) = \iint_{\alpha \geq \beta} \mu(\alpha, \beta) \gamma_{\alpha\beta}(u(t)) d\beta d\alpha \quad (2.3)$$

Although the original Preisach model was a physical model, it is generalized by Equation (2.3) that has a purely mathematical nature [like the entire field of differential equations!]. This definition of the Preisach model reveals its mathematical and phenomenological nature, broaden the area of applicability of this model to the field other than magnetics. For this reason, the purely mathematical definition (2.3) is more attractive and has been widely used by many researchers working in various fields.

2.3.3 Geometric Interpretation

The mathematical investigation of the Preisach model is considerably facilitated by its geometric interpretation. This interpretation is based on the following simple fact. There is a one-to-one correspondence between the operator $\gamma_{\alpha\beta}$ and the real number pair (α, β) . So each relay $\gamma_{\alpha\beta}$ is uniquely determined by its thresholds (α, β) . In other words, each point in the half $\alpha - \beta$ plane $\alpha \geq \beta$ represents a relay. If we think the thresholds have a limited range $\beta_0 \leq \beta \leq \alpha \leq \alpha_0$, then relevant relays constitutes a triangular T like in Fig 2-8. Its hypotenuse is a part of the line $\alpha = \beta$, while the vertex of its right angle has the coordinates α_0 and β_0 . This triangular T is called the limiting triangular. If we only consider the relays inside the triangular, then the weighting

measure $\mu(\alpha, \beta)$ is assumed to be finite inside T and zero outside T . This limiting triangular and the assumption will ease our discussion and will not limit the applicability of the Preisach model.

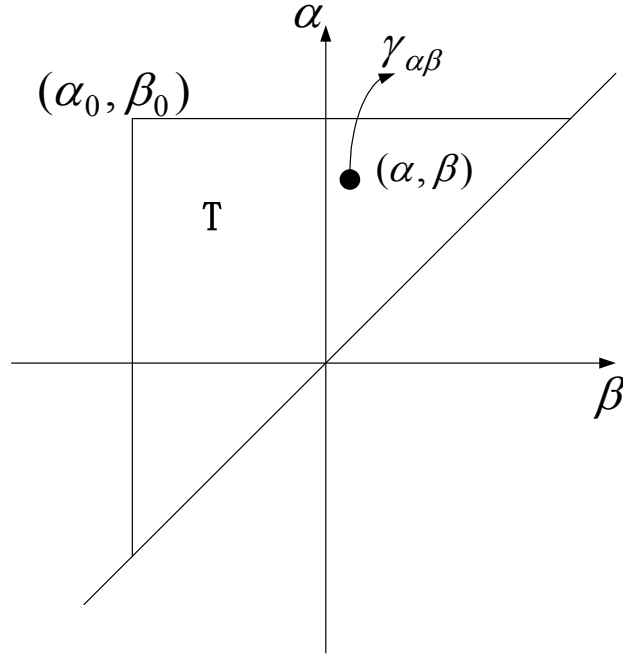


Figure 2-8: Limiting Triangular in $\alpha - \beta$ plane

To start the discussion, we first assume that the input $u(t)$ at some instant of time t_0 has the value which is less than β_0 . Then all the relays are turned off which means the outputs of all the relay operator $\gamma_{\alpha\beta}$ are -1. This corresponds to the state of “negative saturation”.

Now we assume that the input is monotonically increasing until it reaches u_1 at time t_1 . As the inputs are increased, all the relays with the ‘up’ switching value α less than the current input $u(t)$ are turned ‘on’, which means their outputs become equal to +1. Geometrically, it divides the limiting triangular into two sets: $S^+(t)$ consisting of

points (α, β) whose corresponding relays are in ‘on’ states, and $S^-(t)$ whose corresponding relays are still in ‘off’ states. The two sets are separated by the line $\alpha = u(t)$, which moves upwards as the input is being increased. This upward motion is terminated when the input reaches the maximum value u_1 (Fig 2-9 (a)).

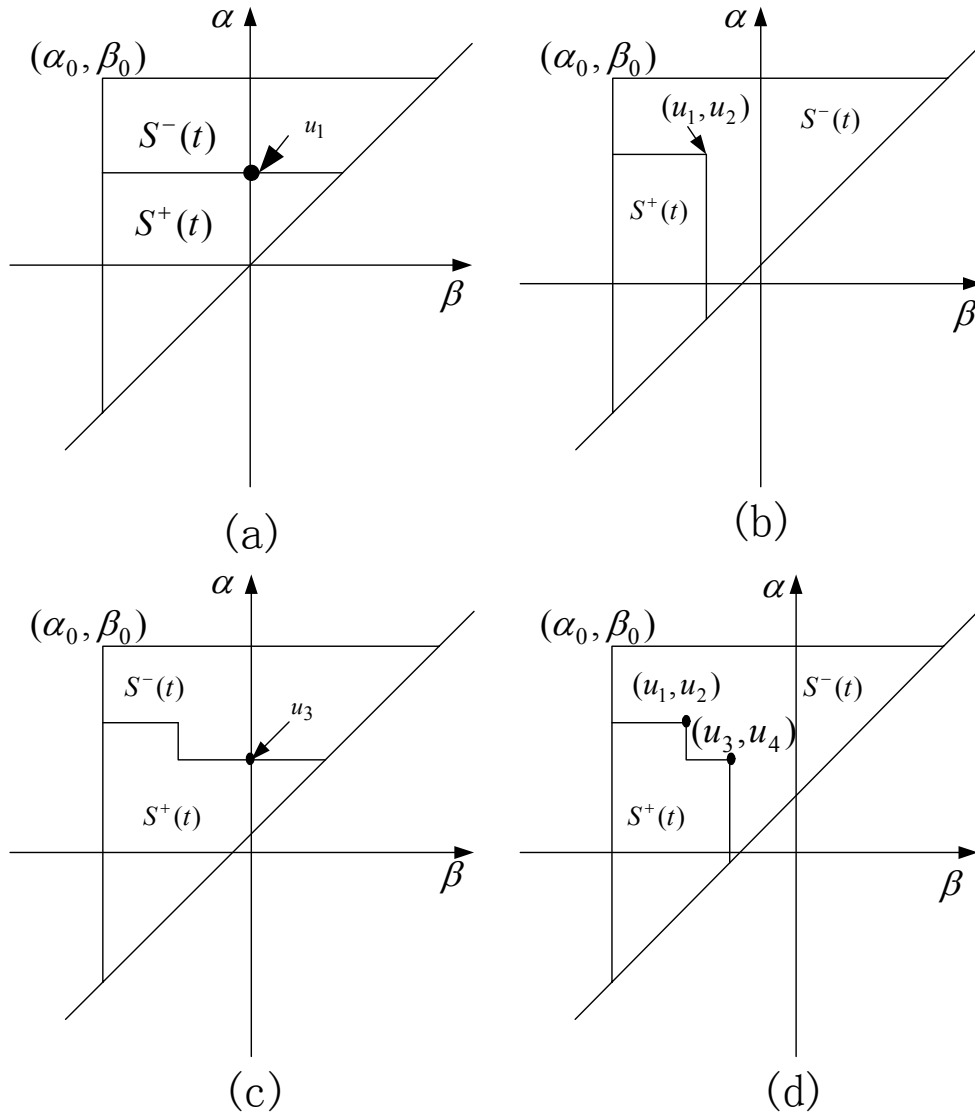


Figure 2-9 Geometric Interpretation of the Preisach Model

Next, let's assume that the input is monotonically decreasing until it reaches u_2 at time t_2 . As the input being decreased, all the relays with the 'down' switching value β above the current input $u(t)$ are turned 'off', which means their outputs become equal to -1. Geometrically, it changes the previous subdivision of T into positive and negative sets as shown in Fig 2-9 (b). The interface $L(t)$ between $S^+(t)$ and $S^-(t)$ now has two links, the horizontal and vertical. The vertical link moves from right to left and its motion is specified by the equation $\beta = u(t)$. The leftward motion of the vertical link is terminated when the input reaches its minimum value u_2 (Fig 2-9 (b)). The vertex of the interface $L(t)$ at t_2 has the coordinates $\alpha = u_1, \beta = u_2$.

Now, we assume that input is increased again until it reaches u_3 which is less than u_1 at time t_3 . Geometrically, this increase results in the formation of a new horizontal link of $L(t)$ which moves up. This upward motion is terminated when the maximum u_3 is reached (Fig 2-9 (c)).

Next, we assume that the input is decreased again until it reaches u_4 which is larger than u_2 . Geometrically, this input variation results in the formation of a new vertical link which moves from right to left. This motion is terminated as the input reaches its minimum value u_4 . As a result, a new vertex of $L(t)$ is formed which has the coordinates $\alpha = u_3, \beta = u_4$ (Fig 2-9(d)).

Continue this process, the states of all the relays can be recorded on the $\alpha - \beta$ plane. The states plus the weights of the relays can uniquely determine the output of the Preisach operator, and thus the geometric interpretation can give us a clear idea of how the Preisach model works.

It needs to be mentioned that, in Fig 2-9 (c), if $u_3 > u_1$, then the interface $L(t)$ will finally have just one horizontal link specified by the equation $\alpha = u_3$. The previous horizontal link $\alpha = u_1$ and vertical link $\beta = u_2$ will be erased. Similarly, in Fig 2-9 (d), if $u_4 < u_2$, the interface $L(t)$ will just like Fig 2-9 (b) with only one vertex which has the coordinates $\alpha = u_1$, $\beta = u_4$. This feature of the Preisach model is called wiping-out property which is formally defined as following:

Proposition 2-1: Wiping-out Property

Each local input maximum wipes out the vertices of $L(t)$ whose α -coordinates are below this maximum, and each local minimum wipes out the vertices whose β -coordinates are above this minimum.

In summary, the following rules can be used to geometrically interpret the Preisach operator:

1. At any instant of time, the triangular T is subdivided into two sets: $S^+(t)$ consisting of points (α, β) whose corresponding relays are in ‘on’ states, and $S^-(t)$ whose corresponding relays are in ‘off’ states.
2. The interface $L(t)$ between $S^+(t)$ and $S^-(t)$ is a staircase line whose vertices have α and β coordinates coinciding respective with local maxima and minima of input at previous instants of time
3. The final link of $L(t)$ is attached to the line $\alpha = \beta$ and it moves when the input is changed.

4. This link is a horizontal one and it moves up as the input is increased.
5. The final link is a vertical one and it moves from right to left as the input is decreased.
6. (Wiping-out Property) Each local input maximum wipes out the vertices of $L(t)$ whose α -coordinates are below this maximum, and each local minimum wipes out the vertices whose β -coordinates are above this minimum.

According to the above conclusion, at any instant of time the integral in (2.3) can be subdivided into two integrals, over $S^+(t)$ and $S^-(t)$, respectively:

$$y(t) = \iint_{S^+(t)} \mu(\alpha, \beta) \gamma_{\alpha\beta}(u(t)) d\alpha d\beta - \iint_{S^-(t)} \mu(\alpha, \beta) \gamma_{\alpha\beta}(u(t)) d\alpha d\beta \quad (2.4)$$

Since

$$\gamma_{\alpha\beta}(u(t)) = +1, \quad \text{if } (\alpha, \beta) \in S^+(t) \quad (2.5)$$

and

$$\gamma_{\alpha\beta}(u(t)) = -1, \quad \text{if } (\alpha, \beta) \in S^-(t) \quad (2.6)$$

from (2.4) we have:

$$y(t) = \iint_{S^+(t)} \mu(\alpha, \beta) d\alpha d\beta - \iint_{S^-(t)} \mu(\alpha, \beta) d\alpha d\beta \quad (2.7)$$

From this expression, it follows that an instantaneous value of output depends on a particular subdivision $L(t)$ of the limiting triangular T and the weighting function $\mu(\alpha, \beta)$. Since $L(t)$ can be calculated according to the system input, $\mu(\alpha, \beta)$ become the

only parameter needs to be identified before using the Preisach model. In fact, the identification of the Preisach model just means the identification of the weighting function $\mu(\alpha, \beta)$.

2.3.4 Discrete Preisach Model

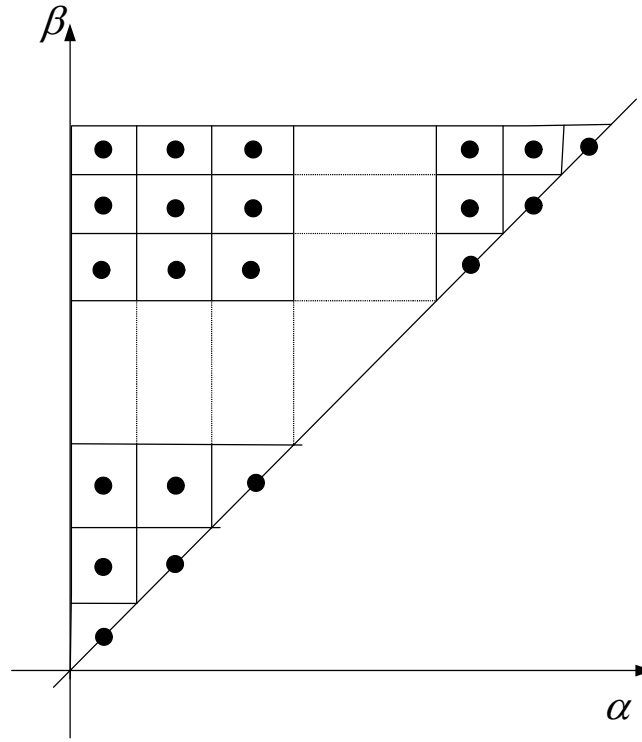


Figure 2-10: Discretization of $\alpha - \beta$ Plane

Given the weight function $u(\alpha, \beta)$, the continuous Preisach operator can be numerically implemented by using the formula (2.7). Although this approach is straightforward, it requires numerical evaluation of double integrals which is a time-consuming procedure and may impede the use of the Preisach model in practical applications. In most real world applications, it is sufficient to just consider a finite number of relays within the limiting triangular. To this end, the triangle T is subdivided

evenly into several square meshes (Fig 2-10) and we suppose each cell represent one relay γ_i whose threshold pair (α_i, β_i) is located at the center (black dot in Fig 2-10) of the cell. In the sequel, the weight function $u(\alpha, \beta)$ is no longer continuous, it becomes the set of discrete weights w_i . In this way, the integral becomes a summation and the output of the Preisach model can be approximated by:

$$y(t) = w_0 + \sum_{i=1}^N w_i \gamma_i(u(t), \xi(t)) \quad (2.8)$$

where w_0 is introduced here to account for the relays outside the limiting triangular.

So the discrete Preisach model is just a superposition of N non-ideal relays. It is easier to understand and has a simpler format compared with its continuous counterpart, so we will use the discrete Preisach model in this dissertation and whenever we mention the Preisach model we actually mean the discrete version.

CHAPTER 3

A NOVEL DYNAMIC PREISACH MODEL WITH DYNAMIC RELAY

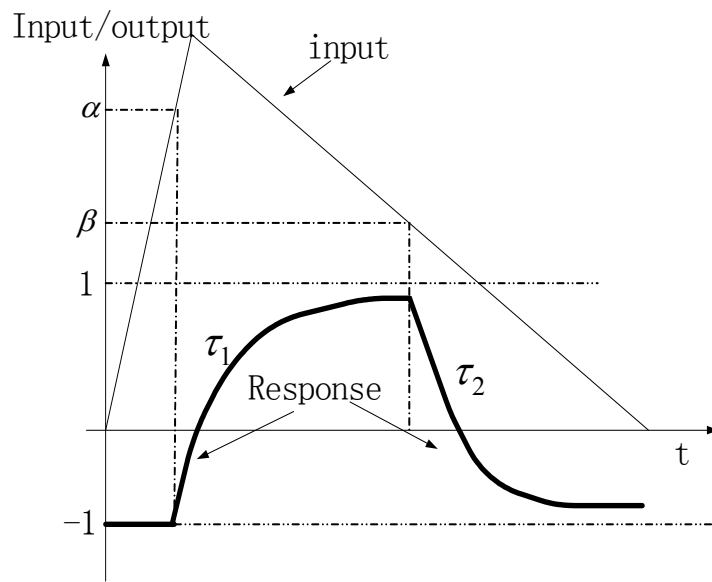
3.1 First-order Dynamic Relay (FDR)

As reviewed before, the Preisach model is mathematically defined as the sum of the weighted response of an infinite set of non-ideal relays $\gamma_{\alpha\beta}$ over all possible switching thresholds $\alpha \geq \beta$:

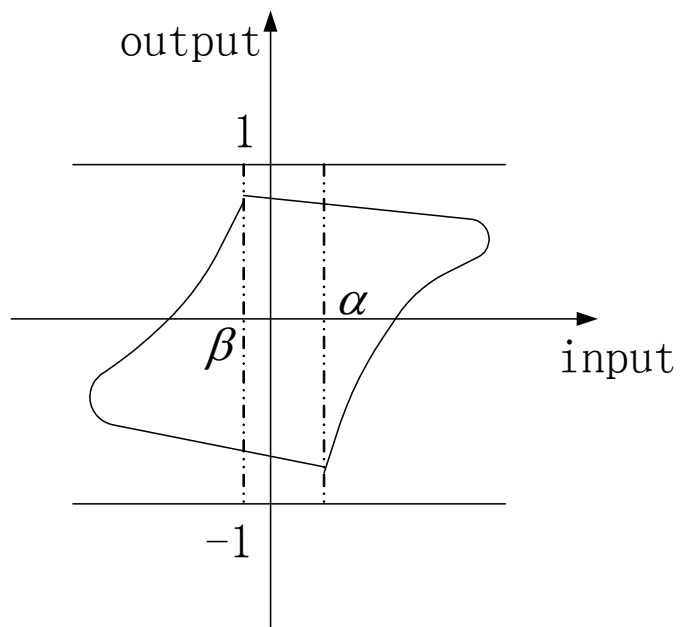
$$y(t) = \iint_{\alpha \geq \beta} \mu(\alpha, \beta) \gamma_{\alpha\beta}(u(t), \xi(t)) d\beta d\alpha \quad (3.1)$$

where β and α correspond to the lower and upper switching thresholds of the relay and $\xi(t)$ represents the state, ‘on’ or ‘off’, of the relay at time t .

Thus for the Preisach model, the non-ideal relay is the fundamental element that constitutes the overall hysteresis nonlinearity. Due to the rate-independent nature of the non-ideal relay, the Preisach model is inherently a static operator that cannot describe the dynamic hysteresis behavior. Hence, if we want to extend the classic Preisach model (CPM) to a dynamic operator and at the same time preserve its effectiveness for static hysteresis modeling, a straightforward way is to add some dynamic behavior into its building block—non-ideal relay.



(a)



(b)

Figure 3-1 First-order Dynamic Relay

(a) Time response of FDR;

(b) Input-output diagram of FDR for the same input as in (a)

As it is known that a non-ideal relay $y(t) = \gamma_{\alpha\beta}(u(t), \xi(t))$ will switch its output value instantaneously as the input crosses the thresholds. However, in the real world nothing can be changed instantaneously. We believe that the relay changes its output gradually from -1 (or 1) to 1 (or -1) and this transient process is too fast to be observed under low frequency input. However, when the input varies very fast, i.e. the time between switching on the relay and switching off the relay is comparable to time of the transient process, the effect of the transient response becomes significant. In this sense, the traditional non-ideal relay is just a low frequency approximation of the dynamic relay with the above assumed transient transition process; and the classical Preisach model is also a low frequency approximate of the dynamic Preisach model with the above dynamic relay.

Different transient response will give us different dynamic relay and thus different dynamic Preisach model. The most natural transient response is the exponential response, which is just the response of the first order dynamic system. Thus we propose that a dynamic relay has a dynamic response like a first-order dynamic system as shown in Fig 3-1 (a). The time response of a typical first order dynamic system is governed by the time constant (τ). And we allow the time constants of ascending (τ_1) and descending (τ_2) to be different. Thus the dynamic relay $y(t) = \gamma_{\alpha\beta}(u(t), \xi(t), t)$ (we add in a new parameter t to the relay operator to make it time dependent) is mathematically defined as:

$$y(t) = \begin{cases} -1 + (y(t_0) + 1) \cdot \exp\{-(\Delta t_2)/\tau_2\} & \text{if } \xi(t) = -1 \\ 1 - (1 - y(t_0)) \cdot \exp\{-(\Delta t_1)/\tau_1\} & \text{if } \xi(t) = 1 \end{cases} \quad (3.2)$$

where $\xi(t)$ is the state of the relay and

$$\xi(t) = \begin{cases} -1 & \text{if } u(t) < \beta \\ 1 & \text{if } u(t) > \alpha \\ \xi(t^-) & \text{if } \beta < u(t) < \alpha \end{cases}, \quad (3.3)$$

$$t_0 = \max(\{x < t : \xi(x) \neq \xi(x^-)\}), \quad (3.4)$$

and

$$\Delta t_1 = \min(t - t_0, 0), \text{ and } \Delta t_2 = \min(t - t_0, 0), \quad (3.5)$$

Based on the above definition, the output of the first-order dynamic relay could be any value between -1 and +1. Switching ‘on’ (or ‘off’) the dynamic relay does not indicate its output is +1 (or -1); instead this only means that the output begins to increase (or decrease) to +1 (or -1) from its present value ($y(t_0)$). Particularly, the first line of equation (3.2) tells us that when the relay is switched ‘off’, its output $y(t)$ may not necessarily be -1. It is determined by the time length for which the relay has been switched ‘off’ (Δt_2) and the distance between the initial value $y(t_0)$ and -1 (i.e. $y(t_0) - (-1) = y(t_0) + 1$). This means that for the same initial distance ($y(t_0) + 1$), the longer time the relay has been switched ‘off’, the closer to -1 the output is; on the other hand, for the same time length (Δt_2), the smaller the initial distance ($y(t_0) + 1$) is, the closer to -1 the output is. The second line of equation (3.2) can be understood in a similar way.

3.2 Dynamic Preisach Model with FDR

Similar to the Classical Preisach model, we use the First-order dynamic relay to build up our new dynamic Preisach model. Because the FDR has a first order response and the overall output of the Preisach model is just the weighted sum of all the individual dynamic relays, the Dynamic Preisach model is inherently rate-dependent and can be defined as:

$$y(t) = \iint_{\alpha \geq \beta} \mu(\alpha, \beta) \gamma_{\alpha\beta}(u(t), \xi(t), t) d\beta d\alpha \quad (3.6-a)$$

where $y(t) = \gamma_{\alpha\beta}(u(t), \xi(t), t)$ is the first-order dynamic relay defined by equations from (3.2) to (3.5). Similar to the classical Preisach model, by discretizing the $\alpha - \beta$ plane as shown in Fig 2-10, a discrete version of the (3.6-a) can be obtained:

$$y(t) = w_0 + \sum_{i=1}^N w_i \gamma_i(u(t), \xi(t), t) \quad (3.6-b)$$

For this new definition it is obvious that when the working frequency is very low, we can assume that the relay has reached its steady state before the input changes, and this model then reduces to the classical Preisach model.

3.3 Discussion

There are many ways to make the Preisach model rate-dependent. To this end, some people add the dynamic terms into the weight function (or discrete weights) [1, 24, 25], other people suggest cascading a Preisach operator with a linear dynamic system to model the dynamic hysteresis [14, 33]. All these methods can produce a dynamic

Preisach model, however, they are not as successful in modeling dynamic hysteresis as classical Preisach model did in modeling static hysteresis. The reason for this is that they are not developed based on the essence of the Preisach model.

Although the Preisach model has been generalized to a purely mathematical and phenomenon based model, the reason of its success is its agreement with the physical nature of the hysteresis. According to the Weiss theory, ferromagnets are composed of a large number of elementary magnets (domains), each of which behaves like a relay. So the Preisach model is just an easy and general way that can effectively described the physical nature of the hysteresis phenomenon. For this reason, any extension or modification of the Preisach model should not forget its physical basis.

The hysteron is believed to be the fundamental reason of the entire hysteresis. And the Preisach model is just the superposition of all these hysterons. As we discussed before, this concept is what makes the Preisach model successful. Keep this in mind, you may easily find that there is no reason to add the dynamic term into the weight function or to a separate dynamic system. A straightforward way is to make the elementary hysteresis operator—hysteron rate-dependent. In this way the entire Preisach model will be inherently dynamic, and its structure, which is believed to effectively reflect the physical nature of hysteresis, is kept unchanged.

The first-order dynamic process is very common in nature, and makes perfect sense here to describe the transition of the dynamic relay from the ‘off’ (or ‘on’) state to ‘on’ (or ‘off’) state. So the dynamic Preisach model consisting of the first order dynamic relay is a reasonable generalization of the classic Preisach model, and has been proved to be a better dynamic hysteresis model through our research.

3.4 Model verification

3.4.1 Interpretation of dynamic hysteresis using the new model

To start our discussion, let's first take a look at the hysteretic loops of the magnetostrictive actuator under different working frequencies.

