Methods for Modeling and Control of Systems with Hysteresis of Shape Memory Alloy Actuators

Yu Feng Wang

A Thesis

in

The Department

of

Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Philosophy (Mechanical Engineering) at
Concordia University
Montreal, Quebec, Canada

August 2006

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-23843-1
Our file Notre référence
ISBN: 978-0-494-23843-1

ABSTRACT

Yu Feng Wang, Ph.D.
Concordia University, 2006

Hysteresis widely exists in smart materials such as shape memory alloys (SMAs), piezoelectrics, magnetorheological (MR) fluids, electrorheological (ER) fluids and so on. It severely affects the applicability of such materials in actuators and sensors. In this thesis, problems of modeling and control of systems with hysteretic SMAs actuators are studied. The approaches are also applicable to control of a wide class of smart actuators.

Hysteresis exhibited by SMAs actuators is rate-independent when the input frequency is low, and can be modeled by a classical Preisach model or a KP model. The classical Preisach hysteresis model is a foundation of other hysteresis models. In this thesis, traditional methods are explained in advance to identify and implement the classical Preisach model. Due to the extremely large amount of computation involved in the methods, a new form of the Preisach model, linearly parameterized Preisach model, is introduced, and then an effective method to implement the model is presented. The KP model is a more effective operator to describe the Preisach class of hysteresis than the Preisach model. The relationship between the two models is revealed to verify the effectiveness of the KP model. Also, a linearly parameterized KP model is proposed. For both of the Preisach hysteresis model and the KP hysteresis model, algorithms of inverse hysteresis operators are developed, and simulations for modeling and inverse compensation are conducted.

Since the Preisach model and the KP model can only describe hysteresis which has saturation states and reverse curves with zero initial slopes, a novel hysteresis model is defined to overcome these shortcomings. The newly defined hysteresis model is a low dimensional hysteresis model and can describe hysteresis which has revertible linear parts and reverse curves with non-zero initial slopes.

The problems for controlling systems with input hysteresis have been pursued along three different paths: inverse compensation, gradient adaptive control and robust adaptive control for linear and nonlinear systems.

Control schemes of open-loop inverse compensation and gradient adaptive inverse compensation for the Preisach hysteresis model are explored to eliminate the effects of the hysteresis when the output of the hysteretic actuator is measurable.

Usually hysteresis of smart actuator in systems is not exactly known, but it can be approximately modeled via the linearly parameterized KP model. For a known linear system preceded by an unknown actuator hysteresis, a model reference control scheme combining with an adaptive inverse compensation is designed for tracking control of the systems. While an unknown linear system preceded by an unknown actuator hysteresis, a model reference adaptive control scheme together with an adaptive inverse compensation is developed for tracking control of the system. Simulations for both cases have been performed to illustrate the control methods.

Finally, when hysteresis of smart actuator in systems has a non-measurable output and is modeled via the KP model or the newly defined model, a novel robust adaptive control configuration is presented for tracking control of systems. The analysis for the stability and the convergence of the control systems is conducted. Simulations are performed to verify the effectiveness of the novel control method.

ACKNOWLEDGMENTS

It has been my privilege to have both Professor Chun-Yi Su and Professor Henry Hong as my advisors. I am grateful to them for their thoughtful guidance and enthusiastic encouragement. They inspired me with their vision and expertise during my PhD study. Apart from providing technical directions, they have also offered me with invaluable advices on how to improve myself as a researcher.

I am grateful for the financial supports of my studies and research from Fonds de recherché sur la nature et les technologies (FCAR) and Concordia University.

My special thanks are due to my friends at Montreal who have offered help in various ways: Tong Sheng, Zhi-Yong Zhong, Ming-Yi Huang, and Dao-Hong Jiang.

To my wife, Danyi: You have worked hard to support our family, to educate our lovely son. You constantly encourage me to continue my study and to overcome all difficulties I encountered. I love you forever.

To my son, Bufan: Thank you for being proud of me. Your good progress on study also encourages me.

To my parents, and my parents in law: Thank you for your love and support.

TABLE OF CONTENTS

CHAPTER 8 A NOVEL HYSTERESIS MODEL

CHAPTER 9 CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

APPENDICES

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

$v(t)$ Input of hysteretic actuators

$u(t)$ Output of hysteretic actuators

$\dot{u}(t)$ The varying tendency of the hysteresis process

$u_d(t)$ Desired output of a hysteretic actuator

$H(\cdot)$ A nonlinear operator to map the input $v(t)$ to the output $u(t)$ of hysteretic actuators

$\hat{H}(\cdot)$ Approximated hysteresis operator

$H^{-1}(\cdot)$ Hysteresis inverse operator to transform the desired output $u_d(t)$ to required input $v(t)$ so that $v(t)$ can generate output $u(t) = u_d(t)$

$\hat{H}^{-1}(\cdot)$ Approximated hysteresis inverse operator

t Moment of time

t_0 A certain moment of time

u^- Output of the negative saturation state

u^+ Output of the positive saturation state

v^- Input value determining when the negative saturation state starts $v(t) \leq v^-$

v^+ Input value determining when the positive saturation state starts $v(t) \geq v^+$

v_{ps} Input value corresponding starting of positive saturation state before coordinate transformation

v_{ns} Input value corresponding starting of negative saturation state before coordinate transformation