From Fig 3-2, we can see that in the first five subplots, the higher the frequency, the less the output increases for a given increase in input, and the wider the hysteretic loop becomes. For frequencies higher than 200Hz the system even become a non-minimal phase system, defined as follows in light of our new model. When the input increases, the same input increment needs a shorter time to generate if the frequency is higher. This means that for the dynamic transition of any system with a given time constant, the output will be farther away from its steady-state. Similarly, the relevant (dynamic) relays will be farther away from their maxima and a smaller output increment results. When the input begins to decrease, some relays' outputs may still increasing towards their maxima, thus part of the output decrement will be counteracted by this type of inertia effect and a more smoothly decreasing curve is observed. When the frequency is so high that the input begins to decrease while most relays are just starting to rise, then the decrement may be even smaller than the increment, which will give us a non-minimal phase response. Hence, our new idea appears suitable to capture the dynamic hysteresis behavior.

3.4.2 Simulation Results

Based on the above discussion, we expect that our new model can effectively capture the dynamic hysteresis behaviors and have the same trend as the real actuator when the working frequency is increasing. Now, let's examine the effectiveness of our model through simulation.

To get a better understanding of the new dynamic Preisach model, let's first consider the hysteretic system consisting of just one 'First-order dynamic relay'. The inputs in this experiment are just some sinusoidal waves with frequencies from 10 Hz to 300 Hz (same as the frequencies in Fig 3-3). The weight of this single relay is set arbitrarily because the main purpose here is to observe the shapes of the hysteresis loops under different frequencies rather than caring about the exact output values. The time constants τ_1 and τ_2 are selected in a trial and error manner to make the hysteresis loops in the simulation mimic the real hysteresis loops in Fig 3-3. Finally, the loops in Fig 3-4 are generated with the following parameters:

Time constant: $\tau_1 = \tau_2 = 0.001$ (sec) ,

Weight: $w = 100$,

Output bias: $w_0 = 100$

Input: $u(t) = w_0 + w \sin(2\pi f t)$, $f = 10, 20, 50, 100, 150, \dots, 300 \text{ Hz}$

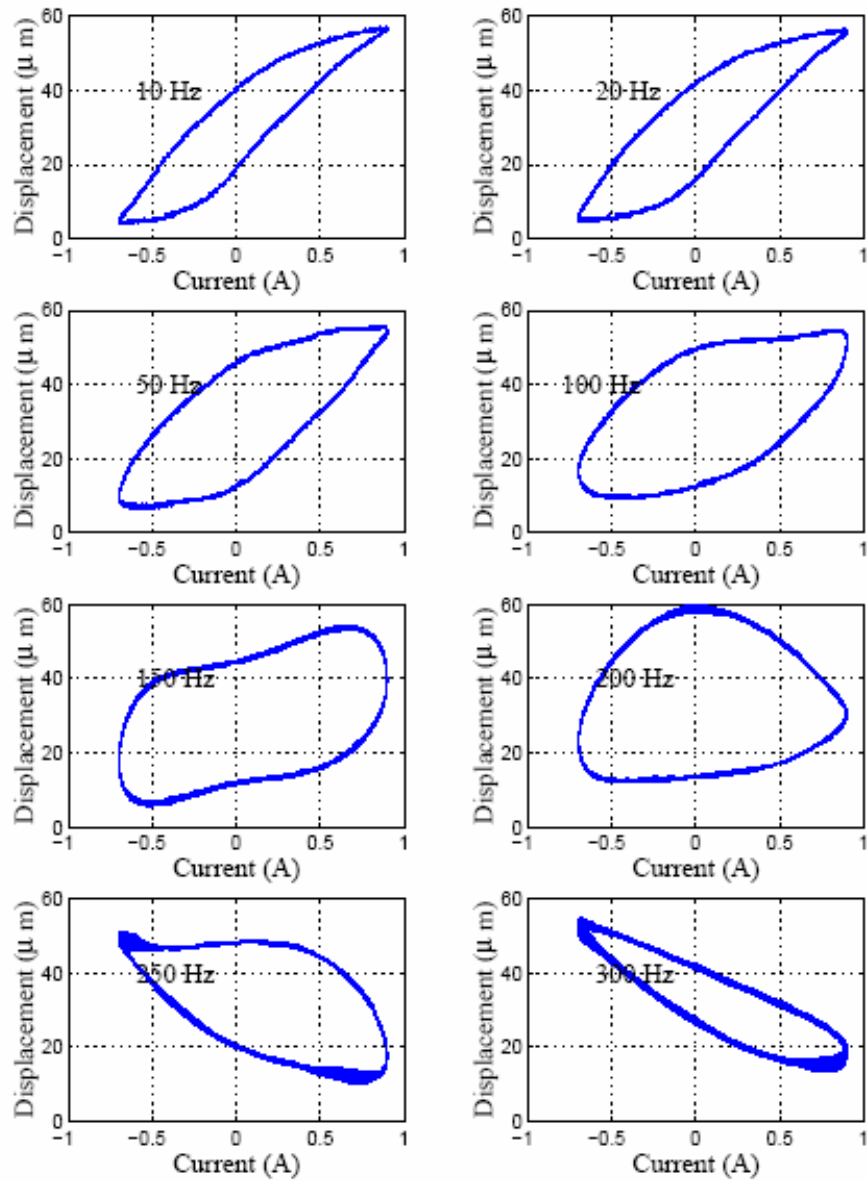


Figure 3-2: Hysteretic Loops of Magnetostrictive Actuators under Different Frequencies [2]

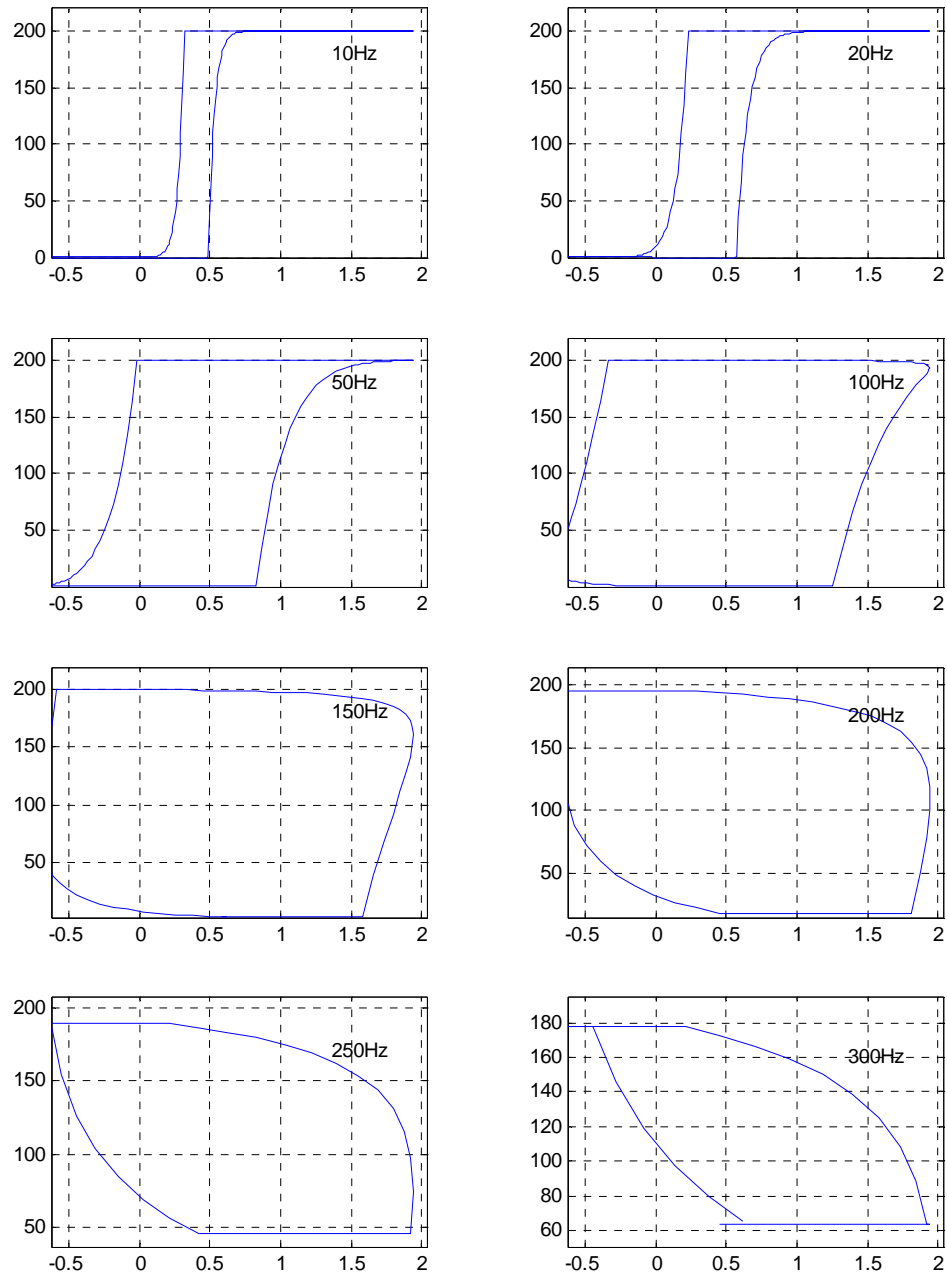


Figure 3-3: Response of single FDR under different frequencies

The way we set the input has no special meaning, although we use the same values for the bias and the magnitude as in the relay. Setting the input in this way will make the output easy to observe.

One thing that needs to be mentioned is that transient responses exist for the dynamic relay. So when you repeat the same sinusoidal wave, you can see several different loops. In Fig 3-4, only the steady-state hysteretic loops are recorded.

From Fig 3-4 we can see that a single ‘dynamic relay’ can also generate complicated hysteresis loops under different frequencies. Because just one FDR is involved in this simulation, the response curves are not that smooth, especially for the low frequencies. However, it is sufficient to prove that our dynamic relay has the same trend in changing the shapes as the real hysteretic system does (Fig 3-3). We can imagine that if the more dynamic relays are added to the dynamic Preisach model, the dynamic hysteresis loops it generates will become smoother. By manipulating the weights of each relay, the dynamic loops can mimic any desired complicated shapes.

3.5 Identification Methods

We assume the two additional parameters—time constants (τ_1, τ_2) are independent of the weights of the first order relays. This assumption allows us to identify the weights and the time constants separately. Before talking about the identification details of the weights and time constants, let's first review some basic concepts of the system identification.

3.5.1 System Identification

To better express the identification methods, system identification theory is first introduced. Structure selection in the identification theory is skipped here because the model structure (dynamic Preisach model) has already been proposed. Some general and basic principles that guide the parameter identification methods are first reviewed.

I Principles behind parameter identification methods

As a general introduction of the principles of the parameter estimation, no particular form of the model is assumed. A general model structure $\Gamma(\theta)$ that can represent any model is employed during the explanation. ($\theta \in D_\Gamma \subset R^d$ is the parameter vector with dimension d and D_Γ is the domain of the θ which is a subset of R^d). The only restriction here is that the model is assumed to be discrete with respect to time, because this is the way a computer works.

A model $\Gamma(\theta)$, in fact, represents a way of predicting future outputs. The prediction can be expressed by:

$$\hat{y}(k / \theta) = \Gamma(k, Z^{k-1}; \theta) \quad (3.7)$$

where k is the discrete time instant, Γ represents a model, $\hat{y}(k / \theta)$ represents the model's output under the particular parameter θ at time k , and Z^i represents all the experimental data from the instant of the time 1 up to the instant of time i . Suppose we have a total of N experimental data pairs, then

$$Z^N = [y(1), u(1), y(2), u(2), \dots, y(N), u(N)] \quad (3.8)$$

The problem of the parameter identification is to decide upon how to use the information contained in Z^N to select a proper value $\hat{\theta}$ of the parameter vector. Formally speaking, we have to determine a mapping from the data Z^N to the set D_Γ :

$$Z^N \rightarrow \hat{\theta} \in D_\Gamma \quad (3.9)$$

Such a mapping is a parameter estimation method.

In order to find a good estimation method, we need a test by which the different models' ability to describe the observed data can be evaluated. It is believed [4] that the essence of a model is its prediction aspect, and we shall also judge its performance in this respect. Thus let the prediction error at time k of a certain model $\Gamma(k, Z^{k-1}; \theta)$ be given by:

$$\varepsilon(k, \theta) = y(k) - \hat{y}(k / \theta) \quad (3.10)$$

When the data set Z^N is known, these errors can be computed for $k = 1, 2, \dots, N$.

A good model, we say, is one that is good at predicting, that is, one that produces small prediction errors when applied to the observed data. Thus a guiding principle [4] for parameter estimation is:

Based on Z^n we can compute the prediction error $\varepsilon(n, \theta)$ using (3.11). Select $\hat{\theta}$ so that the prediction errors $\varepsilon(n, \hat{\theta})$, $n=1, 2, \dots, N$, become as small as possible

The prediction error sequence $\varepsilon(k, \theta)$, $k=1, 2, \dots, N$ can be seen as a vector in R^N . The size of this vector can be measured using any norm in R^N . This leaves a substantial number of choices. We shall restrict the freedom somewhat by only considering the following way of evaluating “how large” the prediction-error sequence is: Let the prediction-error sequence be filtered through a stable linear filter $L(q)$:

$$\varepsilon_F(k, \theta) = L(q)\varepsilon(k, \theta) \quad (3.11)$$

Then use the following norm:

$$V(\theta, Z^N) = \frac{1}{N} \sum_k^N l(\varepsilon_F(k, \theta)) \quad (3.12)$$

where $l(\cdot)$ is a scalar-valued function.

The function $V(\theta, Z^N)$ is, for a given Z^N , a well-defined scalar-valued function of the model parameter. It is a natural measure of the validity of the model $\Gamma(k, Z^{k-1}; \theta)$.

The estimate $\hat{\theta}$ is then defined by minimization of (3.12):

$$\hat{\theta} = \hat{\theta}(Z^N) = \arg \min_{\theta \in D_\Gamma} V(\theta, Z^N) \quad (3.13)$$

This way of estimating θ contains many well-known and much used procedures. We shall use the general term prediction-error identification methods (PEM) for the family of approaches that corresponds to (3.13). Particular methods, with specified “name” attached to themselves, are obtained as special cases of (3.13) depending on the choice of $l(\cdot)$, the choice of prefilter $L(\cdot)$, the choice of model structure, and, in some cases, the choice of method by which the minimization is realized.

II Linear regressions and the lest squares method

Let's talk about a most important case of PEM (3.13)—Least squares method. Here we assume the model has a linear regression structure which describes the input-output relationship as a linear difference equation:

$$\begin{aligned} y(k) + a_1 y(k-1) + a_2 y(k-2) + \dots + a_{n_a} y(k-n_a) \\ = b_1 u(k-1) + b_2 u(k-2) + \dots + b_{n_b} u(k-n_b) + e(k) \end{aligned} \quad (3.14)$$

where $y(k)$ is the output of the model at time k , $u(k)$ is the input of the model at time k , $e(k)$ is the noise getting into the model at time k , and $\theta = [a_1 \ a_2 \ \dots \ a_{n_a} \ b_1 \ b_2 \ \dots \ b_{n_b}]^T$ is the parameter vector for this model. Because the output $y(k)$ is linear with respect to θ and depends not only on the previous input but also the previous output, the model is referred as linear regression.

If we define $\varphi(k)$ as:

$$\varphi(k) = [-y(k-1) \ -y(k-2) \ \dots \ -y(k-n_a) \ u(k-1) \ u(k-2) \ \dots \ u(k-n_b)] \quad (3.15)$$

the output prediction $\hat{y}(k)$ based on the model structure and the previous data can be written as:

$$\hat{y}(k) = \varphi^T(k) \cdot \theta + e(k) \quad (3.16)$$

With the above equation, the prediction error becomes:

$$e(k) = \hat{y}(k) - \varphi^T(k) \cdot \theta \quad (3.17)$$

and the criterion function resulting from (3.11) and (3.12), with $L(q) = 1$ and $l(\varepsilon) = \frac{1}{2} \varepsilon^2$,

is:

$$V(\theta, Z^N) = \frac{1}{N} \sum_k \frac{1}{2} [\hat{y}(k) - \varphi^T(k) \cdot \theta]^2 \quad (3.18)$$

This is the least-squares criterion for the linear regression (3.14). The unique feature of this criterion, developed from the linear parametrization and the quadratic error norm, is that it is a quadratic function of θ . Therefore, it can be minimized analytically.

Let $m = \max(n_a, n_b)$, and $d = n_a + n_b$ is the dimension of the parameter vector.

Suppose we have $N + m$ data pairs,

$$Z^N = [y(1), u(1), y(2), u(2), \dots, y(N + m), u(N + m)]$$

denote Φ as:

$$\Phi = \begin{bmatrix} \varphi^T(m+1) \\ \varphi^T(m+2) \\ \vdots \\ \varphi^T(m+N) \end{bmatrix} = \begin{bmatrix} \varphi_1(m+1) & \varphi_2(m+1) & \cdots & \varphi_d(m+1) \\ \varphi_1(m+2) & \varphi_2(m+2) & \cdots & \varphi_d(m+2) \\ \vdots & \vdots & \vdots & \vdots \\ \varphi_1(m+N) & \varphi_2(m+N) & \cdots & \varphi_d(m+N) \end{bmatrix} \quad (3.19)$$

where the subscript j of $\varphi_j(i)$ represents the j^{th} element of vector $\varphi(i)$.

With the above notation, the solution $\hat{\theta}$ that minimize the cost function (3.18) is given by:

$$\hat{\theta} = (\Phi^T \Phi)^{-1} \Phi^T Y \quad (3.20)$$

where $Y = [y(m+1) \ y(m+2) \ y(m+3) \ \cdots \ y(m+N)]$.

The least square method gives the optimal estimation of the parameters with respect to the assumed model structure and the experimental data. It is an extremely important method in system identification.

III Parameter estimation by numerical search

For a certain structure of the model $\Gamma(k, Z^{k-1}; \theta)$ such as linear regression, (3.12) can be minimized by analytical methods. However, there are no mathematical solutions

of the minimization for a general model structure. So for most nonlinear models, the solution of (3.13) has to be found by iterative, numerical techniques.

Methods for numerical minimization of a function $V(\theta)$ update the estimate of the minimizing point iteratively. This is usually done according to:

$$\hat{\theta}^{(i+1)} = \hat{\theta}^{(i)} + \alpha \cdot f^{(i)} \quad (3.21)$$

where $f^{(i)}$ is a search direction based on information about $V(\theta)$ acquired at previous iterations, and α is a positive constant determined so that an appropriate decrease in the value of $V(\theta)$ is obtained. Depending on the ways we compute $f^{(i)}$, numerical minimization methods can be divided into three groups:

1. $f^{(i)}$ is based on the function values $V(\theta)$ only.
2. $f^{(i)}$ is based on the function values as well as the gradient
3. $f^{(i)}$ is based on the function values, gradient, and Hessian (the second derivative matrix)

The typical member of group 3 corresponds to Newton algorithms, where the correction in (3.21) is chosen in the “Newton” direction:

$$f^{(i)} = -[V''(\hat{\theta}^{(i)})]^{-1} V'(\hat{\theta}^{(i)}) \quad (3.22)$$

The most important subclass of group 2 consists of quasi-Newton methods, which somehow form an estimate of the Hessian and then use (3.22). Algorithms of group 1 either form gradient estimates by difference approximations and proceed as quasi-Newton methods or have other specific search patterns.

The above discussion is about the general principles of the numerical parameter estimation. Now, let's look at an example that has the special case of scalar output and quadratic criterion:

$$V(\theta, Z^N) = \frac{1}{N} \sum_{k=1}^N \frac{1}{2} \varepsilon^2(k, \theta) \quad (3.23)$$

This problem is known as “the nonlinear least-squares problem” in numerical analysis. The criterion (3.23) has the gradient:

$$V'(\theta, Z^N) = -\frac{1}{N} \sum_{k=1}^N \psi(k, \theta) \varepsilon(k, \theta) \quad (3.24)$$

where $\psi(k, \theta)$ is the $d \times p$ gradient matrix of $\varepsilon(k, \theta)$ ($p = \dim y$) with respect to θ . A general family of search routines is then given by:

$$\hat{\theta}^{(i+1)} = \hat{\theta}^{(i)} - \mu^{(i)} [R^{(i)}]^{-1} V'(\hat{\theta}^{(i)}, Z^N) \quad (3.25)$$

where $\hat{\theta}^{(i)}$ denotes the i^{th} iterate. $R^{(i)}$ is a $d \times d$ matrix that modifies the search direction, and the step size $\mu^{(i)}$ is chosen so that:

$$V(\hat{\theta}^{(i+1)}, Z^N) < V(\hat{\theta}^{(i)}, Z^N) \quad (3.26)$$

The simplest choice of $R^{(i)}$ is to take it as the identity matrix,

$$R^{(i)} = I \quad (3.27)$$

which makes (3.25) the gradient or steepest-descent method. This method is fairly inefficient close to the minimum. Newton methods typically perform much better there.

For (3.23), the Hessian is:

$$V''(\theta, Z^N) = \frac{1}{N} \sum_{k=1}^N \psi(k, \theta) \psi^T(k, \theta) - \frac{1}{N} \sum_{k=1}^N \psi'(k, \theta) \cdot \varepsilon(k, \theta) \quad (3.28)$$

where $\psi'(k, \theta)$ is the $d \times d$ Hessian of $\varepsilon(k, \theta)$.

Choosing:

$$R^{(i)} = V''(\hat{\theta}^{(i)}, Z^N) \quad (3.29)$$

makes (3.25) a Newton method.

3.5.2 Weights Identification

The proposed dynamic hysteresis model is just a weighted response of a large number of first order dynamic relays. Thus the first step of the identification of the proposed model is to determine the weights of these dynamic relays. It has been mentioned that the only difference between the proposed model and the classical Preisach model is that we change the non-ideal relay to a first-order dynamic relay. When the working frequency is very low (such as 1 Hz), the output of the dynamic relay is almost the same as the static relay. Thus in this case we can approximate the proposed model by the classical Preisach model. In other words, using the low frequency data, if we can identify a set of the weights for the classical Preisach model, these weights can also be used to approximate the weights of the dynamic relay in the proposed model. Based on this idea, we can use the same identification method as the one used for the classical Preisach model to identify the weights in our proposed model. The only requirement here is that the data used in the identification should be almost static.

Several identification method of classical Preisach model has been proposed. For the discretized CPM model, since the output, given the input data, is just a linear combination of the weights of the relays, the least square method talked in section 3.5.1 is a powerful tool to identify the weights.

The discrete Preisach model is defined as following:

$$y(t) = w_0 + \sum_{i=1}^M w_i \gamma_i(u(t), \xi_i(t)) \quad (3.30)$$

Given a set of experiment data pairs $(u(n), y(n))$, $n = 1, 2, \dots, N$, the following equations can be obtained:

[illegible]

the state of each relay $\xi_i(n)$ can be uniquely determined through the input history $u(1), u(2), \dots, u(n)$ at any constant of time n , as can the output of each relay operator $\gamma_i(u(n), \xi_i(n))$ (either +1 or -1). Hence the only unknowns in the above equation (3.31) are the weights, $w_i, i = 0, \dots, M$, which can be easily identified through a least squares method.

Let's define:

$$Y = [y(1) \ y(2) \ y(3) \ \cdots \ y(N)]^T \quad (3.32)$$

$$\Phi = \begin{bmatrix} 1 & \gamma_1(u(1), \xi(1)) & \cdots & \gamma_M(u(1), \xi(1)) \\ 1 & \gamma_1(u(2), \xi(2)) & \cdots & \gamma_M(u(2), \xi(2)) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \gamma_1(u(N), \xi(N)) & \cdots & \gamma_M(u(N), \xi(N)) \end{bmatrix} \quad (3.33)$$

then based on the least squares method (3.20), the optimal weights

$w = [w_0 \ w_1 \ w_2 \ \cdots w_M]^T$ according to experimental data $(u(n), y(n))$, $n = 1, 2, \dots, N$ are:

$$w = (\Phi^T \Phi)^{-1} \Phi^T Y \quad (3.34)$$

3.5.3 Time constants Identification

Our identification of the time constants is based on the following assumptions:

1. All the first-order dynamic relays that constitute the dynamic Preisach model share the same time constants (τ_1, τ_2)
2. The time constants (τ_1, τ_2) are rate-independent. However, they may depend on the temperature or the external force. At the current stage of our project, since we fix the temperature and the external force, we assume that the time constants are just two positive real numbers.
3. The weights $w_i, i = 0, \dots, M$ are rate-independent and the time constants are independent of the weights. Since the time constants have no effect for the static (or very low frequency) data, the weights can be first identified through the static data using the least square methods. Then the weights are presumed to be known and fixed during the identification of the time constants. The weights may also vary with the temperature or external force. Again, we neglect this effect because the temperature and force are currently fixed.

The first assumption tells us there are in total two dynamic parameters need to be identified. The third assumption allows us to identify the weights and the time constants separately. We call the third assumption the **Separation Law** which is the foundation of the identification of the time constants.

With the weights known and fixed, the time constants are nonlinear parameters in the proposed model ((3.2) to (3.5)). So if we choose a criterion function resulting from

(3.11) and (3.12), with $L(q) = 1$ and $l(\varepsilon) = \frac{1}{2} \varepsilon^2$, the minimization problem in (3.13) can not be solved analytically. Thus the parameters of time constants (τ_1, τ_2) should be found numerically. Based on the numerical parameter estimation introduced in section 3.5.1, the following procedure is developed for the identification of time constants.

1. Gather the experimental data pairs $Z^N = [y(1), u(1), y(2), u(2), \dots, y(N), u(N)]$
2. For the current parameters $\theta^{(i)} = [\tau_1, \tau_2]$, using the formula (3.6), compute our model's predictions $\hat{y}(1), \hat{y}(2), \dots, \hat{y}(N)$ based on the data Z^N and the weights $w_i, i = 0, \dots, M$ that have been identified through static data.
3. Compute the error function $V(\theta, Z^N)$:

$$V(\theta, Z^N) = \frac{1}{N} \sum_k^N (y(k) - \hat{y}(k / \theta))^2 \quad (3.16)$$

4. If the value of $V(\theta, Z^N)$ is small enough stop, otherwise continue to 5.
5. Compute the search direction $R^{(i)}$ according to (3.28) and (3.29), update the parameter $\theta^{(i+1)}$ based on (3.25) and then go back to step 2.

CHAPTER 4

MODELING EXPERIMENT RESULTS

4.1 System Setup

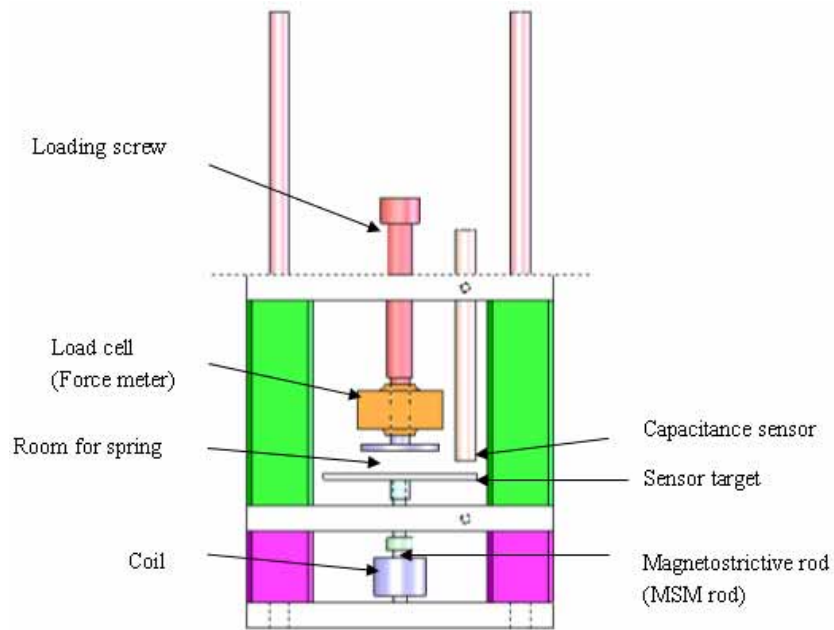


Figure 4-1 Experiment System setup (Cross-section view)

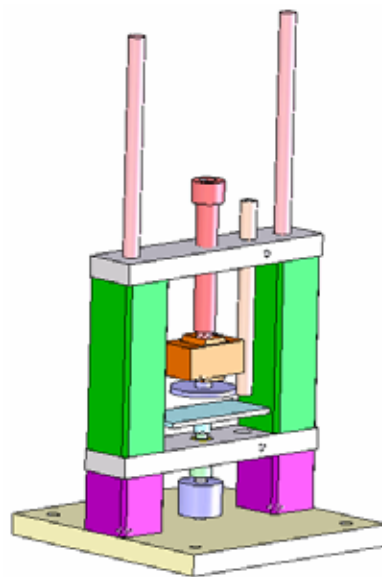


Figure 4-2: Experiment System Setup (Angle view)

All the experiments used in our modeling are performed by Energen Inc company. The test setup is shown in the above two drawings. The sensor target is connected to a solid plunger piece which sits on the top of the MSM rod. A loading force is applied by the top screw and the force range can be read from the load cell. When the input current is given, the coil generates a magnetic field along the central axis and causes an elongation (displacement) in the MSM. The expanded MSM pushes the plunger and the target up to change the gap between the eddy current sensor and the target. This movement causes a voltage change of the sensor and can be converted to the displacement data. The entire testing fixture is clamped on an isolation table which is mounted on a solid wall to avoid ground noise. Finally, the experiments are performed under the conditions as shown in Table 4-1

Temperature:	Room temperature ($\sim 20^{\circ}\text{C}$)
MSM material:	KelvinAll [®]
Driving coil:	Copper coil ($\sim 170 \text{ Oe /Amp}$)
Pre-load force:	100 lbs
Pre-load spring:	Belleville washer spring ($k \sim 17000 \text{ lb/in}$)
Displacement sensor:	Capacitance sensor ($1\text{V} = 50\mu\text{m}$)
Power amplifier:	Techron LVC 5050

Table 4-1: The experiment Conditions

4.2 Experiments Description

As we described in Chapter 3, the whole identification process can be divided into two uncorrelated process: weights identification and time constants identification. For this reason, two sets of experiments are performed.

4.2.1 Low Frequency Experiments

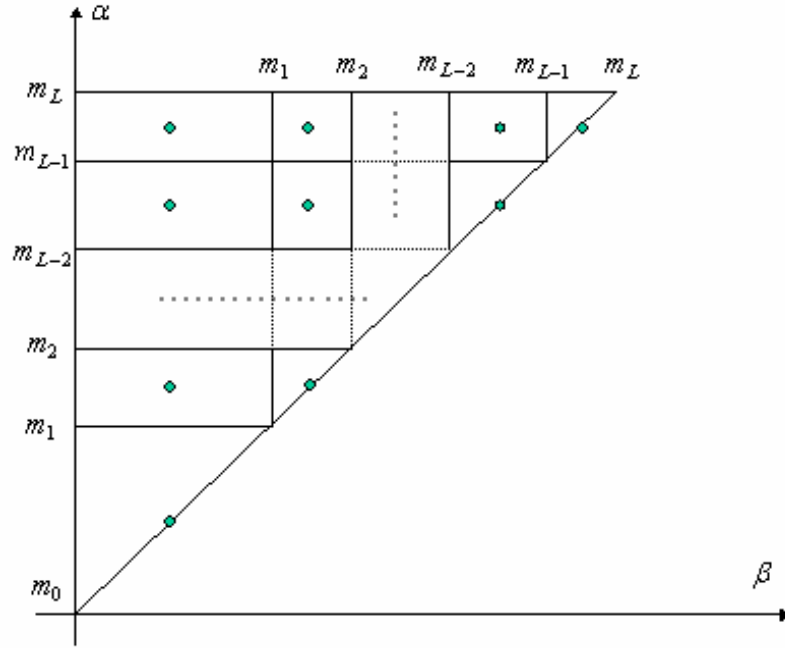


Figure 4-3: Uneven Discretization scheme

The first set of experiments is designed for the weights identification. As discussed in Chapter 3, our dynamic Preisach model is almost the same as the classical Preisach model, except that we use the First-order dynamic relay instead of traditional nonideal relay. When the working frequency is very low, the first-order dynamic relay reduces to the traditional nonideal relay. In addition we assume that the meaning of the

weights in our dynamic Preisach model is the same as those in the classical Preisach model. Thus using the low frequency data (1 Hz) we can identify the weights for our dynamic model through the same identification method as the classical Preisach model.

The first step of the design of the identification experiment is to decide what the input range is and how to discretize it. In our particular actuator system, the input current could be from 0 A to 14 A. So the input range is set to be $u_range=[0,14]$ and the following discrete level is finally selected:

$$u=[1.5 \ 2 \ 3 \ 3.5 \ 4 \ 4.5 \ 5 \ 5.5 \ 6 \ 6.5 \ 7 \ 7.5 \ 8 \\ 8.5 \ 9 \ 9.5 \ 10 \ 10.5 \ 11 \ 11.5 \ 12 \ 12.5 \ 13 \ 13.5 \ 14]$$

There are in total 25 discrete levels and they divide the limiting triangle into $(1+25)*25/2=325$ cells. Since these selected values are not evenly spaced within the input range, the cells are thus not equally sized (Fig 4-3). However, this makes no difference in essence with the equal discretization, where we also suppose each cell just represents one relay located at the center of the cell.

To identify all the weights without ambiguity, the input must be able to single out the effect of each relay (cell). Based on the geometric interpretation in Chapter 2, the following input can meet the above requirement:

Proposition 4-1:

If the input range of the Preisach model is discretized by $m_0, m_1, m_2, \dots, m_L$, then the discrete input sequence

$$\begin{aligned} &c_1, c_2, \dots, c_L, c_{L-1}, \dots, c_2, c_1, \\ &c_1, c_2, \dots, c_{L-1}, c_{L-2}, \dots, c_2, c_1 \\ &c_1, c_2, \dots, c_{L-2}, c_{L-3}, c_2, c_1 \\ &\dots \\ &c_1, c_2, c_1 \end{aligned} \tag{4.1}$$

$$\text{where } c_i = \{x : x \in R \text{ and } x - m_i = \min_{j=1,2,\dots,L} (x - m_j)\}$$

can single out the effect of each individual relay, i.e the experimental data generated by this set of inputs can be used to identify the weights of the relays without ambiguity.

The definition of c_i tells us that c_i could be any set of real numbers that are closer to m_i than to any other $m_j, j \neq i$. Since the limiting triangle is divided into a mesh by $m_0, m_1, m_2, \dots, m_L$, the definition of c_i can guarantee that each c_i can just turn on (or off) one line (or column) of mesh points (relays) in the discrete limiting Triangle in Fig 2-10. Then the combination of the input sequences in (4.1) will make sure the least square matrix $(\Phi^T \Phi)^{-1}$ is nonsingular.

Based on the above proposition, the inputs of the low frequency experiments are chosen to be L different sinusoidal waves with magnitudes $m_0, m_1, m_2, \dots, m_L$, and for each magnitude m_i the sampling rate should be high enough such that we can obtain a sequence of samples $c_1, c_2, \dots, c_i, c_{i-1}, \dots, c_2, c_1$, where c_i is as defined in proposition 4-1.

4.1.2 High Frequency Experiments

After the weights identification, we also need to design experiments to identify the dynamic terms—time constants. The effect of the dynamic terms can be observed only through the dynamic data. As discussed in Chapter 3, the identification of the time constants can be done by numerical methods, which search for the best estimation of the time constants that minimize the total errors between the real experiment data and the

prediction. Hence the only thing we need here is to generate a good set of dynamic data for the numerical estimation.

Theoretically, two data pairs are sufficient to identify the two time constants. However, noise that is a part of real experimental data must be compensated for by a large enough data set. In addition, since the noise level for the sensor is the same for different inputs, the bigger the input amplitude is the smaller the noise ratio. So in the high frequency experiments, we only employ the sinusoidal waves with the largest amplitude of 14A. Although theoretically we suppose the time constants are rate-independent, practically they may have a weak dependence on frequency. So instead of using one set of frequency data, we select different frequencies ranging from 1 Hz to 100 Hz for the identification of time constants. Finally, the following inputs are used to generate experiment data for time constants identification:

$$u(t) = 7 \sin(2\pi f \cdot t) + 7 \quad (4.2)$$

where $f = 1\text{Hz}, 5\text{Hz}, 10\text{Hz}, \dots, 100\text{Hz}$

4.2 Weights identification

4.2.1 Preprocess the data

The data used for the weights identifications are obtained through the experiments described in section 4.1.1. Due to the limitation of the sensors used in the experiments, the data are noisy. The noise is typically about 1 micron and may go up to 2 microns. So

for small magnitude sinusoidal inputs (such as 1.5A), the noise level may be as high as 20%. Thus we need to smooth the data before the identification. The smoothing method employed here is just a simple averaging filter. The results are shown in Fig 4-4.

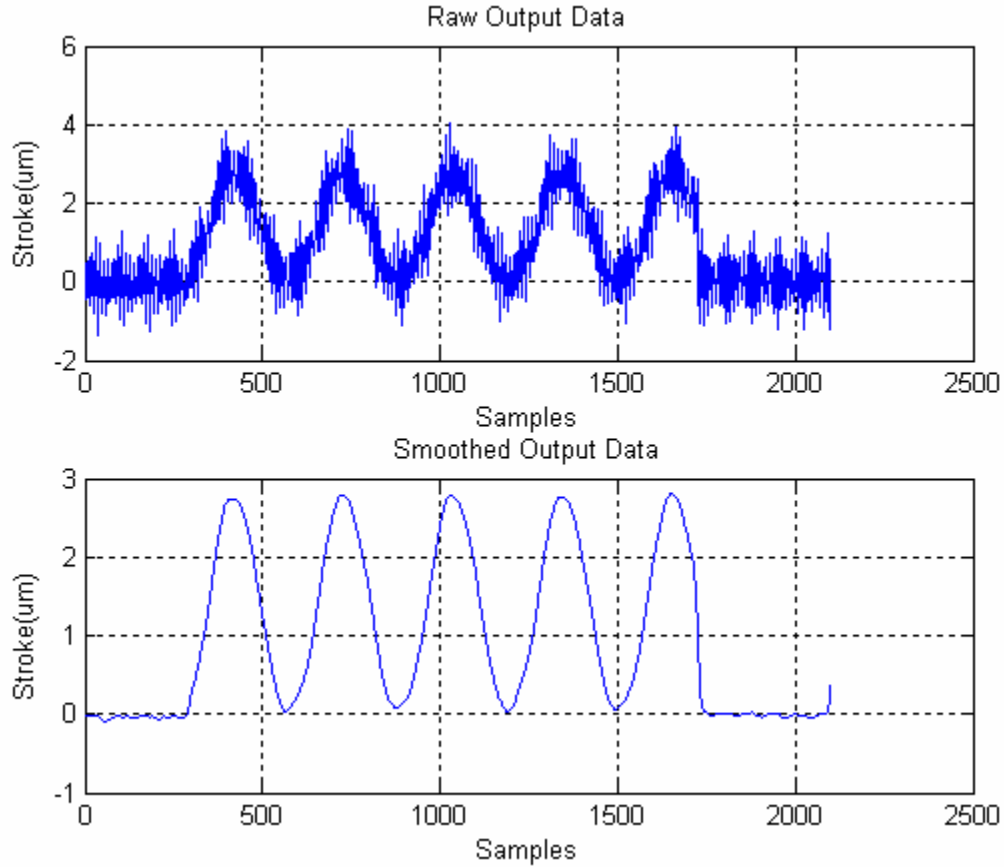


Figure 4-4: Smoothing the experiment data
(Input Magnitude=1.5A; the same method is used for other magnitudes).

4.2.2 Data selection for the identification

Our identification is based on the smoothed data (Section 4.2.1). However, we didn't use all the smoothed data. There are several periods of data available for each input magnitude. We average them to get only one period for each magnitude.

The one-period data for each magnitude is still more than sufficient. Thus we sample the one-period data of each magnitude according to discrete level it passes. If the sampling points are too close to the thresholds, then the noise will easily change the status ('on' or 'off') of the relays which will make our identification algorithm unstable. Hence we need to select the experiment data samples whose inputs are far away from adjacent thresholds. So the best sampling points of the one-period data for magnitude m_i (as shown in Fig 4-3) are the ones generated by the inputs $m_1, m_2, \dots, m_i, m_{i-1}, \dots, m_2, m_1$. In this way, the smaller the magnitude the fewer samples we can use. After sampling, we cascade the samples from all of the selected magnitudes and use them in the identification process. The data used for identification is illustrated in Fig 4-5.

4.2.3 Identification

Given the identification data, the least square method discussed in Chapter 3 is employed to identify the weights. The identified weights are shown in Fig 4-6, and the model performance with the identified weights is shown in Fig 4-7.

From Fig 4-7 we can see that the identified Preisach model can fit the static data exceptionally well. One may argue it is not as persuasive as it appears because the model performance is verified with the same data used for the identification. For this reason, we also generate a set of test data that are not used during the identification process to test our model. The test data are one-period of sinusoidal waves with different magnitudes that are randomly selected from the real experiment data. The prediction results with respect to the test data are shown in Fig 4-8 and Fig 4-9.

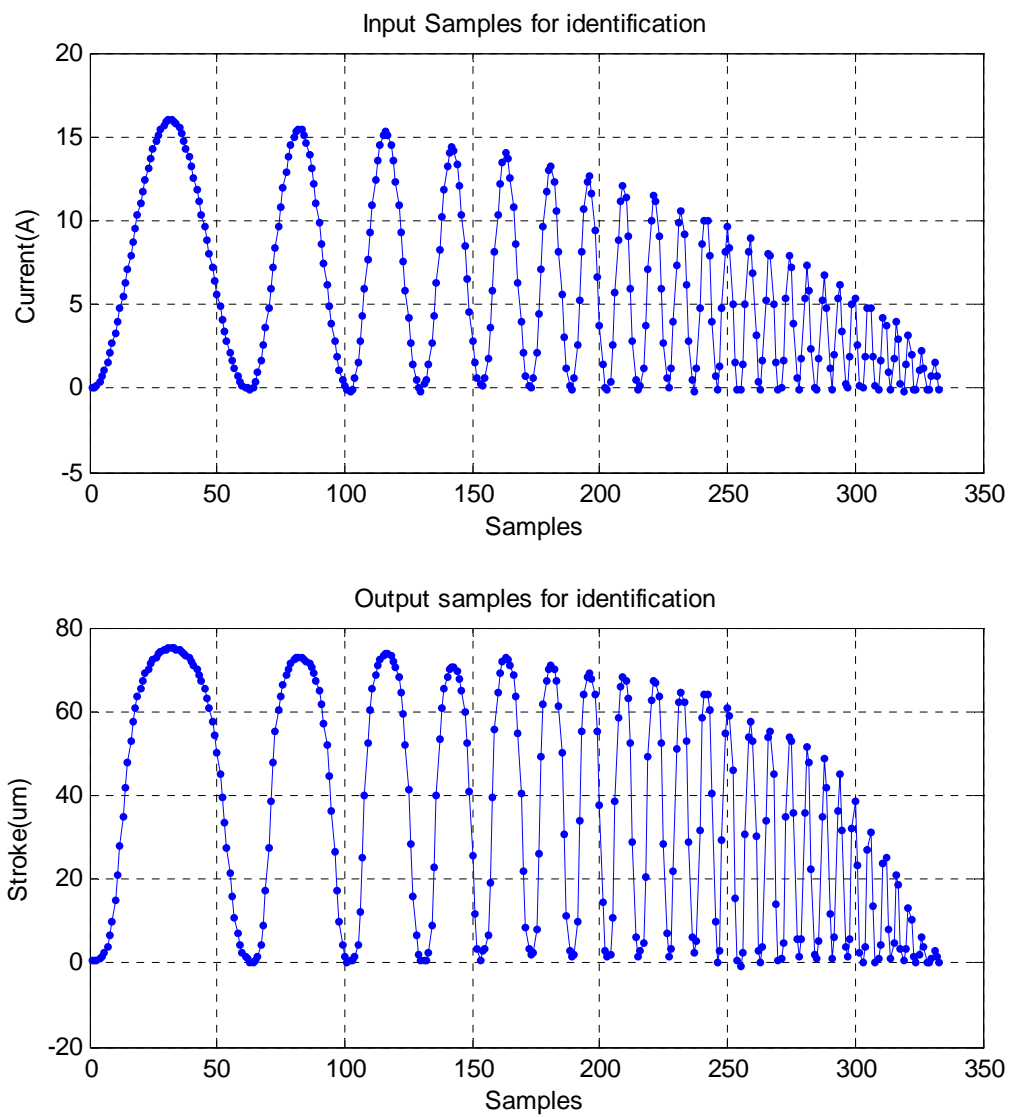


Figure 4-5: Sampled data for identification

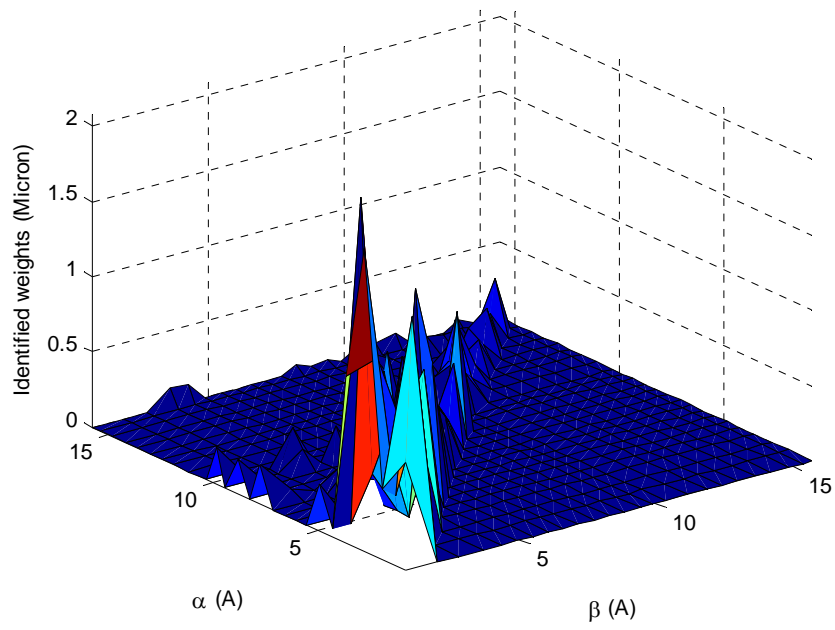


Figure 4-6: Identified weights

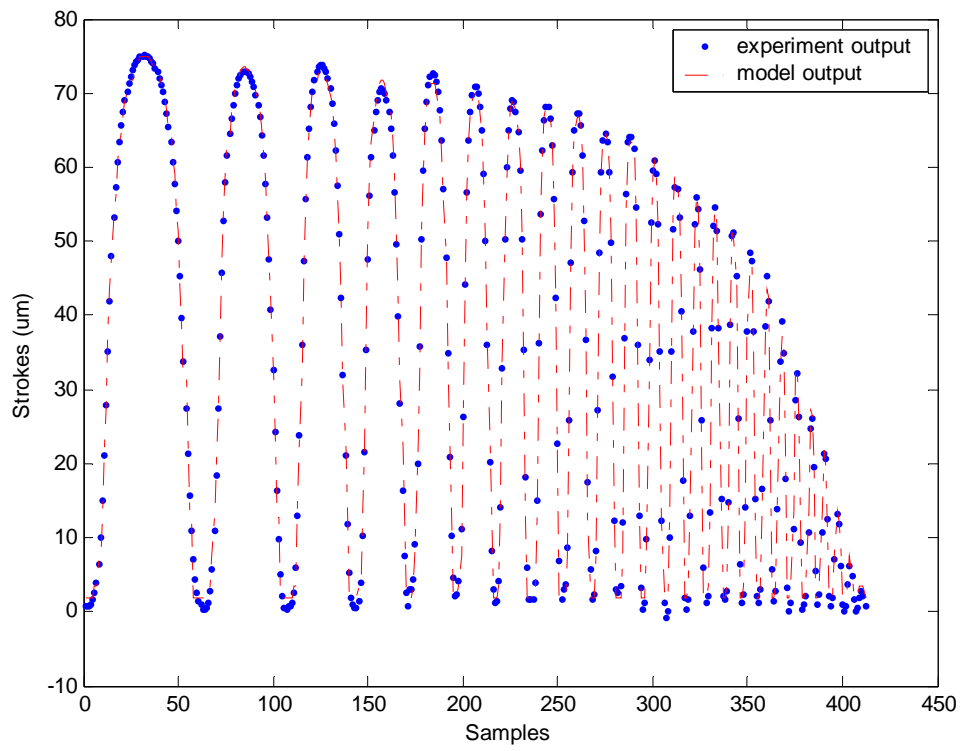


Figure 4-7: Model Performance for 1Hz identification data

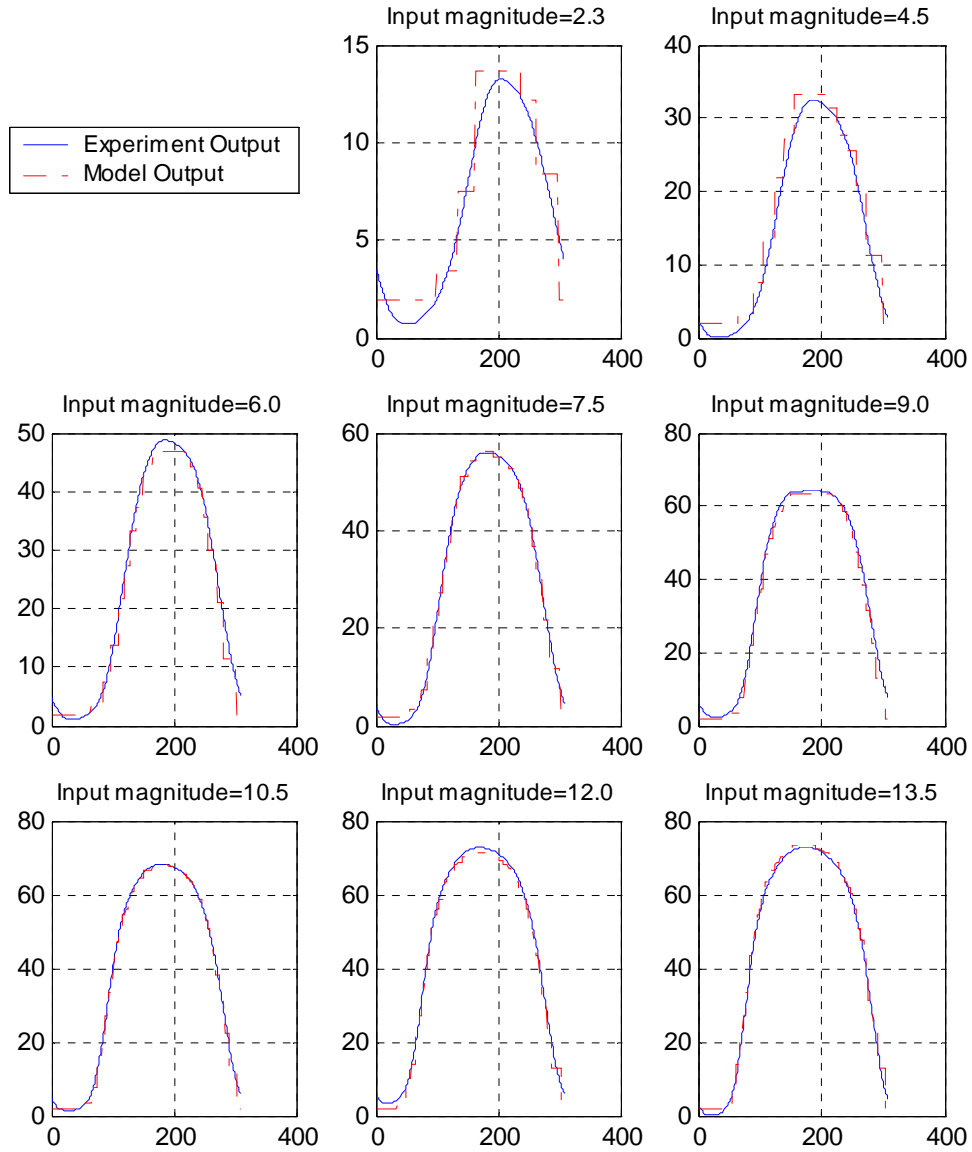


Figure 4-8: Model Performance for the test data (Output).
(X-axis: Samples; Y-axis: Strokes (um))