$\hat{\gamma}_{\alpha\beta}$ Output of an elementary operator, relay, corresponding to parameters α and β

α The "on" switching threshold of input applied on the elementary operator (relay) of Preisach hysteresis model

β The “off” switching threshold of input applied on the elementary operator (relay) of Preisach hysteresis model

$\mu(\alpha,\beta)$ Weighting function (density function) of parameters α and β

T The Preisach plane consisting of all possible pairs of parameters α and β

$L(t)$ Memory interface curve separates the Preisach plane into two domains $S^{+}(t)$ and $S^{-}(t)$

$S^{+}(t)$ A domain in the Preisach plane where $\hat{\gamma}_{\alpha\beta}=+1$

$S^{-}(t)$ A domain in the Preisach plane where $\hat{\gamma}_{\alpha\beta}=-1$

$\Delta S(t)$ Change of domain $S^{+}(t)$ or $S^{-}(t)$ caused by the varying memory interface curve $L(t)$

$t_1,t_2,...$ Time sequence

$v_1,v_2,...$ Input sequence corresponding to the time sequence of $t_1,t_2,...$

m_k Global minimum of inputs

M_k Global maximum of inputs

v_- A local minimum input value

v_+ A local maximum input value

Δv Change between the local minimum input v_- and local maximum input v_+

$\Delta T_1,\Delta T_2$ Changes of domain $S^{+}(t)$ or $S^{-}(t)$ caused by the varying memory interface curve $L_1(t)$ and $L_2(t)$ between two local extrema v_- and v_+

$k_1,k_2,...$ Initial slopes of reverse curves of minor loops of hysteresis

Ω Change of domain $S^{+}(t)$ or $S^{-}(t)$ caused by the descending inputs along the first order descending curve (FOD)

$u_{\alpha'}$, $u_{\alpha'\alpha'}$ Output of hysteresis when input increases from an input $v(t) \le v^-$ to an input value $v(t) = \alpha'$

$u_{\alpha'\beta'}$ Output of hysteresis when input decreases along a FOD curve from an input $v(t) = \alpha'$ to an input value $v(t) = \beta'$

l Number of lines, horizontal and vertical respectively, to divide the Preisach plant T into small cells

N Number of points divided by l horizontal lines and l vertical lines

i The sequence number of horizontal lines to divide the Preisach plane T

j The sequence number of vertical lines to divide the Preisach plane T

α_i Vertical coordinates of points divided by l horizontal lines and l vertical lines

β_j Horizontal coordinates of points divided by l horizontal lines and l vertical lines

$\hat{u}_{\alpha_i\beta_j}$ Calculated output of hysteresis through an assumed formula when input decreases along a FOD curve from an input $v(t) = \alpha_i$ to an input $v(t) = \beta_j$

$u_{\alpha_i\beta_j}$ Measured output of hysteresis when input decreases along a FOD curve from an input $v(t) = \alpha_i$ to an input value $v(t) = \beta_j$

$x_1, x_2, \ldots$ Parameters of an assumed output formula $\hat{u}_{\alpha\beta} = f(x_1, x_2, x_3, \ldots, x_n, \alpha, \beta)$

$e(x_1, \ldots, x_n)$ Least square error of output $u_{\alpha_i\beta_j}$ expressed in terms of parameters $x_1, \ldots, x_n$

X Vector form of parameters $x_1, x_2, \ldots$ defined as $X = [x_1, x_2, x_3, \ldots, x_n]$

$F(X)$ Vector form of calculated outputs $\hat{u}_{\alpha_i\beta_j}$

U Measured output expressed in vector form as $U = [u_1, u_2, u_3, \ldots, u_N]$ whose elements are rearranged from outputs $u_{\alpha_i\beta_j}$

$F_{\alpha\beta}$ The difference between the output $u_{\alpha\alpha}$ of the starting point (α, α) on the FOD curve and the output $u_{\alpha\beta}$ of a point (α, β) on the FOD curve, defined by $F_{\alpha\beta} \overset{\Delta}{=} u_{\alpha\alpha} - u_{\alpha\beta}$

$F_{\alpha}^{-1}(\cdot,\beta)$ An inverse function to calculate increasing input value α respected to a given input value β which is the end input value of previous descending process

$v^{*}(\tau)$ Exact input value corresponding to the desired output $u(\tau)$

$v_{present}$ Present input value

$u_{present}$ Present output value

dv Step interval for approaching exact input value $v^{*}(\tau)$ in the approximate inverse hysteresis algorithm

v_{LE} Vector of local input extrema

$\rho_{\alpha_i\beta_j}$ Lumped density of the $ij th$ cell to its lower-left node (β_j,α_i)

c Total density value of the Preisach hysteresis model

$\mu_{n\alpha_i\beta_j}$ Normalized densities of all lower-left nodes on the Preisach plane T