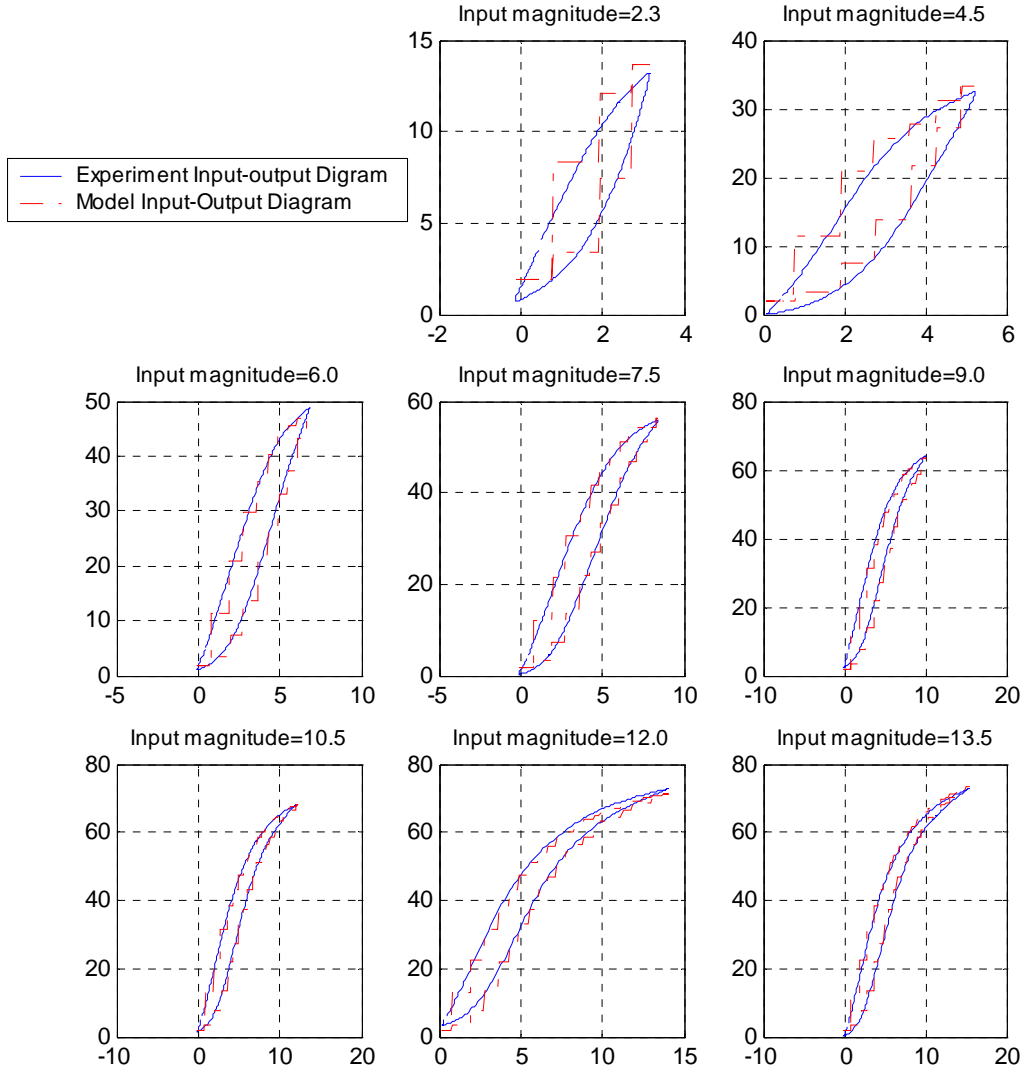


Figure 4-9: Model Performance for the test data (Input-Output Diagram)
(X-axis: Input current (A); Y-axis: Strokes (um))

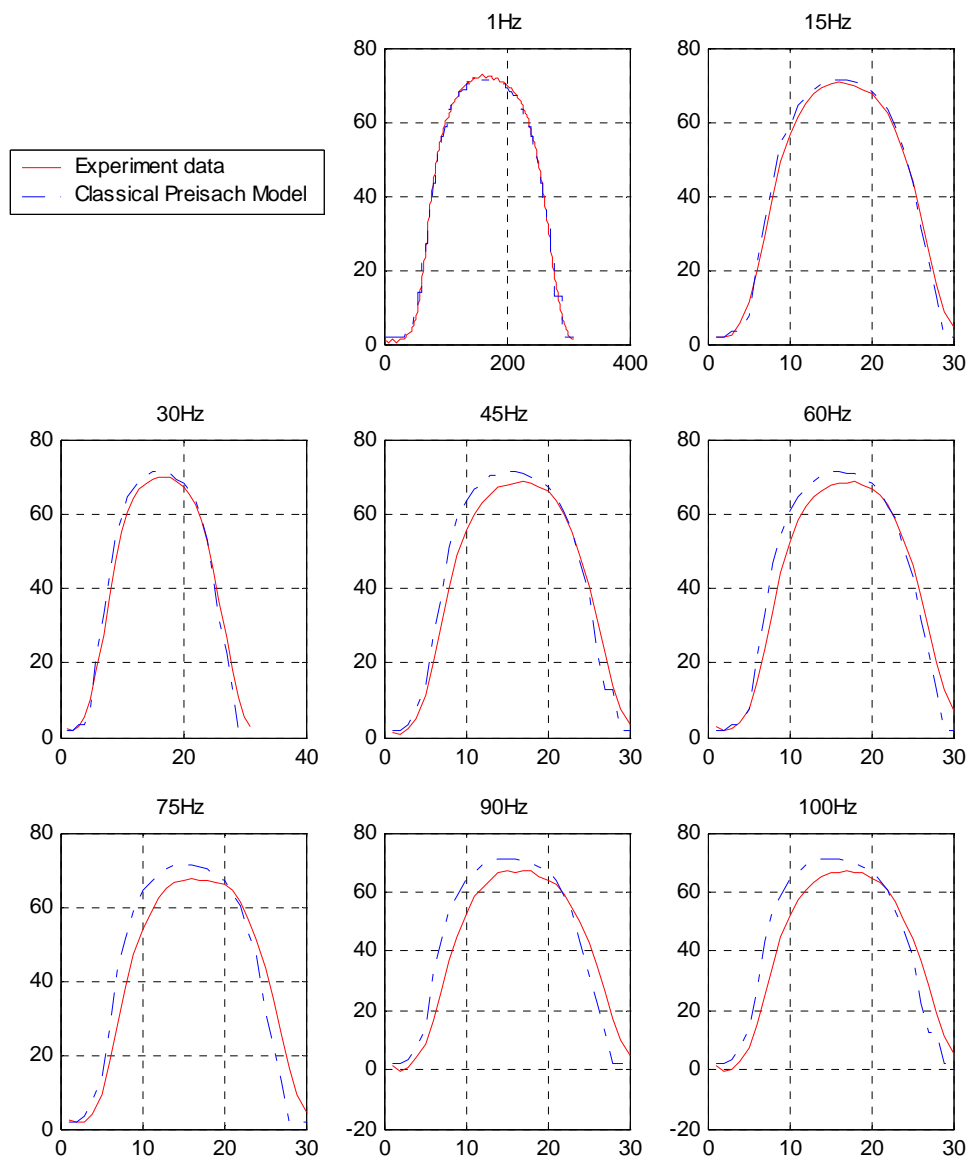


Figure 4-10: Classical Preisach Model Performance for Dynamic Hysteresis data
(X-axis: Samples; Y-axis: Strokes (um))

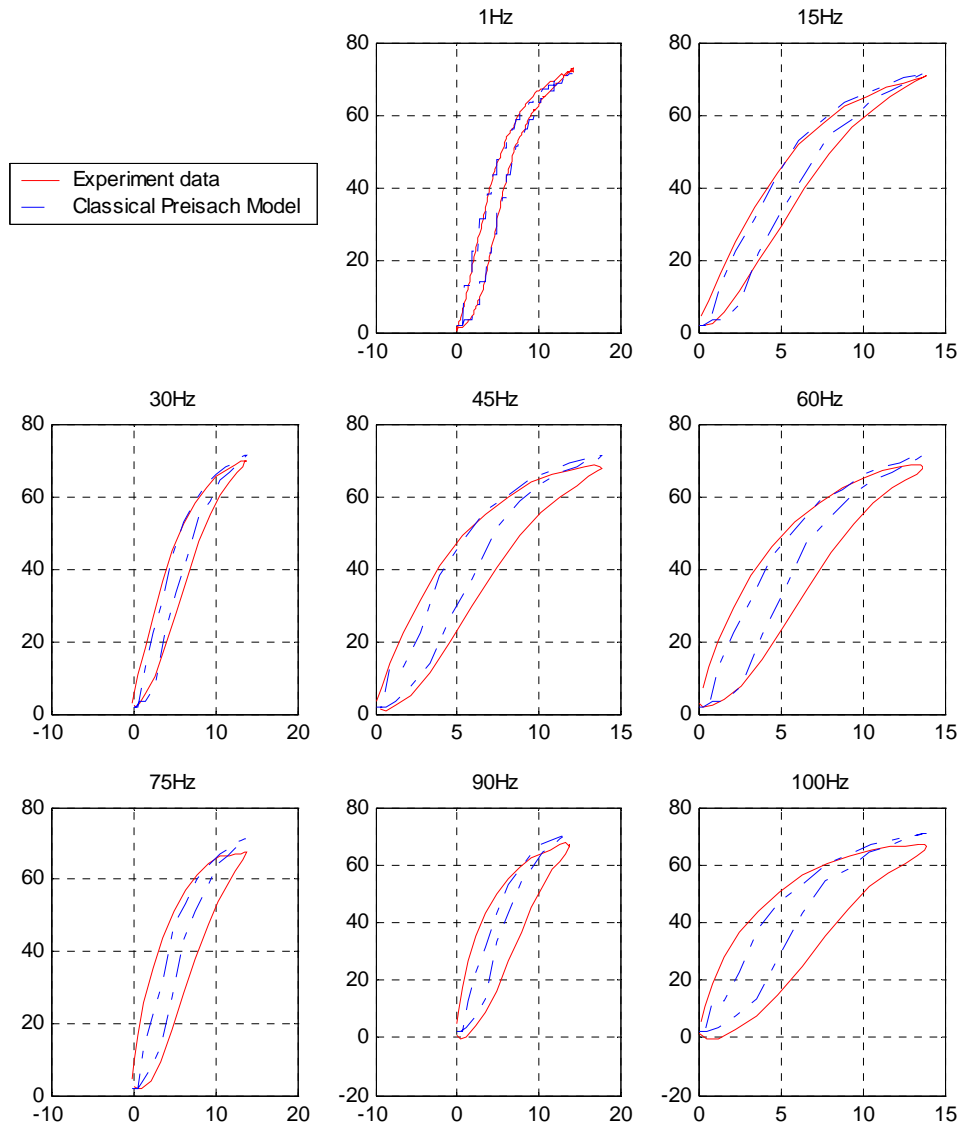


Figure 4-11: Classical Preisach Model Performance for Dynamic Hysteresis data
(X-axis: Input current (A); Y-axis: Strokes (um))

4.3 Time Constants Identification

Without the dynamic terms, our model is just the classical Preisach model. Although it can model the static hysteresis data very well, its performance of modeling the dynamic hysteresis data is very poor (Fig 4-10 and Fig 4-11). From Fig 4-10 and Fig 4-11 we can see that the higher the frequency the worse the classical Preisach model performs. So for many dynamic applications of magnetostrictive actuators, the classical Preisach model is inappropriate, which is the main reason for proposing our dynamic Preisach model.

The time constants are the only parameters in our dynamic Preisach model that the classical Preisach model do not have. To identify them, the following procedure is taken:

- 1 First, we smooth the data to remove the noise.
- 2 Second, for each frequency we delete the incomplete cycles in the smoothed data.
- 3 Next, the data from different frequencies are connected together.
- 4 Last, the Newton algorithm as described in 3.5.3 is employed to identify the time constants.

. For our particular magnetostrictive actuator, the time constants (τ_1, τ_2) are identified to be:

$$[\tau_1, \tau_2] = [0.0008764, 0.0006478].$$

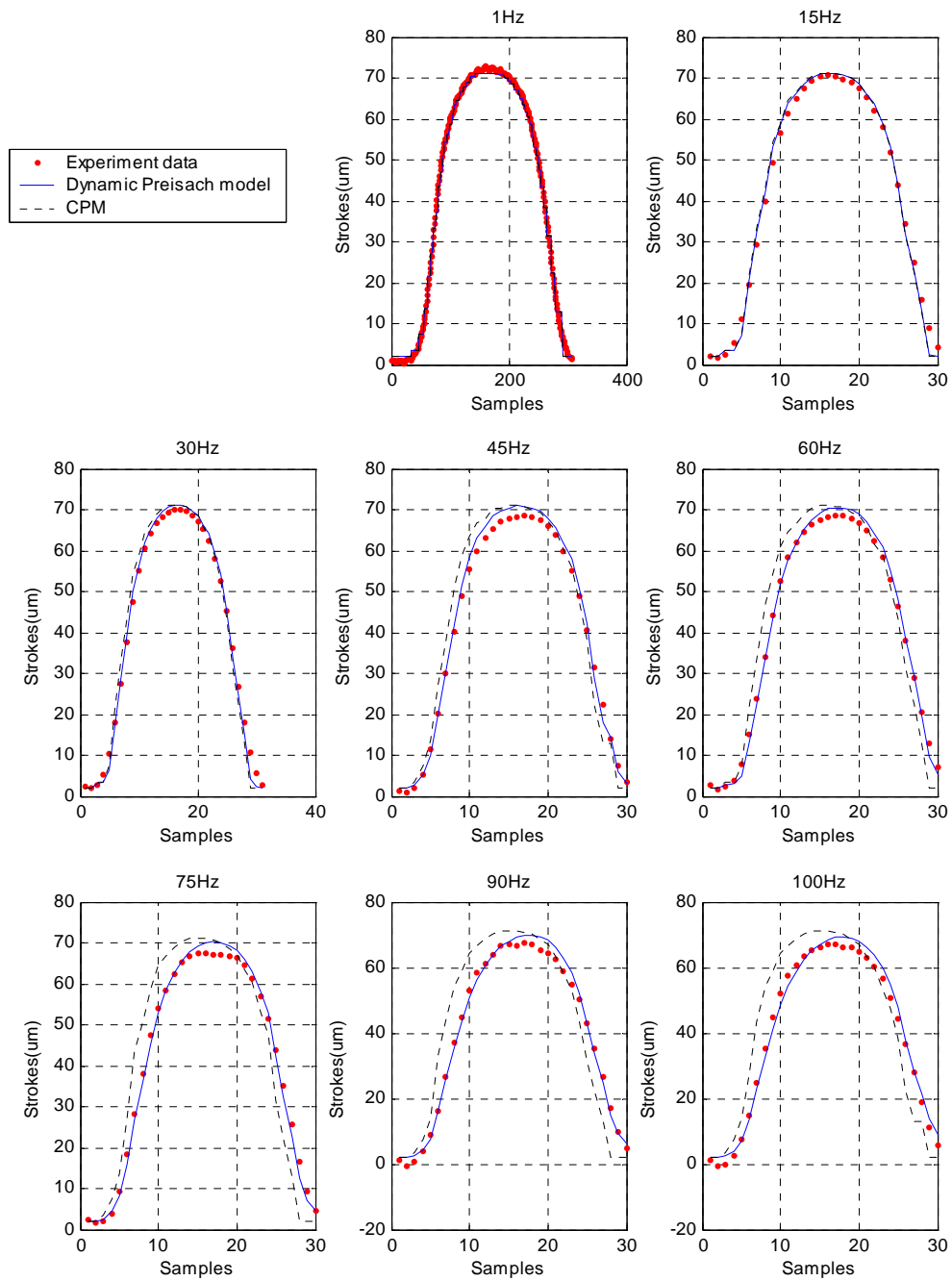


Figure 4-12: Dynamic Modeling Results for the identification data
(Output plot)

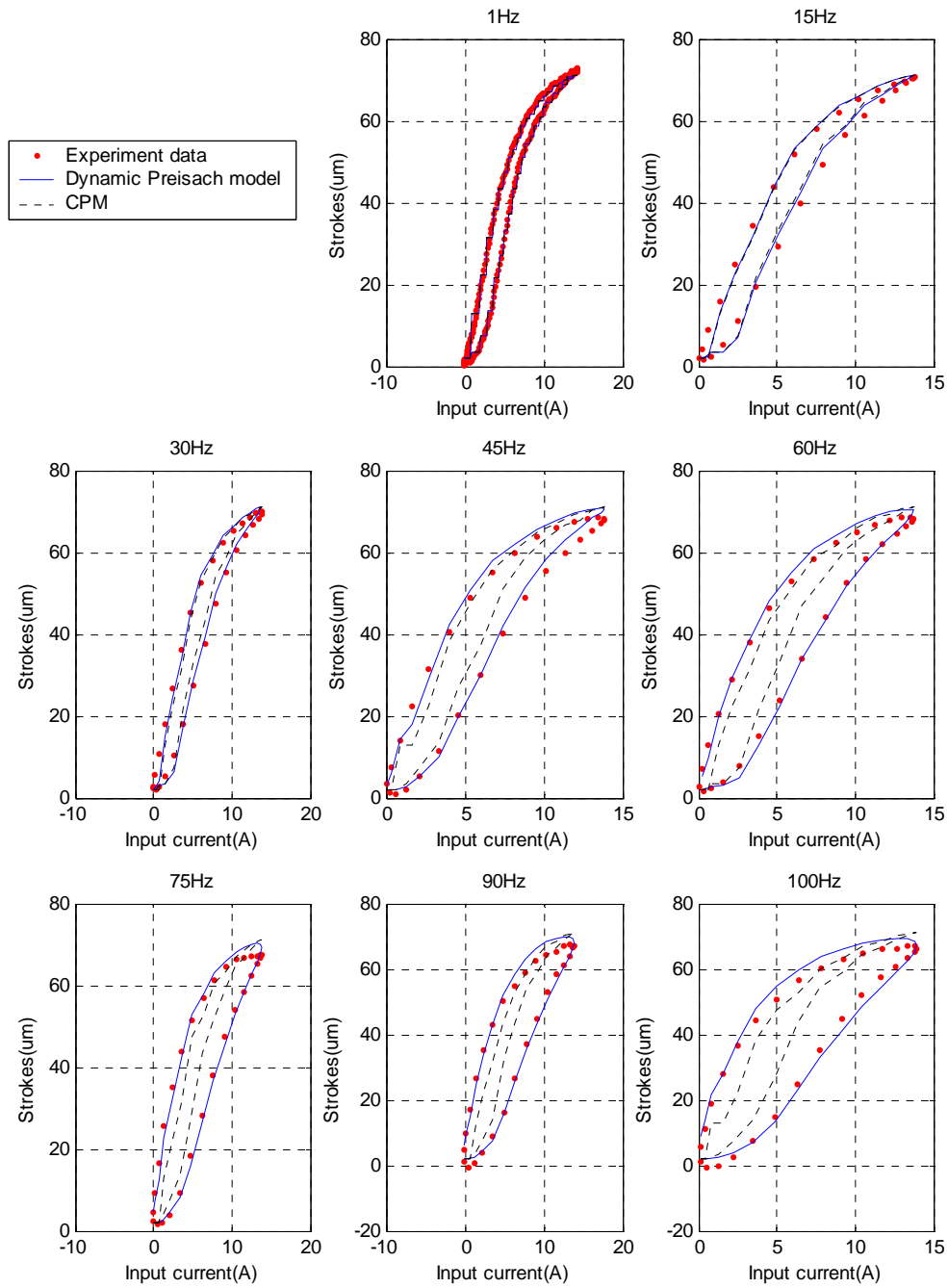


Figure 4-13: Dynamic Modeling result for identification data
(Input-Output Diagram)

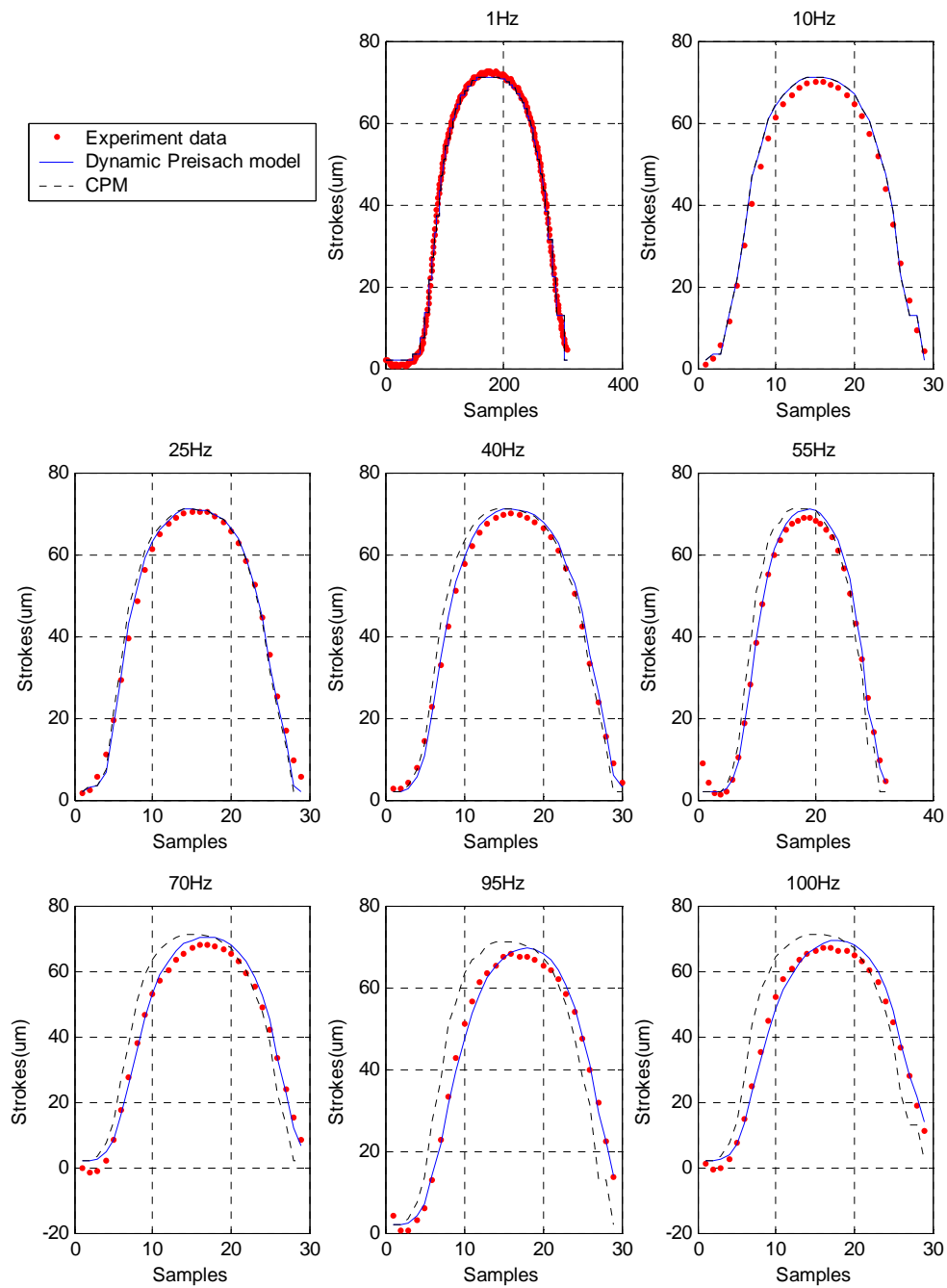


Figure 4-14: Dynamic Modeling Results for test data
(Output plots)

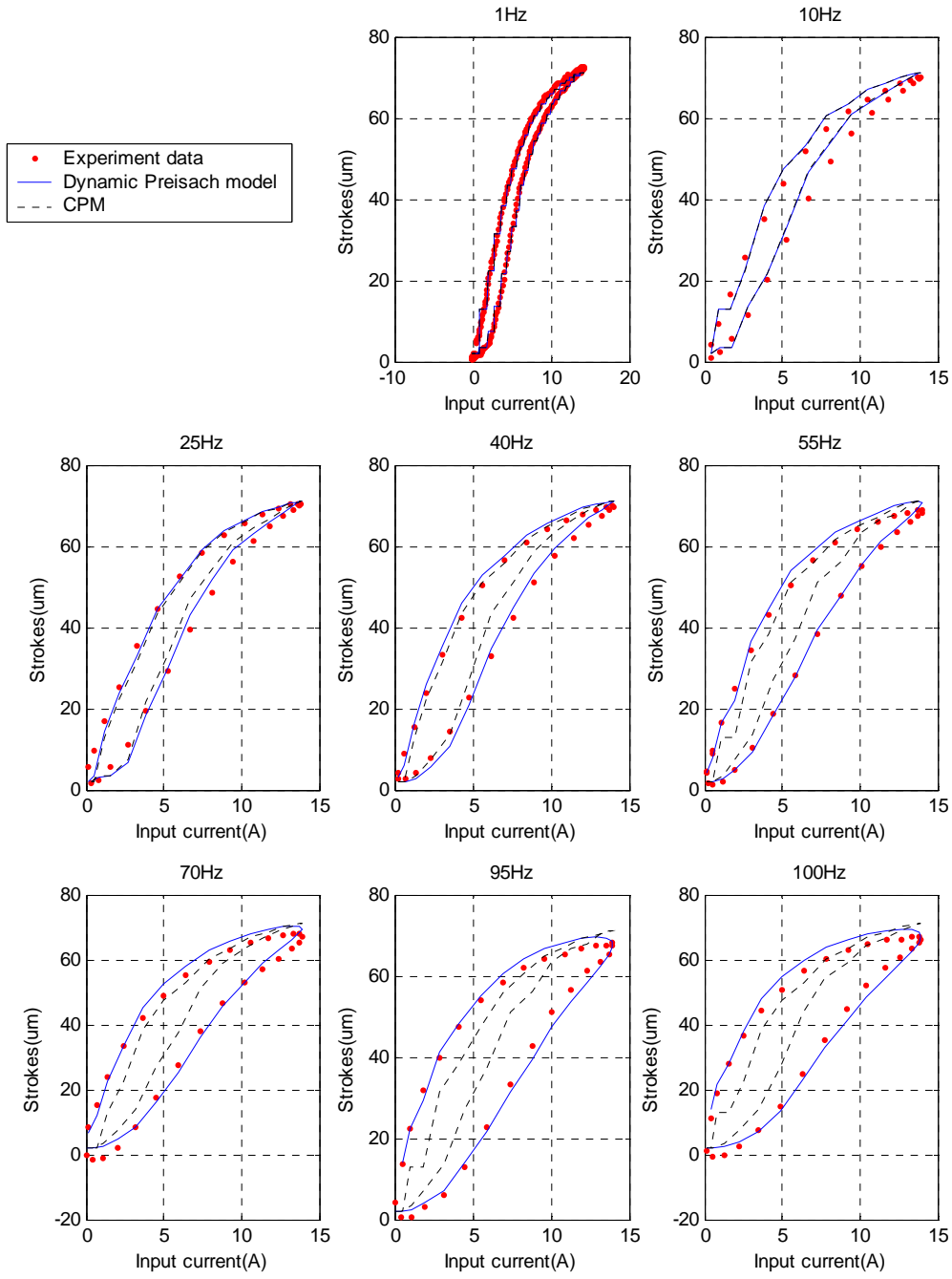


Figure 4-15: Dynamic Modeling Results for test data
(Input-Output Diagram)

Based on the experimental results shown in figures 4-12 to 4-15, the following conclusions can be made:

1. When the frequency is low, the proposed dynamic model performs almost the same as the classical Preisach model.
2. As the working frequency increases, the hysteresis loops observed inside the magnetostrictive actuators become broader. The proposed dynamic model can effectively capture this trend, while the classical Preisach model can not.
3. As shown in the Fig 4-14 and Fig 4-15, the proposed model can also predict exceptionally well the hysteresis loops that are not included in the training process. This proves the rationality and validity of the proposed model.

We also expect that the proposed dynamic hysteresis model can be used to model the hysteresis loops under frequencies higher than 100 Hz. However, for the frequencies higher than 100Hz, the experimental setup used to test the model [Fig 4-1] experiences resonance, causing a spurious frequency response in the data. Since the proposed model can not consider this additional frequency response, this apparatus must be improved before it can be used for further verification of the proposed model.

4.4 Sensitivity of Time Constants

The time constants are the parameters that distinguish the proposed Preisach model from the classical Preisach model. To study the rationality and validity of these newly introduced parameters, additional experiments are performed. In these experiments, the identified time constants are slightly changed around the identified values,

($[\tau_1, \tau_2] = [0.3 \ 0.5 \ 0.7 \ 0.9 \ 1 \ 1.1 \ 1.3 \ 2] * [\hat{\tau}_1, \hat{\tau}_2]$ are explored, where $\hat{\tau}_1, \hat{\tau}_2$ are identified time constants) and the model's output corresponding to these variations under 90 Hz input are obtained (Fig 4-16) and compared with the real experiment data (Fig 4-17). From Fig 4-16 we can clearly see that the model achieves its best performance at the identified time constants and a small variation around these identified values will cause a small decrease in the performance (The same conclusion can be made for frequencies other than 90 Hz). This not only shows the effectiveness of the proposed identification method, but also tells us that the model performance is continuous with respect to time constants, which makes the proposed model robust to small noise of disturbance.

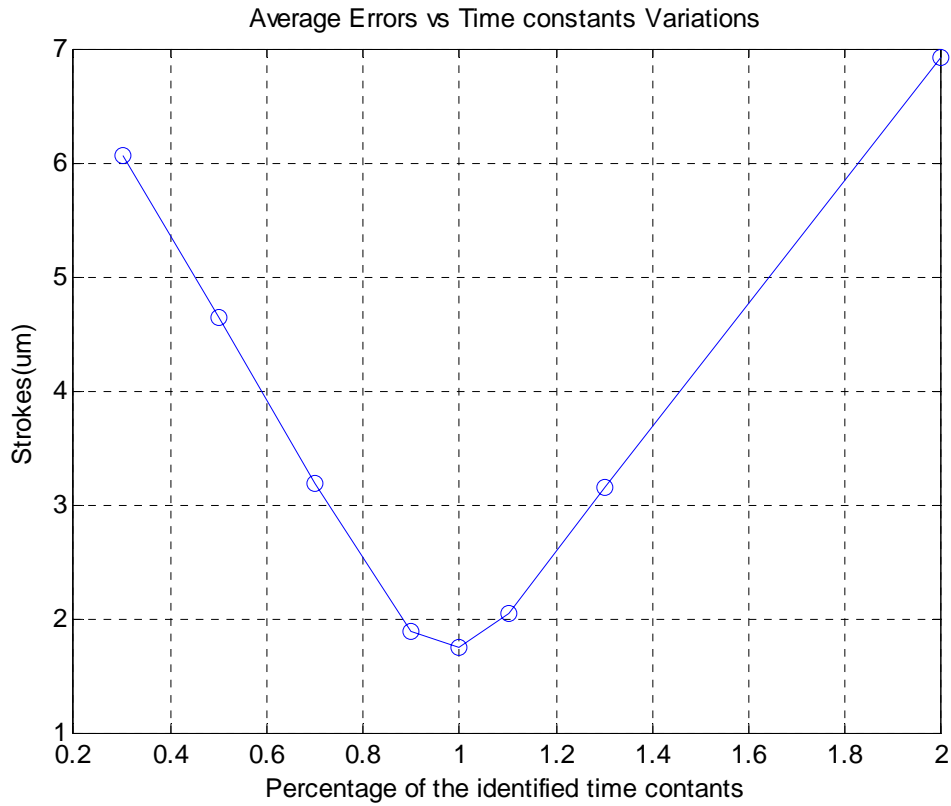


Figure 4-16: Model Performance under Different Time Constants Variations

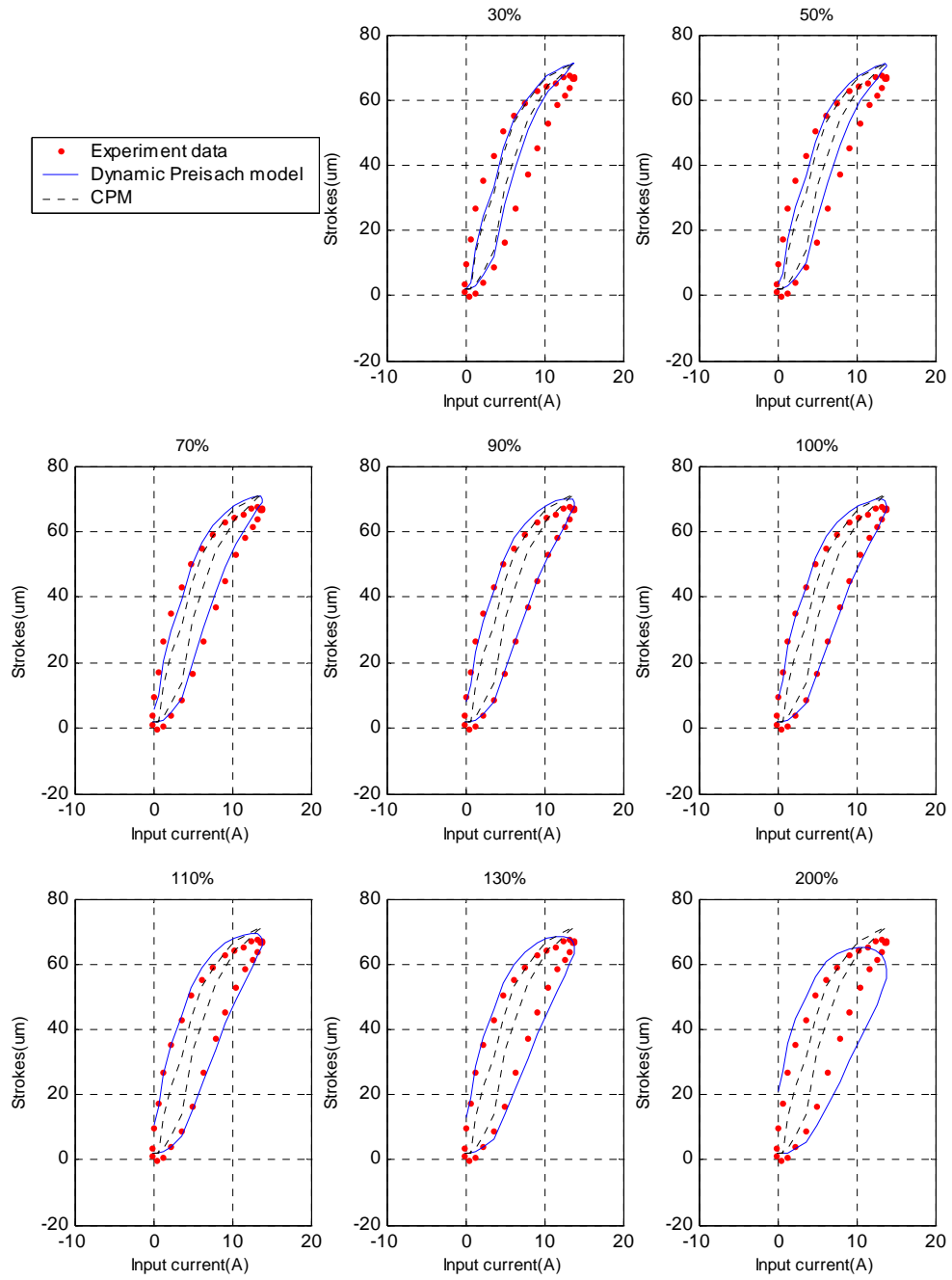


Figure 4-17: Hysteresis Loops under Different Time Constant Variations

CHAPTER 5

CONTROL OF HYSTERETIC SYSTEMS

5.1 Objectives and Preparation

Objectives:

In this chapter, we will discuss the control of hysteretic systems with the proposed dynamic Preisach model representation. The easiest way to control a hysteretic system is to use an inverse control. In [2] and [43], Tan used the discrete Preisach operator to model the rate-independent hysteresis of magnetostrictive actuators. The regulation and tracking problems were solved by inverting the Preisach model. Although an acceptable performance was obtained by using the sophisticated Preisach model, the open loop nature of this kind of inverse control limits its accuracy. Hence, various feedback control schemes have been proposed, and most of them use a hysteretic compensator to reduce or remove the hysteresis behavior and then design a linear controller supposing no hysteresis remains. In a recent study, a pseudo-compensator, which used another Preisach operator to fit the data gathered from the original Preisach model, with input and output swapped, was utilized to compensate for magnetostrictive actuator hysteresis [37]. A PID controller was then designed based only on the linear part of the dynamics. In another study [5], Cruz-Hernandez and Hayward used a phaser to compensate the rate-independent hysteresis. The method is effective and easy to implement. As an uncertain system, the hysteretic system may have a better performance with robust controllers [41, 42], which can accommodate model uncertainties. However, the design methods typically

require bounding of the model uncertainties and the inversion errors, which can only be obtained as rough estimates, resulting in conservative controls. As another approach, adaptive control can also be used to improve robustness [38,39,40]. Unfortunately, most of the proposed methods use simplified hysteresis models (such as the piecewise linear model or even just the relay model) to calculate the corresponding inverse models on-line and develop the parameter update laws.

General drawbacks of current hysteresis control schemes are: (1) Most of them are based on a rate-independent hysteresis model [37, 38, 39, 40], while most applications need to consider dynamic hysteresis behavior. (2) Only the major hysteresis loop is considered [37, 38, 39, 40], which is not sufficient for magnetostrictive actuators because they demonstrate strong hysteresis behavior. (3) Most hysteresis control is based on inverse compensation ([43], [45]). Due to its open loop nature, it suffers from steady state errors and cannot handle noise and disturbances. (4) A common close-loop scheme of a hysteresis control system is to append an inversion hysteretic operator to a linear feedback controller (robust controller [41] or PID controller [37]). It can guarantee zero steady-state error, but has poor tracking performance, especially for higher frequencies.

So the problems here are twofold. Firstly, an accurate dynamic model is needed to handle the strong hysteresis in magnetostrictive actuators. The proposed dynamic Preisach model in this dissertation can accurately predict the major and minor hysteresis loops over a large frequency range, and thus is a good choice for the model. Secondly, a feedback control scheme that is good at both regulating and tracking should be developed. Even if all the static hysteresis can be cancelled out by the inverse operator, the lag effect of dynamic hysteretic system will make the tracking error significant. Thus a traditional

linear controller must be designed or tuned to be very aggressive in order to obtain a good tracking performance. However, this will result in worse regulating performance. This tradeoff is the topic to be explored in this chapter. A well designed controller that is good at both tracking and regulating is developed and tested through simulation.

Preparation

Due to the equipment limitations discussed previously, the control experiments can not be performed on the real actuator system. Thus the controller can only be tested through simulation. During the simulation, a model that represents the real actuator is needed. The model for this purpose should not only demonstrate the dynamic hysteresis behavior, but also be continuous with respect to the input (otherwise, the model may not generate certain reference values). However, only the discrete version of the proposed dynamic Preisach model is implemented in this dissertation. Although it can be modified to be continuous, it is more appropriate to use another dynamic hysteresis model as the real actuator in the simulation and start to model and design a controller for this “virtual actuator” as if its behavior were unknown. Since a model structure, which is different from the one we proposed, is employed as the actuator, the simulation results will be more convincing.

For the above consideration, the following procedure is performed before the control simulation:

1. A dynamic hysteresis model (referred as Model A) in [2] is implemented and used as the virtual actuator during our simulation. Its model details are supposed to be unknown to us. We can only observe

its behavior through input output data. This model has dynamic hysteresis behavior as shown in Figure 5-1.

2. The same identification experiments as in chapter 4 are performed on this virtual actuator (Model A) to obtain of a dynamic Preisach model (referred to as Model B). The static and dynamic identification results are shown in Figure 5-2 and Figure 5-3, respectively. (These identification results can again prove the capability of the proposed dynamic Preisach model in capturing dynamic hysteresis behavior).
3. The controller is designed totally based on the identified dynamic Preisach model (Model B) and is tested through Model A.

5.2 Inverse Control

Most effective control schemes for hysteretic systems will employ, in some way, the inversion of the hysteresis model, thus let's first talk about the inversion algorithm of the proposed dynamic Preisach model (Model B).

As discussed in Chapter 3, the output of the proposed model is a superposition of the responses of all the first-order dynamic relays. It is inherently a nonlinear dynamic system. The output is thus not uniquely determined by the current input; it also depends on time and the state of the system (state of the first-order relay). Thus the term 'inversion' here dose not have the same meaning as in the static one-to-one mapping function. If we express the proposed dynamic Preisach model as an operator $\Gamma(\cdot)$ that is defined as:

$$y = \Gamma(u(t), \xi(t), t) \quad (5.1)$$

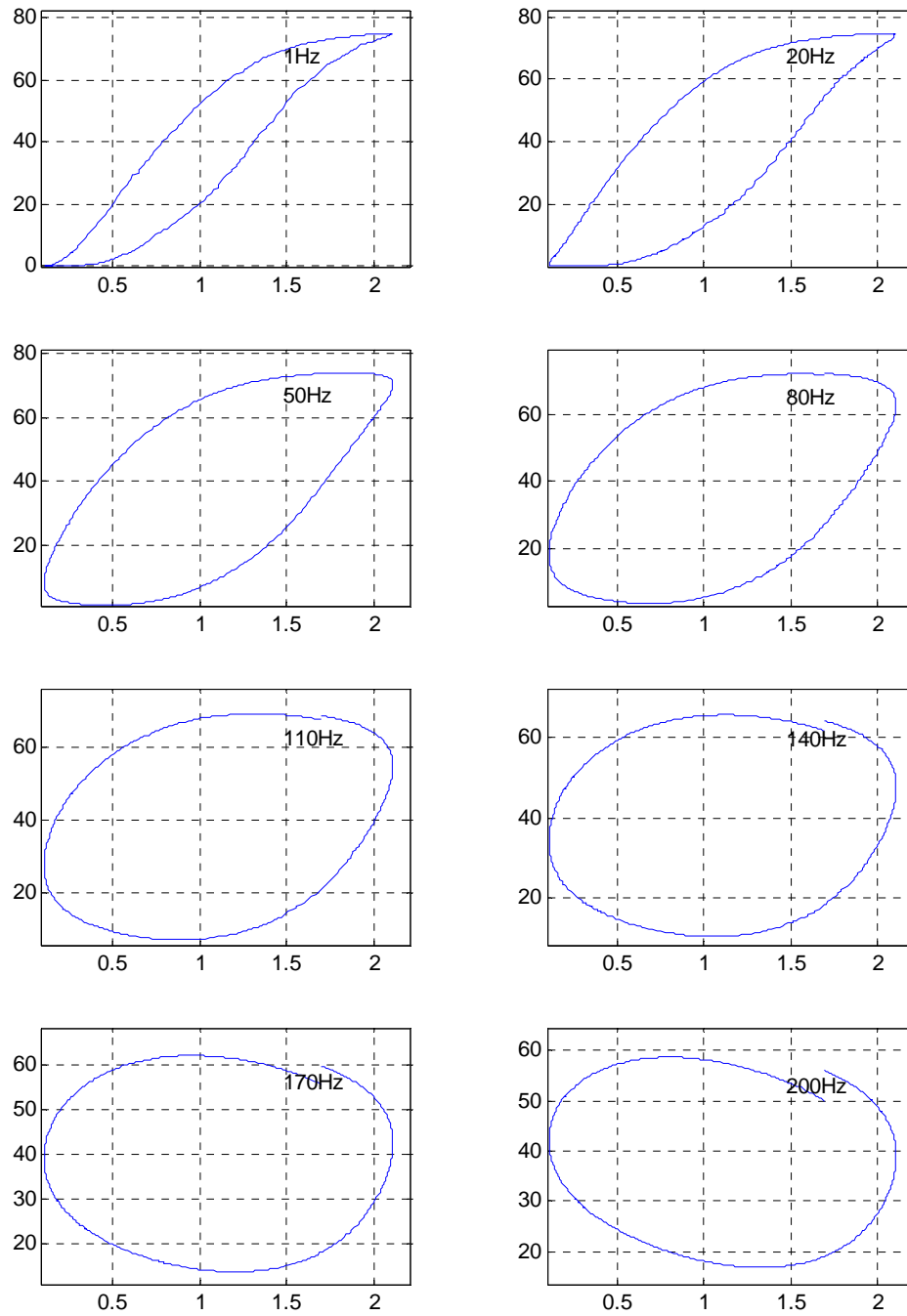


Figure 5-1: Dynamic hysteresis loops of Virtual Actuator (Model A)

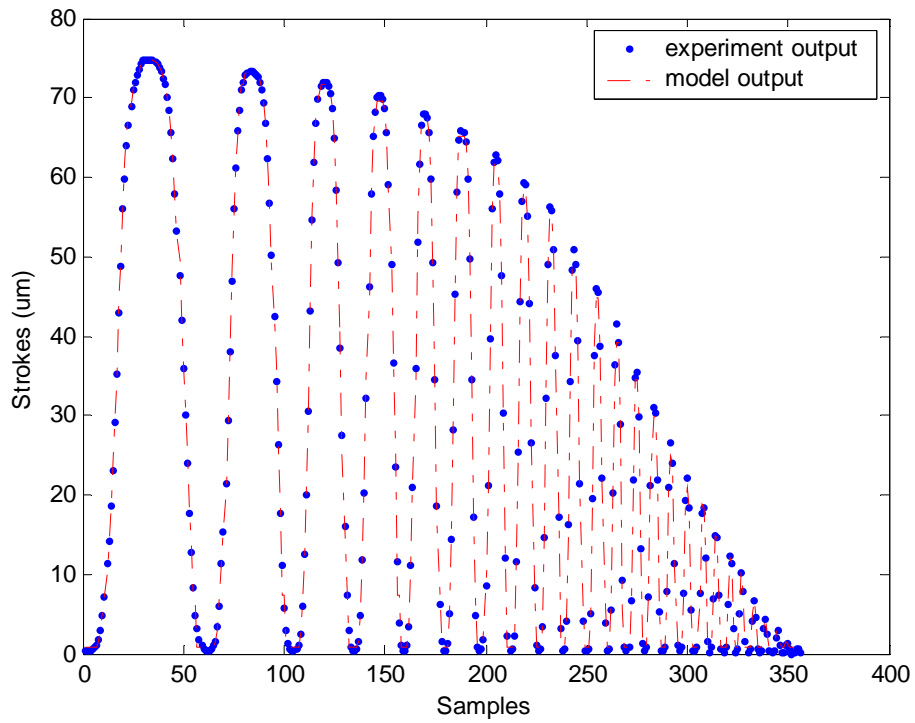


Figure 5-2: Static Modeling Results for Virtual Actuator (Identification Data)

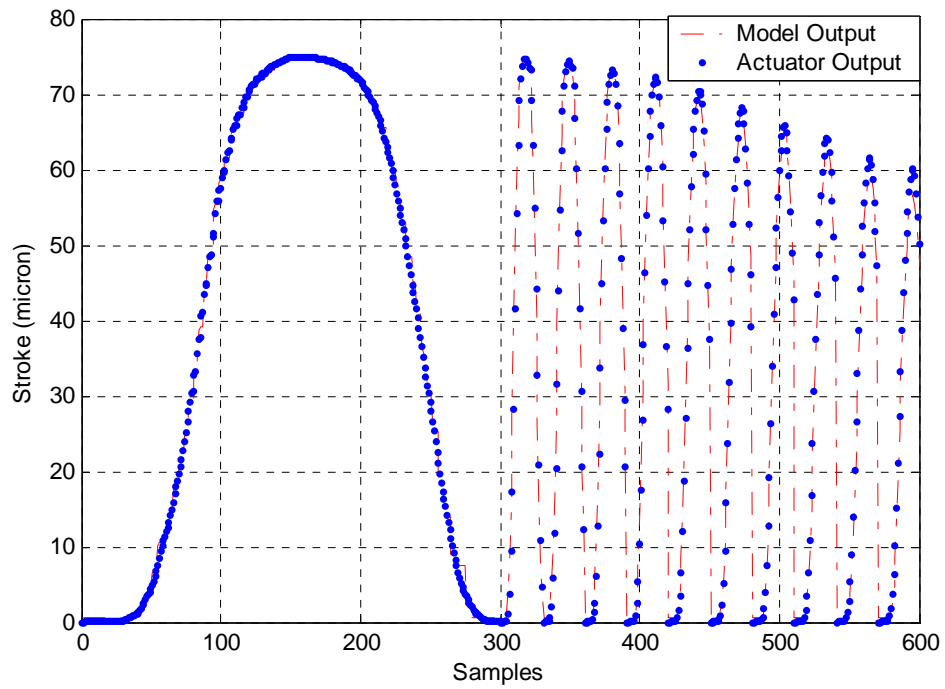


Figure 5-3: Dynamic Modeling Results for Virtual Actuator (Identification Data)
(Frequencies for each cycle(left to right):1Hz, 20Hz, 40Hz,...,200Hz)

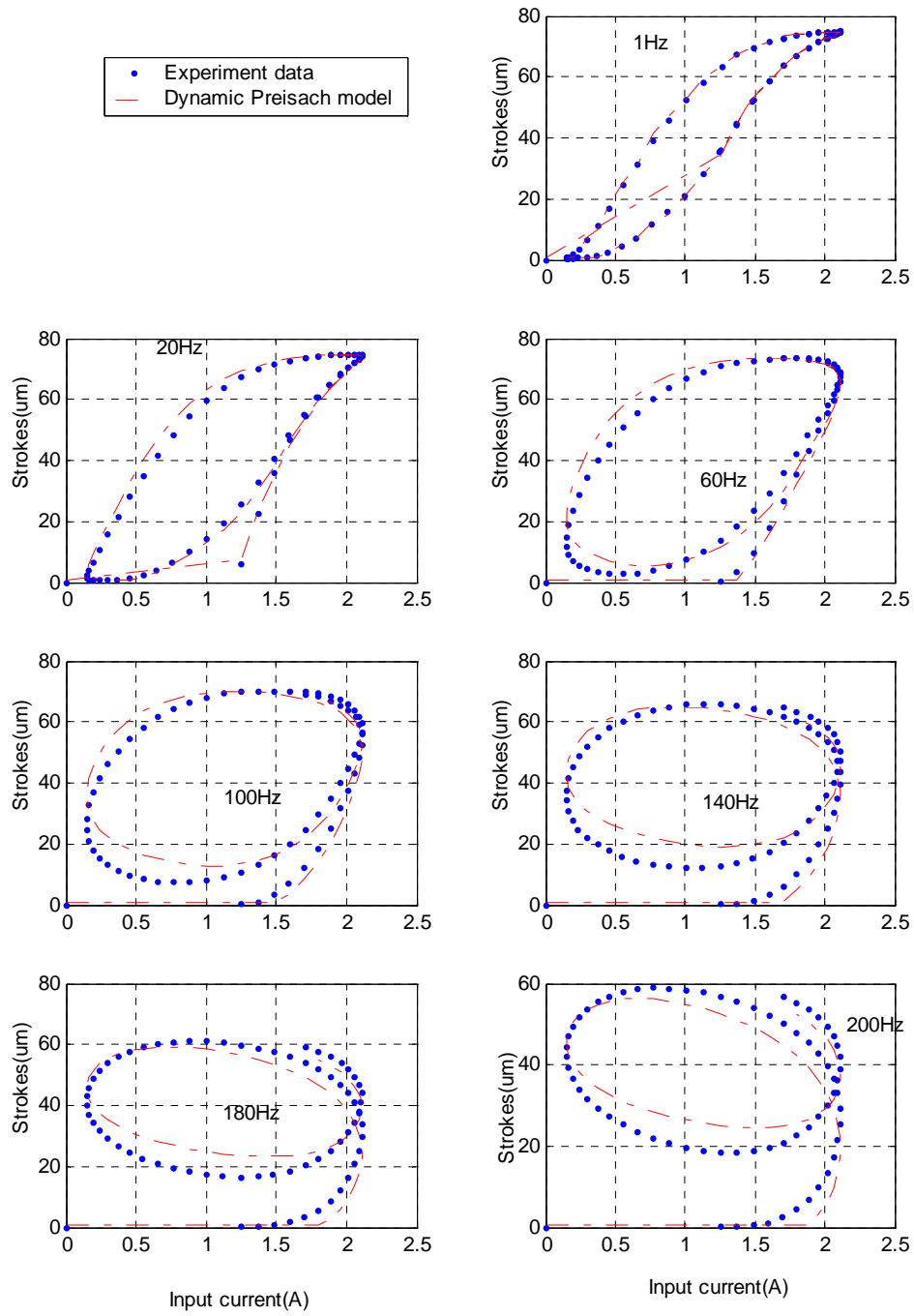


Figure 5-4: Dynamic Modeling Results for Virtual Actuator (Test Data)
(Input-Output Diagrams)

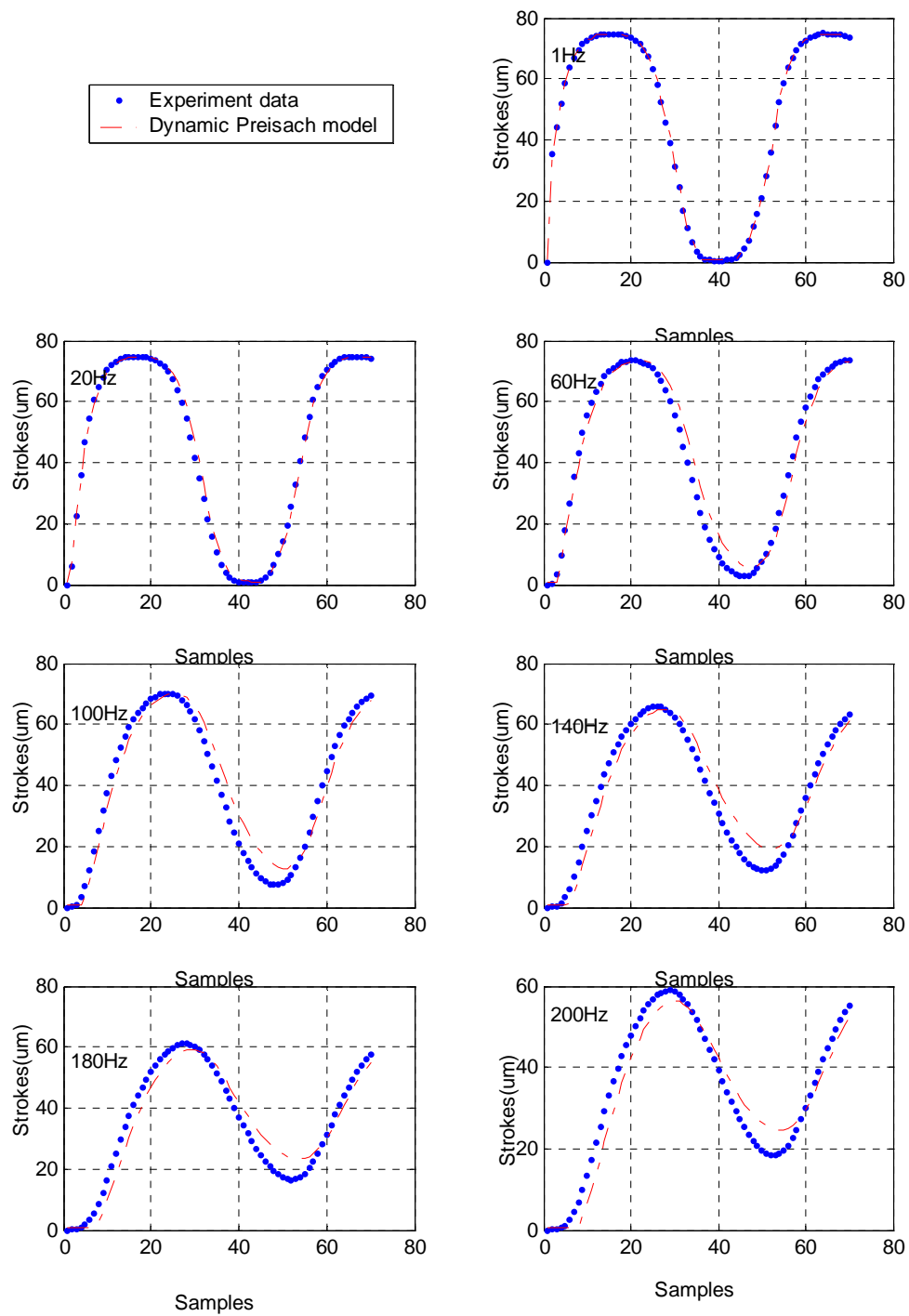


Figure 5-5: Dynamic Modeling Results for Virtual Actuator (Test Data)
(Output)

The inversion of the operator is defined as follows:

Definition 5.1

At time t_0 , for a given output value y^* , the inverse mapping u^* of the dynamic operator $\Gamma(\cdot)$ is defined as:

$$u^* = \arg \min_{u^* \in D_u} \{ \lim_{t \rightarrow \infty} |y(t) - y^*| \} \quad (5.2)$$

where $y(t) = \Gamma(u(t), \xi(t), t)$, and $u(t) = u^*$ for all $t > t_0$.

Based on this definition, if u^* is the inverse value of output y^* , we apply the input u^* at time t_0 and do not change it, the output will finally reach y^* (if y^* is in the range of the operator $\Gamma(\cdot)$) or is the closest value (in the range of the model) to y^* .

With the above idea in mind, it is not difficult to develop a numerical algorithm to solve the inversion for a given output y^* . The following is the description of such an algorithm:

Algorithm 5.1

```

up = max input
down = min input
order = (settling time ts)/(sampling time T)
while ((Up - Down) > tol)
    u = (up + down)/2
    state = current state
    for i = 1 : order
        state = UpdateState(state, u, t);
        y = Output (state, u, t);
    end
    If y > y*, up = u
    else down = u
end

```

Note: The above algorithm requires the output of the model to be continuous with respect to the input. However, as mentioned before, only the discrete dynamic Preisach model has been used through this dissertation. Thus some modification needs to be made on Algorithm 5.1 to make it work for a discontinuous model. Suppose the input range of the model is discretized into L levels by m_1, m_2, \dots, m_L , algorithm 5.2 can be used for the discrete version of the proposed model.

Algorithm 5.2

```

up = L
down = 1
order = (settling time ts)/(sampling time T)
while ((Up - Down) > 1)
    i = (up + down)/2
    u = mi
    state = current state
    for j = 1 : order
        state = UpdateState(state, u, t);
        y = Output (state, u, t);
    end
    If y > y*, up = i
    else down = j
end
y1 = steady state output by mdown
y2 = steady state output under mup

$$u = (y^* - y1)/(y2 - y1) * (m_{up} - m_{down}) + m_{down}$$


```

Since the discrete model can only provide several discrete output values, for an specified output y^* , the best we can know based on the model is that the input should be between m_{up} and m_{down} . However, if we apply m_{up} or m_{down} to the real system, the

actual output will be much different from y^* . Hence, the last three steps in algorithm 5.2 are to interpolate between m_{up} and m_{down} to reduce the inversion error.

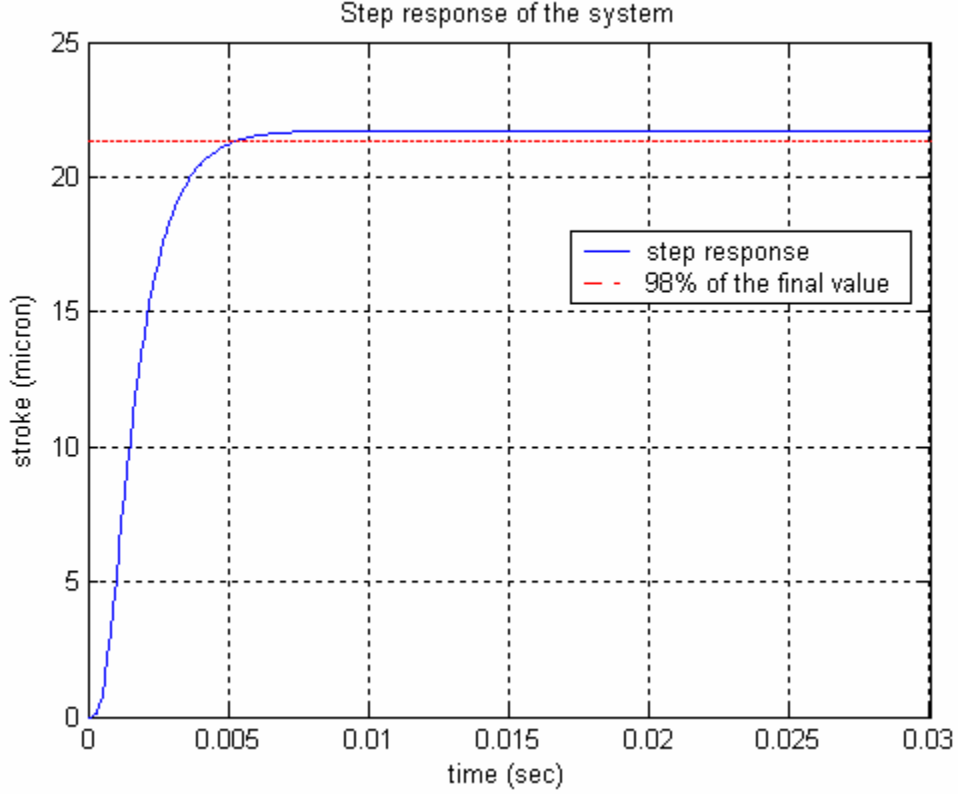


Figure 5-6: Settling Time Estimation of Virtual Actuator (Model A)

The variable ‘order’ in the algorithm is to determine after how many sampling steps the system reaches steady state. Its value depends on the sampling time T and the settling time t_s . The sampling time t_s is defined as the time required for the response curve to reach and stay within 2% of the final value. For the virtual actuator (Model A), t_s is estimated to be 0.005 second through experiment (Figure 5-6).

The inversion algorithm itself can be used as a controller. Given a reference value, it can compute the appropriate input that can drive the output to follow the reference. Due to its open-loop nature, this kind of control usually has steady state error.

But for some applications, a small error is allowed. The setup of such an inversion hysteresis control system and its performance on displacement control are shown in Figure 5-7 and Figure 5-8.

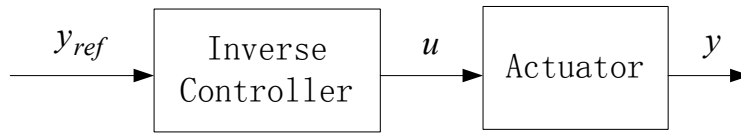


Figure 5-7: Inverse Control System

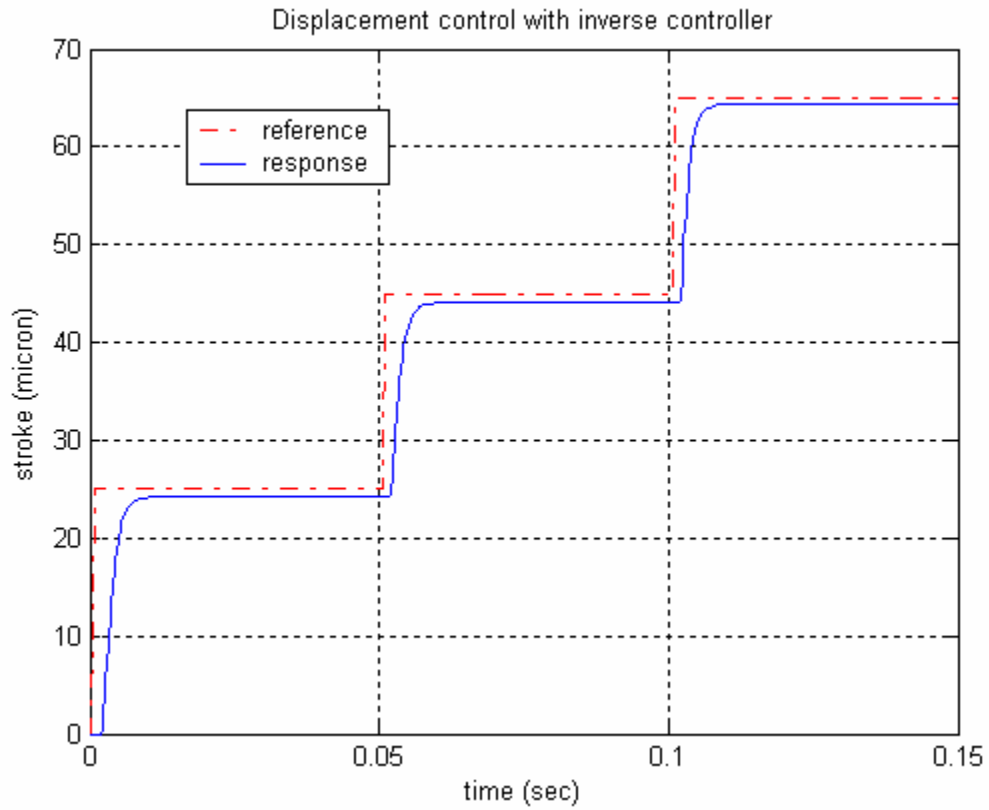


Figure 5-8: Displacement Control with Inverse Controller Only

5.3 PID Control with Inverse Hysteresis Compensation

Figure 5-8 shows that the inverse controller cannot avoid steady state error. This may be unacceptable for some applications. To solve this problem, feedback information must be employed. The simplest feedback control scheme that can guarantee zero steady state error is the PID controller. By adjusting its parameters K_p , K_i , and K_d , a good-performance, stable system can be obtained. However, a PID controller is linear in nature, thus if it is used to control a highly nonlinear dynamic system such as the hysteresis system, the parameters must be tuned to be very conservative in order to maintain system stability. Although zero steady state error can also be guaranteed, the system response will be very slow. To improve its performance, an inverse hysteresis operator is appended after the PID controller to cancel out the hysteresis nonlinearity as shown in Figure 5-9.

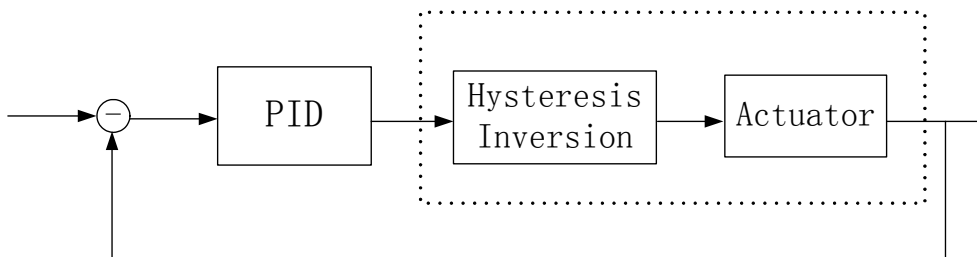


Figure 5-9: PID+Inversion Control Scheme

In practice, the model could not be an exact description of the actuator, and the inverse operator will also produce some error. Hence the hysteresis nonlinearity cannot be totally cancelled out. However, usually the model and its inversion are accurate enough to give a good reduction of the hysteresis nonlinearity. Thus the inverse

hysteresis operator added to the system can significantly ease the design of the PID controller and improve the system performance.

The “differential” term is the most problematic part of the PID controller. It may amplify the noise and cause high frequency oscillation. Hence, during our design the differential term is set to be zero and the PID controller becomes a PI controller.

For a computer control system, the PI controller should be implemented through programming. The output of the PI controller in our system is programmed based on the following iterative formula:

$$\begin{aligned} u(k) &= K_p e(k) + K_i \sum_{i=0}^k e(i) \\ &= u(k-1) + (K_p + K_i)e(k) - K_p e(k-1) \end{aligned} \quad (5.3)$$

Our job now is to obtain an appropriate set of K_p K_i that can make the system perform as well as possible. There are many ways to calculate the parameters of the PID controller. However, they are all application dependent. Thus we adopt the trial-and-error method to practically tune the PI parameters for our hysteresis control system. A good control performance can be obtained through the following PI parameters and sampling time:

$$K_p = 0.78, K_i = 0.058, T = 0.0001$$

The designed PID controller with inverse hysteresis compensation is first tested through a displacement control experiment. The result is shown in Figure 5-10. It can be clearly seen that the PID controller with inverse compensation can produce zero steady state error as well as a fast response.

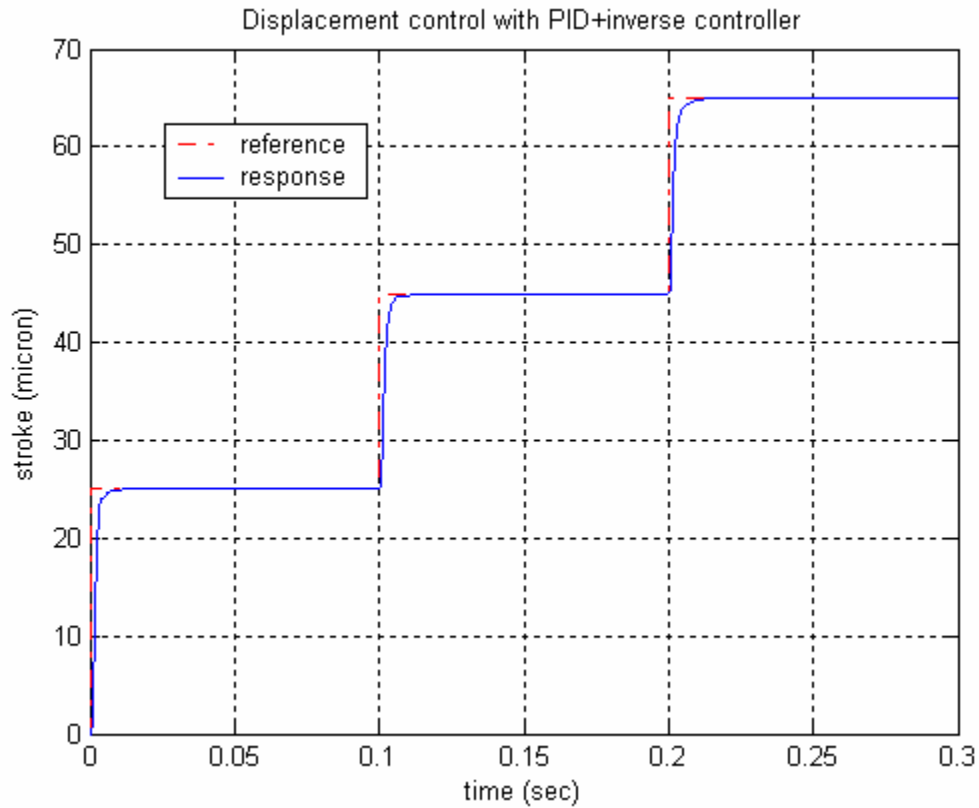


Figure 5-10: Displacement Control Results with PID+inverse Controller

To make the simulation more realistic, random noise (between -1 to +1 micron) is added to the system to simulate the effect of measurement noise. The new system setup is illustrated in Figure 5-11 and its regulating and tracking performance is shown in Figures 5-12 to 5-14.

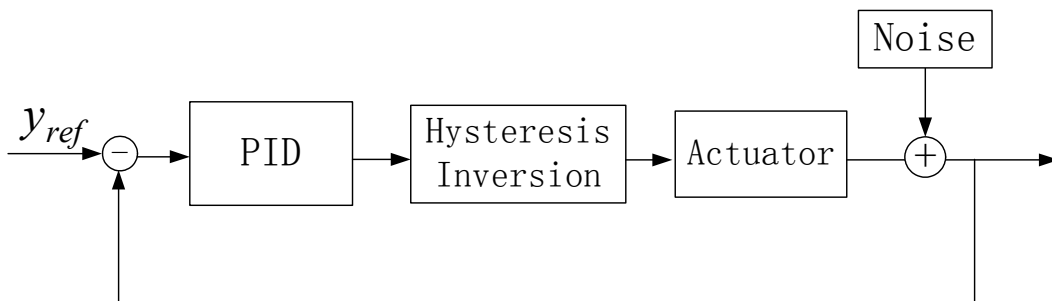


Figure 5-11: PID+Inverse Control with Noisy Measurement

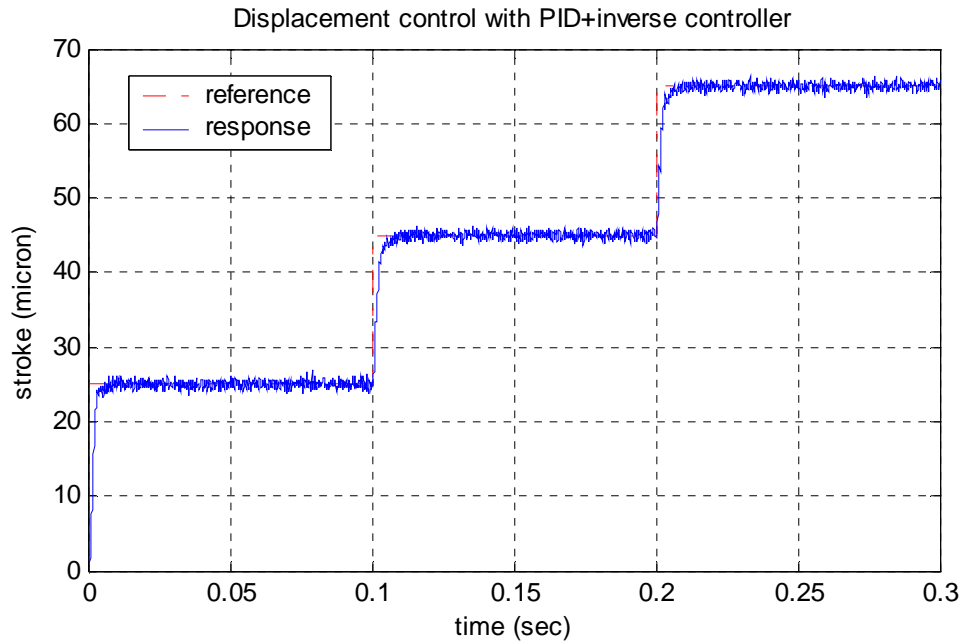


Figure 5-12: Displacement Control with PID+Inverse Controller and Noisy Measurement

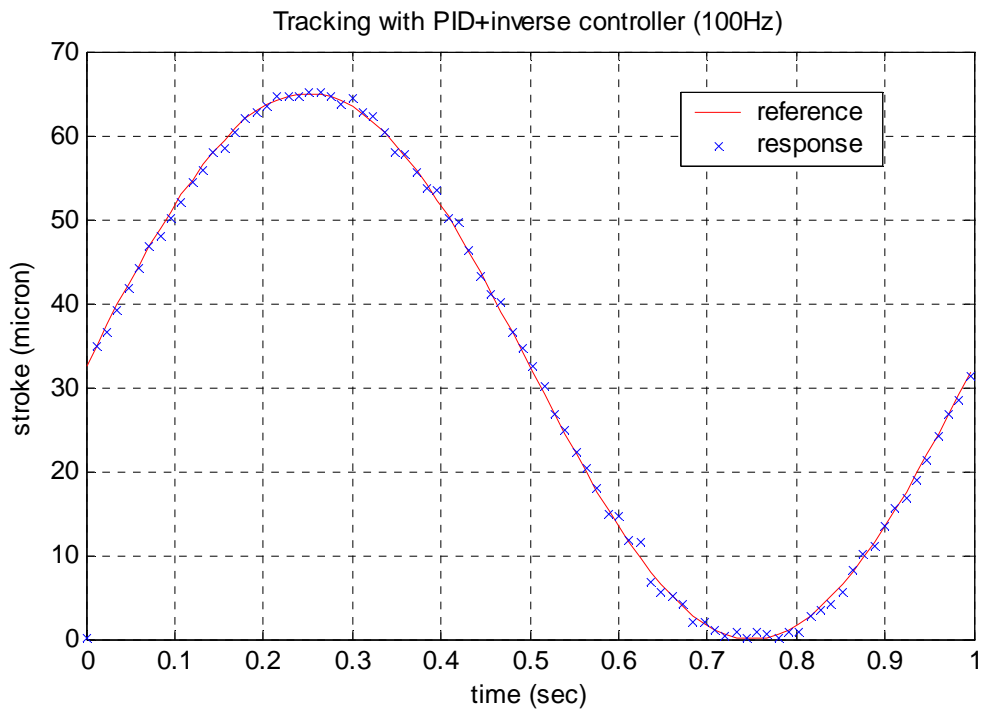


Figure 5-13: 1 Hz Tracking Result with PID+Inverse Controller and Noisy Measurement
(The signals have been resampled for display purposes)

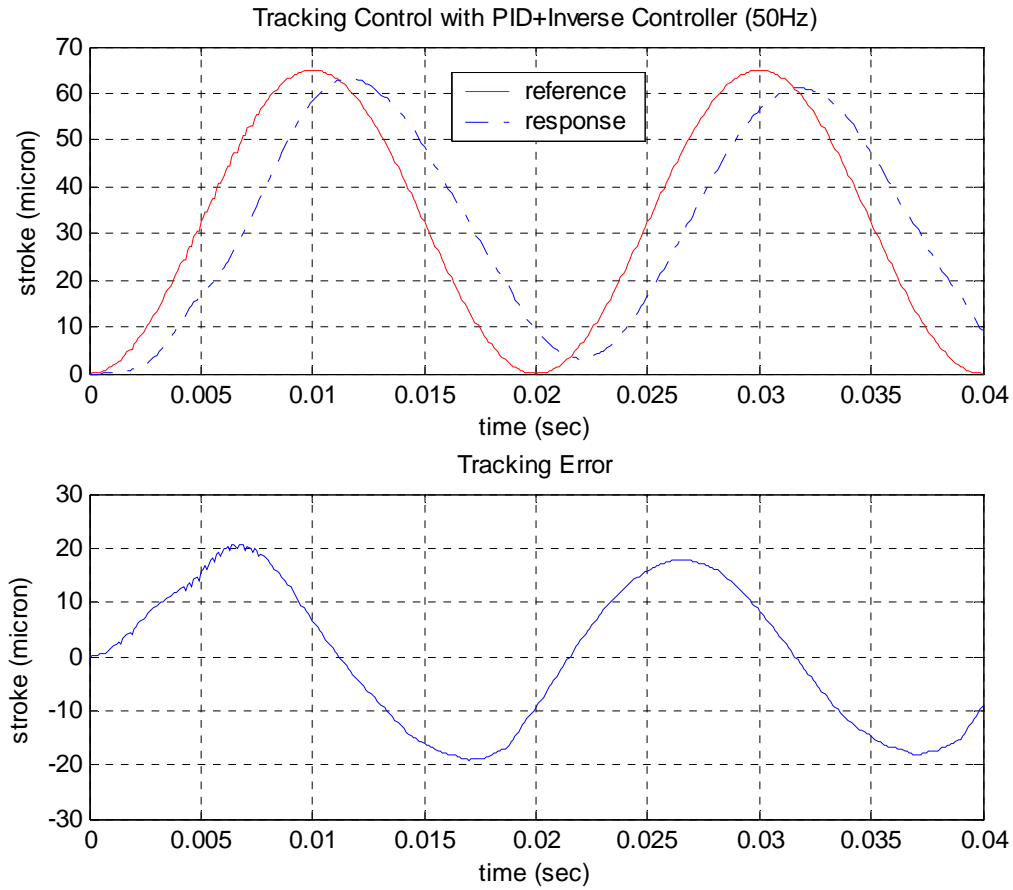


Figure 5-14: Tracking Results with PID+Inverse Controller (50Hz)

As shown in Figure 5-12, even with the measurement noise, the PID controller with inverse compensation can also produce accurate displacements. However, its tracking capability is limited, especially for high frequencies. The reason for this is that the controller is always trying to make the output follow the present reference, and the reference keeps changing. Hence the controlled output will always lag behind the reference, which will cause a big tracking error. To solve this problem, the controller should be able to take future reference into account when making the control decision. Thus, we need a prediction-based controller.

5.4 Predictive Control

5.4.1 Nonlinear Predictive Control [46,47,48]

A well-known class of nonlinear controllers that directly uses the nonlinear model is model predictive controllers (MPC) [47]. Linear MPC is a discrete time controller that calculates the present control, at each sampling time, by predicting over a horizon p the process response to changes in control. The change in control that is within specified constraints and that gives the most desirable process response is then implemented.

Nonlinear MPC is similar and is constructed as solving an on-line finite horizon open-loop optimal control problem subject to system dynamics and constraints involving states and controls. Based on measurements obtained at time t , the controller predicts the future dynamic behavior of the system over a prediction horizon T_p and determines (over a control horizon $T_c \leq T_p$) the input such that a predetermined open-loop performance objective functional is optimized.

Due to disturbances and model-plant mismatch, the true system behavior is different from the predicted behavior. In order to incorporate some feedback mechanism, the open-loop manipulated input function obtained will be implemented only until the next measurement becomes available. If there were no disturbances and no model-plant mismatch, and if the optimization problem could be solved for infinite horizons, then one could apply the input function found at time $t = 0$ to the system for all times $t \geq 0$. However, this is not possible in general.

The time difference between the recalculation/measurements can vary, however often it is assumed to be fixed, i.e. the measurement will take place every δ sampling

time-units. Using the new measurement at time $t + \delta$, the whole procedure – prediction and optimization – is repeated to find a new input function with the control and prediction horizons moving forward.

Consider the stabilization problem for a class of systems described by the following nonlinear set of differential equations.

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \quad \mathbf{x}(0) = \mathbf{x}_0 \quad (5.4)$$

Subject to input and state constraints of the form:

$$\mathbf{u}(t) \in U, \forall t \geq 0 \quad \mathbf{x}(t) \in X, \forall t \geq 0, \quad (5.5)$$

$$\begin{aligned} U &:= \left\{ \mathbf{u} \in \mathbb{R}^m \mid \mathbf{u}_{\min} \leq \mathbf{u} \leq \mathbf{u}_{\max} \right\}, \\ X &:= \left\{ \mathbf{x} \in \mathbb{R}^n \mid \mathbf{x}_{\min} \leq \mathbf{x} \leq \mathbf{x}_{\max} \right\}. \end{aligned} \quad (5.6)$$

Assumption 1: $U \subset \mathbb{R}^p$ is compact, $X \subseteq \mathbb{R}^n$ is connected and $(0,0) \in X \times U$.

Assumption 2: The vector field $\mathbf{f}: \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ is continuous and satisfies $\mathbf{f}(\mathbf{0}, \mathbf{0}) = \mathbf{0}$.

In addition, it is locally Lipschitz continuous in \mathbf{x} .

Assumption 3: The system (Equation 2-18) has an unique continuous solution for any initial condition in the region of interest and any piecewise continuous and right continuous input function $\mathbf{u}(\cdot): [0, T_p] \rightarrow U$.

Usually, the finite horizon open-loop optimal control problem described above is mathematically formulated as follows: (internal controller variables are denoted by a bar)

Find

$$\min_{\bar{\mathbf{u}}(\cdot)} J(\mathbf{x}(t), \bar{\mathbf{u}}(\cdot); T_c T_p) \quad (5.7)$$

with

$$J(\mathbf{x}(t), \bar{\mathbf{u}}(\cdot); T_c T_p) := \int_t^{t+T_p} F(\bar{\mathbf{x}}(\tau), \bar{\mathbf{u}}(\tau)) d\tau \quad (5.8)$$

subject to:

$$\begin{aligned} \dot{\bar{\mathbf{x}}}(\tau) &= \mathbf{f}(\bar{\mathbf{x}}(\tau), \bar{\mathbf{u}}(\tau)), \quad \bar{\mathbf{x}}(t) = \mathbf{x}(t) \\ \bar{\mathbf{u}}(\tau) &\in U, \quad \forall \tau \in [t, t+T_c] \\ \bar{\mathbf{u}}(\tau) &= \bar{\mathbf{u}}(t+T_c), \quad \forall \tau \in [t+T_c, t+T_p] \\ \bar{\mathbf{x}}(\tau) &\in X, \quad \forall \tau \in [t, t+T_p] \end{aligned} \quad (5.9)$$

where T_p and T_c are the prediction and control horizon, respectively, with $T_c \leq T_p$ and internal controller variables are denoted by a bar.

The function F , called stage cost in the following, specifies the desired control performance that can arise, for example, from economical and ecological considerations. The standard quadratic form is the simplest and most often used:

$$F(\mathbf{x}, \mathbf{u}) = (\mathbf{x} - \mathbf{x}_s)^T Q (\mathbf{x} - \mathbf{x}_s) + (\mathbf{u} - \mathbf{u}_s)^T R (\mathbf{u} - \mathbf{u}_s), \quad (5.10)$$

where \mathbf{x}_s and \mathbf{u}_s denote given setpoints: Q and R denote positive definite, symmetric weighting matrices.

The *closed-loop* control is defined by the optimal solution of Equation 2-20 at the sampling instants:

$$\mathbf{u}^*(\tau) := \bar{\mathbf{u}}^*(\tau; \mathbf{x}(t), T_p, T_c), \tau \in [t, \delta] \quad (5.11)$$

The optimal value of the NMPC open-loop optimal control problem as a function of the state will be denoted in the following as value function:

$$V(\mathbf{x}; T_p, T_c) = J(\mathbf{x}, \bar{\mathbf{u}}^*(\cdot; \mathbf{x}(t)); T_p, T_c). \quad (5.12)$$

The value function plays an important role in the proof of the stability of various NMPC schemes, as it serves as a Lyapunov function candidate.

The disadvantages of nonlinear MPC are primarily due to the finite horizon optimal control problem being non-convex. Non-convexity introduces the questions of how long will the optimization take, whether it will terminate, and is a suboptimal solution acceptable. The finite horizon optimal control problem associated with nonlinear MPC is not guaranteed to be convex and it is difficult to obtain the global optimal solution. Therefore, because of the non-convexity, NMPC formulations need to be derived that guarantee solution feasibility, robustness, and performance despite the solution being sub-optimal. Moreover, for further development of NMPC algorithms, faster optimization solvers need to exploit the inherent structure of the process. For it is possible that in solving the finite horizon optimal control problem one can exploit the specific system dynamics, e.g. Lipschitz continuous, static nonlinearity, input-affine, bilinear, hybrid, piecewise affine, non-holonomic or homogeneous.

5.4.2 Proposed Predictive Control

Most applications of magnetostrictive actuators require the actuators be controlled in real-time. Traditional predictive control involves complicated optimization. This optimization could be finished in real-time for some simple models; however, it may cost much longer time for our Preisach-type actuator model. Due to this reason, a specially designed predictive controller is proposed that considers both tracking performance and implementation issues.

If the prediction horizon is p discrete steps, predictive control is essentially to minimize a predefined cost function by adjusting the control variables of the following p steps. Usually p is bigger than 1 and the optimization over several variables is rather time-consuming. Thus in the proposed method, we suppose all the control variables of the following p steps are equal to each other, and then the optimization become single variable which could be solved in real-time. The corresponding predictive control scheme can be expressed as the following:

Given the actuator model:

$$y_{k+1} = \Gamma(u(k), \xi(k), k) \quad (5.13)$$

and the cost function:

$$J_k = |y_{ref}(k+p) - y(k+p)| \quad (5.14)$$

the control action u_k at time k is:

$$u_k = \underset{u_k, u_{k+1}, \dots, u_{k+p-1}}{\operatorname{argmin}} (J_k), \quad (5.15)$$

$$\text{under the constraint: } u_k = u_{k-1} = \dots = u_{k-p-1} \in [u_{\min}, u_{\max}]$$

If p is set to be the settling time over the sampling time (t_s/T_s), the predictive control defined by equations (5.13) to (5.15) can be called as “step response predictive” control. Because the control action is supposed to not be changed, and is selected such that the corresponding steady state output is equal or close to the reference.

Although it appears quite unacquainted, the above optimization problem is essentially the same as the dynamic inversion problem described in Section 5.2. Thus it can be solved using algorithm 5.1 or algorithm 5.2. What makes the predictive different

from the inverse control scheme is that: (1) it calculates the control action based on the future reference instead of the current reference; (2) it can be easily incorporated with feedback mechanism as described subsequently.

Incorporate feedback information

From (5.13), we have:

$$\begin{aligned}
 y(k+1) - y(k) &= \Gamma(u(k), \xi(k), k) - \Gamma(u(k-1), \xi(k-1), k-1) \\
 y(k+2) - y(k+1) &= \Gamma(u(k), \xi(k+1), k+1) - \Gamma(u(k), \xi(k), k) \\
 y(k+3) - y(k+2) &= \Gamma(u(k), \xi(k+2), k+2) - \Gamma(u(k), \xi(k+1), k+1) \\
 &\vdots \\
 &\vdots \\
 y(k+p) - y(k+p-1) &= \Gamma(u(k), \xi(k+p-1), k+p-1) \\
 &\quad - \Gamma(u(k), \xi(k+p-2), k+p-2)
 \end{aligned} \tag{5.16}$$

So

$$\begin{aligned}
 y(k+p) &= y(k) + \Gamma(u(k), \xi(k+p-1), k+p-1) \\
 &\quad - \Gamma(u(k-1), \xi(k-1), k-1)
 \end{aligned} \tag{5.17}$$

Given a $u(k)$ and known it is will not be changed, all the $\xi(i), i \geq k$ can be calculated, as can $\Gamma(u(k), \xi(k+p-1), k+p-1)$, thus the left side of (5.17) is totally determined by $u(k)$. Hence, $u(k)$ can be numerically solved by trial-and-error (similar to algorithm 5.1 or 5.2) such that $y(k+p) = y_{ref}(k+p)$. Since at current time k , the actual system output $y_f(k)$ is available. Replace $y(k)$ in (5.17) with this feedback information $y_f(k)$ can thus make the predictive control system a close loop system.

$$\begin{aligned}
 y(k+p) &= y_f(k) + \Gamma(u(k), \xi(k+p-1), k+p-1) \\
 &\quad - \Gamma(u(k-1), \xi(k-1), k-1)
 \end{aligned} \tag{5.18}$$

(5.18) tells us that the control action $u(k)$ is always different from $u(k-1)$ unless $y(k+p) = y_{fb}(k)$, which means the system will not have steady state error.

The above closed loop predictive control scheme has been tested through both regulation and tracking simulations (Figure 5-15 and Figure 5-16). It can be seen that the proposed predictive controller not only has a good regulation performance, but also reduces the tracking error to 2 micron (on average), which is almost 20% of the average tracking error of the PID+inverse controller.

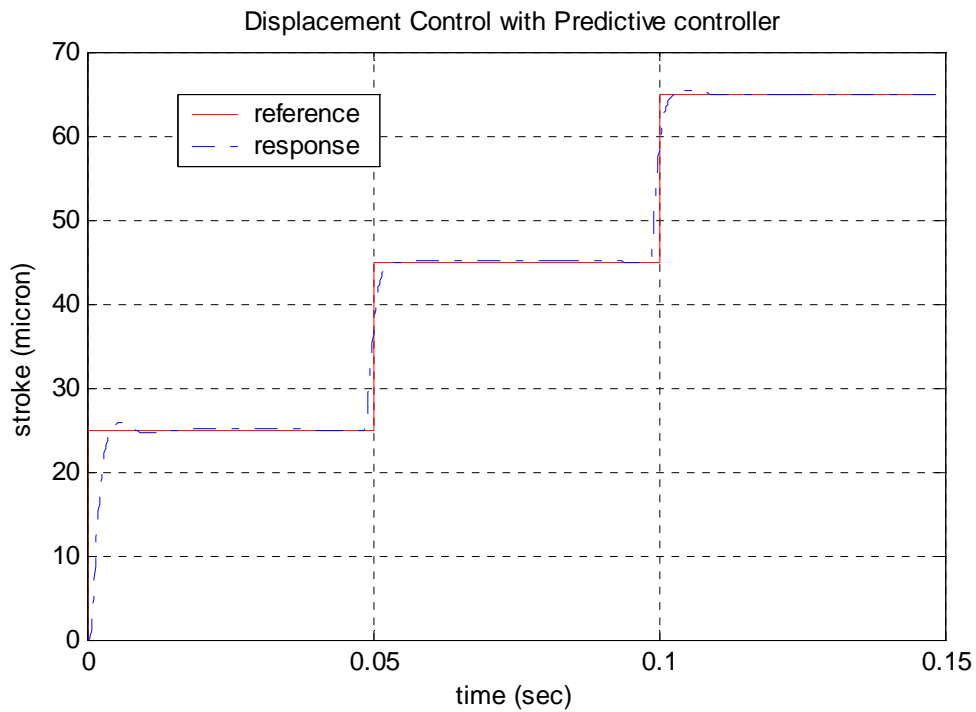


Figure 5-15: Displacement Control with Predictive Controller

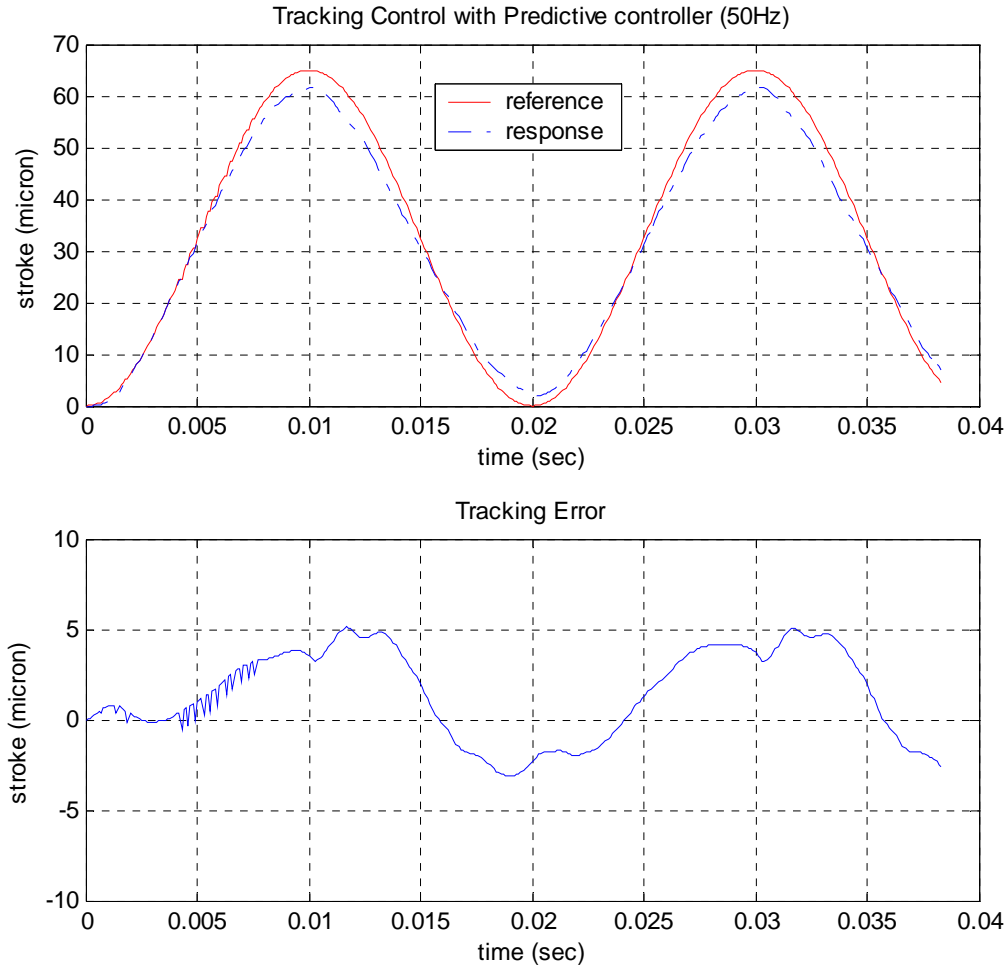


Figure 5-16: Tracking Control with Predictive Controller

Although the predictive controller improves the tracking performance a lot, the error (3 micron) is still significant for some applications. This is because the “Virtual Actuator” used in our simulation has a very slow time response. If we want such a low speed system to accurately follow a fast changing signal, the simplified predictive controller may be inadequate. In this case, a standard predictive control which minimizes the errors of all the future p steps without constant $u(k)$ constraints is more

appropriate. However, since the model is complicated and nonlinear, online optimization is a major problem and will be studied in our future research.

CHAPTER 6

CONCLUSIONS

This project aims at developing an effective model of magnetostrictive actuators for high frequency applications. A novel dynamic relay operator, referred to as a First-order dynamic relay (FDR), has been proposed as the elementary hysteron (kernel) of the modified Preisach model. The resulting model has thus become a rate-dependent model that can be used to model the dynamic magnetostrictive actuators. The newly proposed model has almost the same structure as the classical Preisach model except that it has two additional parameters to describe the dynamic behavior of its kernel (dynamic relay). The two additional parameters are assumed to be rate-independent and do not relate to the weight function of the Preisach model. This assumption allows us to separately identify the weight function (discrete weights) and the additional dynamic parameters. A least squares method has been used to identify the weights of the first-order dynamic relays. A numerical optimization algorithm has been developed to identify the additional dynamic parameters based on the experiment data. Experiment results have shown that the proposed dynamic model preserves the strength of the classical Preisach model in static hysteresis modeling and can effectively captures the dynamic hysteresis behavior in the magnetostrictive actuators. Although the testing experiments are performed based on a magnetostrictive actuator, the proposed model is believed to be a general way to model the dynamic hysteresis in any kind of smart actuators or hysteretic systems. In addition, the controller design using the proposed model has also been discussed. An inversion algorithm has been developed and a PID controller with inverse hysteresis compensation

has been proposed and tested through simulations. The results have shown that the PID controller with inverse compensation is good at regulating control; its tracking performance is really limited, especially for high frequency signals. Hence, a simplified predictive control scheme has been proposed to improve the tracking performance. It has been proved through experiments that the proposed predictive control can reduce the average tracking error to 2 micron while preserve a good regulating performance.

Future research will be focused on further verification of our model for higher frequency data (higher than 100 Hz) and further improvement of the tracking performance for high frequency signals.

Appendix:

I Program manual

Step 1: Data preparation

The raw 1 Hz experiment data should be stored in plain txt files under ‘1hz’ directory. Each text file should just contain the data generated by one amplitude of sine waves, with the magnitude as its file name. For example, you should store the data generated by ‘1.5*sin(t) and 2*sin(t) as ‘1.5A.txt’ and ‘2A.txt’, respectively. There are four columns in each text file, they represent ‘power (v) stroke(v) Current (A) stroke(micron)’, respectively.

The high frequency data (only the ones lower than 100Hz are used by now) have the same format as the 1Hz data. However, they are named by their frequencies, such as ‘30hz.txt’ and stored under ‘hf100’ directory.

After making two new directories ‘1hzsmooth’ and ‘hf100smooth’, we can run ‘smoothdata.m’ and ‘smoothhfddata.m’ to reduce the noise of the experiment data. Make sure the ‘current’ vector in the ‘smoothdata.m’ contains all the amplitudes of the experiment data under ‘1hz’ directory and the ‘frequency’ vector in ‘smoothhfddata.m’ contains all the frequencies under the ‘hf100’ directory. The smoothed data are stored under the two new directories (‘1hzsmooth’ and ‘hf100smooth’) and will be used by the sampling programs.

Make a new directory ‘iden_hf100’, and then run ‘sampledataforhighfrequency.m’ and ‘newsampleddata.m’. The 1hz sampled data for weights identification are stored in ‘iden_data.mat’, while the high frequency data used for the identification of (τ_1, τ_2) are stored frequency-wisely under the ‘iden_hf100’

directory. The sampled data rather than the raw data are actually used in the identification programs.

Step 2: Training the model

After generating the sampled data, we can start the identification by running 'iden_xls.m' and then 'iden_dynamic.m'. The identified weights are store in 'xls_weights.mat' and the resulting (τ_1, τ_2) are stored in 'tc.mat'.

II Codes

preparedataforexcel.m

```
%prepare data for excel data
clear all;

global weight weight0 ref I_range D_range T tao delay f u_range output data magnitudes level
index mag

global current iden_current frequency findex
current=[0 1.5 2 2.3 3.5 4 4.5 5 5.5 6 6.5 7 7.5 8 8.5 9 9.5 10 10.5 11 11.5 12 12.5 13 13.5 14];
%first should be 0,
load 'mag.mat';
iden_current= sort(mag);
frequency=[1 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100];
findex=length(current); %magnitude index for high frequency data current(findex)=14A;
                    %it is useful when process the data
index=length(current); %index determines which magnitude is the maximum magnitude for the
modeling
level=length(iden_current)-1; %levels determines how many intervals does the entire range have.
T=0.0001;
f=100;
delay=0.0004;
tao=0.0001;
I_range=[0 current(index)];
u_range=I_range;
interval=abs(I_range(1)-I_range(2))/level;
tt=1;
for i=1:level
    for j=1:i
        ref(tt,1)=iden_current(j)+(iden_current(j+1)-iden_current(j))/2;
        ref(tt,2)=iden_current(i)+(iden_current(i+1)-iden_current(i))/2;
        tt=tt+1;
    end
end
end
```

smoothdata.m

```
%smooth the new data
current=[0 1.5 2 3 3.5 4 4.5 5 5.5 6 6.5 7 7.5 8 8.5 9 9.5 10 10.5 11 11.5 12 12.5 13 13.5 14];
mag(1)=0;
for i=2:length(current)
    i
    clear data;
    clear cur;
    clear dis;
    ss=['1hz\' num2str(current(i)) 'A.txt'];
    [tmp1 tmp2 cur dis]=textread(ss);
    for j=1:20
        cur=smooth(cur);
        dis=smooth(dis);
    end
    data(:,1)=smooth(cur);
    data(:,2)=smooth(dis);
    mag(i)=max(data(:,1));
    %ss=sprintf('1hzsmooth\\\'%.1f.mat',current(i));
    ss=['1hzsmooth\' num2str(mag(i)) '.mat'];
    save(ss,'data');
end
save 'mag.mat' mag;
```

smoothhfdata.m


```

frequency=[1 10 30 60 90 120 150 180 210 240 270 300 330 360 390 420 450 480 510 540 570
600];
for i=1:length(frequency)
    n=frequency(i)
    clear data;
    ss=['hf100\' num2str(n) 'Hz.txt'];
    [tmp1 tmp2 cur dis]=textread(ss);
    data(:,1)=smooth(cur);
    data(:,2)=smooth(dis);
    ss=['hf100smooth\' num2str(n) '.mat'];
    save(ss,'data');
end

```

```

*****
*****

```

newsamplingdata.m

```

%sample the data
global I_range level current iden_current
pp=1; %for realdata
st=1; %for tmpdata
clear data;
clear realdata;
clear tmpdata;
pp=1;
for index=length(iden_current):-1:2
    per=1;
    ss=['1hzsmooth\' num2str(iden_current(index)) '.mat'];
    load(ss,'data');
    len=size(data,1);
    i=1;

    while(data(i,1)<=0.8*iden_current(index))
        i=i+1;
    end
end

```

```

while(data(i,1)>0.2*iden_current(index))
    i=i+1;
end

while(mean(data(i-2:i+2,1)>mean(data(i:i+4,1))))
    i=i+1;
end
start=i;
last=start;
while(per>0)
    while(data(last,1)<=0.8*current(index))
        last=last+1;
    end
    while(data(last,1)>0.3*current(index))
        last=last+1;
    end
    while(mean(data(last-2:last+2,1)>mean(data(last:last+4,1))))
        last=last+1;
    end
    per=per-1;
end
tmpdata(pp:pp+last-start,:)=data(start:last,:);
pp=pp+last-start+1;
plot(tmpdata(:,1),'*');
clear data;
end %end for i
plot(tmpdata(:,1),'*');
iden_data=tmpdata(1:3:size(tmpdata,1),:);
save 'iden_data.mat' iden_data ;

*****
*****

```

sampledataforhighfrequency.m

```

global I_range level current iden_current findex frequency
pp=1; %for realdata
st=1; %for tmpdata
clear data;
clear realdata;
clear tmpdata;
for fi=1:length(frequency)
    per=1;
    pp=1;
    ss=['hf100smooth\' num2str(frequency(fi)) '.mat'];
    load(ss,'data');
    len=size(data,1);
    i=1;
    while(data(i,1)<=5)
        i=i+1;
    end
    while(data(i,1)>2)
        i=i+1;
    end
    while(data(i,1)>data(i+2,1))
        i=i+1;
    end
    start=i+1;
    last=start;
    while(per>0)
        while(data(last,1)<=10)
            last=last+1;
        end
        while(data(last,1)>9)
            last=last+1;
        end
        while(data(last,1)>data(last+2,1))
            last=last+1;
        end
    end
end

```

```

        per=per-1;
    end
    last=last;
    iden_hfdata=data(start:last,:);
    plot(iden_hfdata(:,1));
    ss=['iden_hf100\' num2str(frequency(fi)) '.mat'];
    save(ss,'iden_hfdata');
end %end for i

```

```

*****
*****

```

output.m

```

function y=Output(x,weights)
global level
num=level*(level+1)/2;
y=weights(1)+x'*weights(2:num+1);

```

```

*****
*****

```

nextstate.m

```

function sys=NextState(x,u)
global ref level;
num=level*(level+1)/2;
sys=x;
for i=1:num
    if(u>=ref(i,2))
        sys(i)=1;
    elseif(u<=ref(i,1))
        sys(i)=-1;
    else
        sys(i)=x(i);
    end
end
end

```

FindOneCycle.m

```
function y=FindOneCycle(data)
```

```
len=size(data,1);
```

```
i=1;
```

```
stop=0;
```

```
while(stop<3)
```

```
    i=i+1;
```

```
    if(data(i,1)>0.5)
```

```
        stop=stop+1;
```

```
    end
```

```
end
```

```
stop=0;
```

```
while(stop<3)
```

```
    i=i+1;
```

```
    if(data(i,1)<=0.5)
```

```
        stop=stop+1;
```

```
    end
```

```
end
```

```
start=i;
```

```
while(data(i,1)<0.5)
```

```
    i=i+1;
```

```
end
```

```
while(data(i,1)>=0.5)
```

```
    i=i+1;
```

```
end
```

```
stop=i;
```

```
y=data(start:stop,:);
```

objfun_weights.m

```
function y=objfun_weights(w,x)
```

```
global ref order level np nap output
```

```
num=level*(level+1)/2;
```

```
state=-1*ones(num,1); %each state is related to the corresponding ref
```

```
for i=1:length(x)
```

```
    state=NextState(state,x(i));
```

```
    y(i,1)=Output(state,w);
```

```
end
```

```
function sys=NextState(x,u)
```

```
global ref level;
```

```
num=level*(level+1)/2;
```

```
sys=x;
```

```
for i=1:num
```

```
    if(u>=ref(i,2))
```

```
        sys(i)=1;
```

```
    elseif(u<=ref(i,1))
```

```
        sys(i)=-1;
```

```
    else
```

```
        sys(i)=x(i);
```

```
    end
```

```
end
```

```
function y=Output(x,weights)
```

```
global level
```

```
num=level*(level+1)/2;
```

```
y=weights(1)+x'*weights(2:num+1);
```

objfun_dynamics.m

```
function y=objfun_dynamic(w,x)
global frequency tao delay seg simin simout T ti initial
pp=1;
tao=w(1:2);
%delay=[0 0];
for fi=1:length(frequency)
    if(frequency(fi)<=10)
        T=1/300;
        initial(1:2)=w(3:4)*T;
    else
        T=1/(frequency(fi)*30);
        initial(1:2)=w(3:4)*T;
    end
    y(pp:pp+seg(fi)-1,1)=obj_Preisach(w,x(pp:pp+seg(fi)-1));
    pp=pp+seg(fi);
end
```

obj_Preisach.m

```
function y=obj_Presaich(tc,x)
global T tao delay level initial

num=level*(level+1)/2;
state=-1*ones(3*num,1);
%delay=[0 0];%tc(3:4);
tao=tc(1:2);

for i=1:length(x)
    t=i*T;
```

```

state=Update(t,state,x(i));
y(i,1)=Output(t,state,x(i));
end

```

```

function sys=Update(t,x,u)
global ref interval delay tao level T initial
num=level*(level+1)/2;
sys=x;
for i=1:num
    base=3*(i-1);
    if(x(base+1)==1) % this relay is 'on'
        if(u<ref(i,1))
            sys(base+2)=t;
            sys(base+1)=-1;
            tmp=((t+initial(1))-x(base+2)-delay(1));
            if(tmp<0)
                tmp=0;
            end
            if(tao(1)<10^-6)
                ratio=0;
            else
                ratio=exp(-(tmp)/tao(1));
            end
            sys(base+3)=1-(1-x(base+3))*ratio;
        end
        continue;
    else %this relay is 'off'
        if(u>=ref(i,2))
            sys(base+2)=t;
            sys(base+1)=1;
            if(x(base+2)==-1) %first time
                sys(base+3)=-1;
            else
                % sys(base+3)=-1;
            end
        end
    end
end

```



```

        tmp=((t+initial(2))-x(base+2)-delay(2));
        if(tmp<0)
            tmp=0;
        end
        if(tao(2)<10^-6)
            ratio=0;
        else
            ratio=exp(-(tmp)/tao(2));
        end
        sys(base+3)=-1+(x(base+3)+1)*ratio;
    end
end
end
end
end

```

```

function sys=Output(t,x,u)
global weights level delay tao T initial
sys=weights(1);
num=level*(level+1)/2;
for i=1:num
    base=3*(i-1);
    if(x(base+2)==-1) %for the first time
        sys=sys+(-1)*weights(i+1);
        continue;
    end
    if(x(base+1)==1) %is on
        tmp=((t+initial(1))-x(base+2)-delay(1));
        if(tmp<0)
            tmp=0;
        end
        if(tao(1)<10^-6)
            ratio=0;
        else
            ratio=exp(-(tmp)/tao(1));

```

```

        end
        response=1-(1-x(base+3))*ratio;
    else %is off
        tmp=((t+initial(2))-x(base+2)-delay(2));
        if(tmp<0)
            tmp=0;
        end
        if(tao(2)<10^-6)
            ratio=0;
        else
            ratio=exp(-(tmp)/tao(2));
        end

        response=-1+(x(base+3)+1)*ratio;
    end
    sys=sys+response*weights(i+1);
end
sys=sys;

```

```

*****
*****

```

iden_xls.m

```

%identification weights
global iden_current
%generating the matrix for LSM
load 'iden_data.mat';
level=length(iden_current)-1;
num=level*(level+1)/2+1;
x0=zeros(num,1)+abs(rand(num,1));
lb=zeros(num,1);
options = optimset('Display','iter','MaxFunEvals',10^10,'MaxIter',70,'TolFun',10^-50,'TolX',0);
x = lsqcurvefit(@objfun_weights,x0,iden_data(:,1),iden_data(:,2),lb,[],options);
weights=x;

```

```

save 'xls_weights.mat' weights;
num=level*(level+1)/2+1;
load 'xls_weights.mat';
x=weights;
clear data;
clear y;
load '1hzsmooth\12.mat' data;
state=-1*ones(num-1,1);
for i=1:size(data,1)
    state=NextState(state,data(i,1));
    y(i,1)=Output(state,x);
end
xx=1:size(data,1);
plot(data(:,1),data(:,2),'r',data(:,1),y);
figure(2);
plot(xx,data(:,2),xx,y,'r-');
legend('experiment data','model output');

```

```

*****
*****

```

iden_dynamics.m

```

%identify Tc and Td
clc
clear all;
run preparedataforexcel;
global weights seg simin simout ti initial frequency
load 'xls_weights.mat' weights;
pp=1;
for fi=1:length(frequency)
    ss=['iden_hf100\' num2str(frequency(fi)) '.mat'];
    load(ss);
    len=size(iden_hfdata,1);
    alldata(pp:pp+len-1,:)=iden_hfdata;
    pp=pp+len;

```

```

    seg(fi)=len;
end
lb=zeros(2,1);
up=1*ones(2,1);
x0(1,1)=0.0014;
x0(2,1)=0.0008;
options = optimset('Display','iter','MaxFunEvals',10^10,'MaxIter',50,'TolFun',10^-50,'TolX',0);
x = lsqcurvefit(@objfun_dynamic,x0,alldata(:,1),alldata(:,2),lb,up,options);
tc=x;
save 'tc.mat' tcd;

```

Reference:

- [1] I.D. Mayergoyz, *Mathematical Models of Hysteresis*, Springer-Verlag, New York, 1991
- [2] X. Tan. *Control of Smart Structures*, PhD Thesis, University of Maryland, College Park, MD, 2003
- [3] R. Venkataraman. *Modeling and Adaptive Control of Magnetostrictive Actuators*, PhD Dissertation, University of Maryland, College Park, MD, 1999.
- [4] Lennart Ljung. *System Identification—theory for the user*, Prentice Hall PTR, New Jersey, 1999.
- [5] J.M. Cruz-Hernandez and V. Hayward. “Phase control approach to hysteresis reduction,” *IEEE Transactions on Control Systems Technology*, 9(1): 17-26, 2001
- [6] A. A. Adly, I. D. Mayergoyz, and A. Bergqvist. “Preisach modeling of magnetostrictive hysteresis,” *Journal of Applied Physics*, 69(8):5777–5779, 1991.
- [7] J. Takacs, “A phenomenological mathematical model of hysteresis”, *International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, Vol. 20, No. 4, 2001, pp. 1002-1014.
- [8] D. Hughes and J. T. Wen. “Preisach modeling and compensation for smart material hysteresis,” In G. L. Anderson and D. C. Lagoudas, editors, *Active Materials and Smart Structures*, volume 2427 of *SPIE*, pages 50–64, 1994.
- [9] X. Tan, R. Venkataraman, and P. S. Krishnaprasad. “Control of hysteresis: theory and experimental results,” In V. S. Rao, editor, *Modeling, Signal Processing, and Control in Smart Structures*, volume 4326 of *SPIE*, pages 101-112, 2001.

- [10] Galinaitis, W. S., & Rogers, R. C. (1998). "Control of a hysteretic actuator using inverse hysteresis compensation," *Mathematics and control in smart structures*, Vol. 3323 (pp. 267–277). Bellingham, Washington, USA: SPIE.
- [11] Banks, H. T., Kurdila, A. J., & Webb, G. (1997). "Identification of hysteretic control influence operators representing smart actuators, Part I: Formulation," *Mathematical Problems in Engineering*, 3(4), 287–328.
- [12] Giorgio Bertotti. (1992). Dynamic Generalization of the Scalar Preisach Model of Hyteresis, *IEEE Transactions on Magnetics*, Vol. 28, No. 5, September 1992.
- [13] G. Bertotti, f. Fiorillo, and M. Pasquale, Measurement and prediction of dynamic loop shapes and power losses in soft magnetic materials, *IEEE Transactions on Magnetics*, vol. 29, pp. 13-16, Nov.1993
- [14] J. Fuzi and A. Ivanyi, Features of two rate-dependent hysteresis models, *Physica B*, vol. 306, pp. 137-142, 2001
- [15] D.C.Jiles, "Frequency dependence of hysteresis curves in 'nonconducting' magnetic materials," *IEEE Transactions on Magnetics*, vol. 29, pp. 3490-3492. November 1993.
- [16] D.C. Jiles and D. L. Atherton, "Theory of ferromagnetic hysteresis," *Journal of Magnetism and Magnetic Materials*, vol. 61, pp. 48-60, 1986.
- [17] M.L. Hodgdon, "Applications of a theory of magnetic hysteresis," *IEEE Transactions on Magnetics*, vol. 24, pp. 218-221, January 1988.
- [18] B.D. Coleman and M.L. Hodgdon, "A constitutive relation for rate-idependent hysteresis on ferromagnetically soft materials," *International Journal of Engineering Science*, vol. 24, no. 6, pp. 897-917, 1986.

- [19] I.D. Mayergoyz, "Dynamic Preisach models of hysteresis," *IEEE Transactions on Magnetics*, vol. 24, pp. 2925-2927, September 1988.
- [20] Paolo Del Vecchio and Alessandro Salvini, "Neural Network and fourier descriptor macromodeling dynamic hysteresis," *IEEE Transactions on Magnetics*, vol. 36, No. 4, July 2000.
- [21] Alessandro Salvini and Christian coltelli, "Prediction of dynamic hysteresis under highly distorted exciting fields by neural networks and actual frequency transplantation," *IEEE Transactions on magnetics*, vol. 37, No. 5, September 2001.
- [22] Alessandro Salvini and Francesco Riganti Fulginei. (2002). "Genetic Algorithms and Neural Networks Generalizing the Jiles-Atherton Model of Static Hysteresis for Dynamic Loops," *IEEE Transactions On Magnetics*, Vol.38, No.2, March 2002
- [23] Alessandro Salvini, Francesco Riganti Fulginei and Giuseppe Pucacco. (2003). "Generalization of the static Preisach Model for Dynamic Hysteresis by a Genetic Approach," *IEEE transactions on Magnetics*, VOL.39, NO.3, May 2003.
- [24] Hong hu, Huanshui Zhang and Ridha Ben Mrad, "Preisach based dynamic hysteresis model," *Proceedings of the 2004 International Conference on Intelligent Mechatronics and Automation*, Chengdu, China, August 2004.
- [25] R. Ben Mrad, and H. Hu. "A model for voltage-to-displacement dynamics in piezoceramic actuators subject to dynamic-voltage excitations", *IEEE Transactions on Mechatronics*, vol. 7, No. 4, December 2002.
- [26] Silvano Cincotti. (2001). Dynamic properties of a Piece-Wise Linear Circuit Model of Hysteresis, *IEEE Transactions on Magnetics*, Vol.37, NO.5, September 2001.

- [27] Silvano Cincotti and Ivano Daneri, A non-linear circuit model of hysteresis, *IEEE Transactions on magnetics*, vol. 35, No. 3, May 1999.
- [28] Nickie Menemenlis. (1998). Noniterative Dynamic Hysteresis Modeling for Real-Time Implementation, *IEEE Transactions on Power Systems*, Vol. 13, No.4, November 1998.
- [29] T. Chevalier, A. Kedous-Lebouc, B. Cornut, and C. Cester. (2000). "A new dynamic hysteresis model for electrical steel sheet," *Physica B*, 275 (2000) 197-201.
- [30] Dimitre Makaveev, Luc Dpre, Marc De Wulf, and Jan Melkebeek. (2003). "Dynamic hysteresis modeling using feed-forward neural networks," *Journal of magnetism and magnetic materials*, 254-255 (2003) 256-258.
- [31] S.E. zirka, Y.I. Moroz, P. Marketos, and A.J.Moses. (2004). "Dynamic hysteresis modeling," *Physica B*, 343 (2004) 90-95.
- [32] Kenneth H. Carpenter, "Simple models for dynamic hysteresis which add frequency-dependent losses to static models," *IEEE Transactions on Magnetics*, vol. 34, No. 3, May 1998.
- [33] Janos Fuzi, "Computationally efficient rate dependent hysteresis model," *International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, vol. 18, No. 3, 1999, pp. 445-457.
- [34] J. Takacs, *Mathematics of hysteretic phenomena : the T(x) model for the description of hysteresis*, Weinheim, Cambridge, 2003.
- [35] R.C. Smith and C.L. Hom, "A domain wall theory for ferroelectric hysteresis", *Technical Report*, Center for Research in Scientific Computation, North Carolina State University, CRSC-TR99-01, January 1999.

- [36] M.A. krasnosel'skii and A.V. Pokrovskii, *Systems with Hysteresis*, Nauka, Russia, 1983; translation, Springer-Verlag, 1989.
- [37] A. Cavallo, C. Natale, S. Pirozzi, and C. Visone. "Effects of hysteresis compensation in feedback control systems," *IEEE Transactions on Magnetics*, 39(3): 1389–1392, 2003
- [38] R. Smith and J. Nealis. "An Adaptive Control Method for Magnetostrictive Transducers with Hysteresis," *Technical Report*, Center for Research in Scientific Computation, North Carolina State University, CRSC-TR01-02, August 2001.
- [39] G.Tao and P.V.Kokotovic. "Adaptive control of Plants with unknown hysteresis," *IEEE Transactions on Automatic Control*, 40(2): 200-212, 1995.
- [40] G.Tao and P.V.Kokotovic. *Adaptive Control of Systems with Actuator and Sensor Nonlinearities*, John Wiley & Sons, 1996.
- [41] X. Tan and J.S. Baras. "Modeling and control of hysteresis in magnetostrictive actuators," *Automatica*, 40 (2004) 1469-1480
- [42] J.Nealis and R. Smith. "Robust Control of a Magnetosrictive Actuator," *Technical Report*, Center for Research in Scientific Computation, North Carolina State University, CRSC-TR03-05, Feb 2003.
- [43] X. Tan, J. S. Baras, and P. S. Krishnaprasad. "Control of Hysteresis in Smart Actuators with Application to Micropositioning," ISR Technical Report, TR 2003-39, University of Maryland, College Park, 2003.
- [44] Y. M. Zhang and R. Kovacevic, "Robust control of interval plants: A time domain method," *IEE Proceedings-Control Theory Applications*, 144 (4): 347-353 JUL 1997.
- [45] R. Venkataraman and P. S. Krishnaprasad. "Approximate inversion of hysteresis: theory and numerical results", In *Proceedings of the 39th IEEE Conference on Decision and Control*, pages 4448–4454, Sydney, Australia, December 2000.

- [46] Joseph Michael Istre, *Robust Generic Model Control for Parameter Interval Systems*, PhD Thesis, University of Kentucky, Lexington, KY 2004.
- [47] R. Rindeisen and R. Allgower, “An Introduction to Nonlinear Model Predictive Control.” Institute for Systems Theory in Engineering, Stuttgart, Germany.
- [48] Rawlings, “Tutorial Overview of Model Predictive Control,” *IEEE Control Systems Magazine*, pp. 38-52, June 2000.
- [49] Energen Inc Company, <http://www.energeninc.com/magneticmartmat.html>

VITA

Date of Birth: February 19th, 1980

Place of Birth: Benxi, Liaoning, China

Education:

1999~2003: University of Science and Technology of China, Hefei, Anhui, China

2003~2005: University of Kentucky, Lexington, Kentucky, U.S.A

Honors and Awards

2003: Outstanding Bachelor Thesis of University of Science and Technology of China

2003 Samsung Scholarship

2002 First-Place Award in Hua Wei Science and Technology Competition.

2002: Excellent Student Scholarship

2001: Excellent Student Leader of USTC

2000: China Goodwill&Peace Development Foundation Panasonic Electronics Talent Cultivation Foundation Scholarship

1999: HuangWanYu Scholarship in the National College Entrance Examination.

1998: First-Grade Prize in National Senior Middle School Physics Contest

1996: First place in the Senior Middle School Entrance Examination of Benxi, Liaoning

1995: Second-Grade Prize in National Junior Middle School Chemical Contest

1995: Excellent Student Leader of Benxi

1995: Excellent Student Leader of Liaoning province

1994: Top Ten Excellent Middle School Student Prize of Benxi

1993: First-Grade Prize in City-Wide Middle School Biology Contest

1993: Full mark Prize of National Elementary School Math Olympic Contest