$F(v)$ Vector form of outputs of all nodes (relays) in the Preisach plane T

Θ Vector form of normalized densities of all nodes in the Preisach plane T

$\hat{\Theta}$ Guessed density vector of vector Θ

$\tilde{\Theta}$ Vector of density errors

σ Parameter determines the spread of the density distribution over the Preisach plane T

α_0,β_0 The center of the density distribution over the Preisach plane T

$G_p(s)$ Transfer function of a plant

$G_m(s)$ Transfer function of a reference model

$D_m(s)$ Denominator of the transfer function of the reference model $G_m(s)$

$D_p(s)$ Denominator of the transfer function of the plant $G_p(s)$

$g_m(s)$ Gain of the transfer function of the reference model $G_m(s)$

$g_p(s)$	Gain of the transfer function of the plant $G_p(s)$
n_s	Normalizing signal designed
e	Estimation error between the outputs of plants
ε	Normalized estimation error corresponding to e
$J(\hat{\Theta})$	A quadratic cost function of parameters $\hat{\Theta}$
Γ	Vector consisting of positive adaptive gains
P	The Preisach plane noted for KP hysteresis model
k_p	Output of a KP elementary operator (kernel)
p_1, p_2	Two parameters determine switching thresholds of a kernel
a	A parameter determines the rise-constant of a kernel
$\xi_p(t)$	A variable to memory the previous extreme output of a kernel
$\mu(p)$	Density of the kernel k_p
$r(\cdot)$	Ridge function of the kernel k_p
m_p	An elementary compensator to the kernel k_p
(ψ, ς_p)	Coordinates of a turning point of an elementary compensator m_p
$r_c(\cdot)$	Ridge function of the elementary compensator m_p
$\Delta v'$	The interval of input between two adjacent dividing lines
$\rho_{p_{ij}}$	Lumped density of a cell to its lower-left node (v_i, v_j)
$\bar{\mu}_{ij}$	Normalized densities of all lower-left nodes on the Preisach plane P
K^T	Output vector of kernels k_p

M^T	Output vector of elementary compensators m_p of KP model
Γ^T	Parameter vector (normalized densities) of kernels k_p
$\hat{\Gamma}^T$	Estimated parameter vector (estimated normalized densities) of kernels k_p
$\tilde{\Gamma}^T$	Parameter error vector of kernels k_p
$\bar{p}_1,\bar{p}_2$	The center of the density distribution over the Preisach plane P of KP model
$D_m(s)$	Denominator of transfer function of reference model
$D_p(s)$	Denominator of transfer function of plant
Z	Vector consisting of positive adaptive gains
$J(\hat{\Gamma})$	A quadratic cost function of parameters $\hat{\Gamma}$
Q	Constraint set for the estimated parameter vector $\hat{\Gamma}$
Q_o	The interior of Q
$\delta(Q)$	The boundary of Q
y	Output of the plant $G_p(s)$
y_m	Output of the reference model $G_m(s)$
$\omega_1(t)$	Filtered signal form the input u of the plant $G_p(s)$
$\omega_2(t)$,	Filtered signal form the output y of the plant $G_p(s)$
$\theta_0,\theta_1,$ θ_2,θ_3	Control gains in MRC scheme or adaptive control gains of MRAC scheme
$r(t)$	Reference signal
$\Delta r(t)$	An extra reference signal
$e_h(t)$	An input error to the plant caused by inaccurate compensation

$\varepsilon_h(t)$	An augmented error
$\xi_h(t)$	An auxiliary error
$m_h(t)$	Normalization signal
y_i	Nonlinear components of a nonlinear plant
a_i	Parameters of nonlinear components of the nonlinear plant
$x^{(n)}(t)$	States of the nonlinear plant
${x_d}^{(n)}(t)$	Desired states of the nonlinear plant
Γ_n	A vector consisting of normalized densities of hysteresis
$\tilde{X}(t)$	Tracking error vector
$s(t)$	Filtered tracking error
s_ε	A tuning error of the filtered tracking error $s(t)$
k_d	Positive gains for the filtered tracking error $s(t)$ controller
$\hat{\phi}(t),\hat{\Theta}(t),\hat{\Gamma}_n$	Adaptive parameters in adaptive laws
η, γ, q	Positive gains for parameters $\hat{\phi}(t),\hat{\Theta}(t),\hat{\Gamma}_n$

CHAPTER 1

BACKGROUND

1.1 Hysteresis in Shape Memory Alloys

The phenomenon of hysteresis have been found and have been noticed in scientific and engineering applications for many decades, and the study to eliminate and/or to control hysteresis effects have been attracting more research studies than before. Hysteresis occurs in many different areas of science such as plasticity, ferromagnetism, ferroelectricity, shape memory alloys, superconductivity, porous media filtration, semiconductors and so on [13, 14]. Typically smart materials, such as magnetostrictives, piezoelectrics, electroactive polymers (EAPs), shape memory alloys (SMAs), electrorheological (ER) fluids and magnetorheological (MR) fluids, exhibit significant hysteresis [10]. Hysteresis is a form of nonlinearity that contains memory, thus there may be multiple possible outputs for a given input. Unmodeled hysteresis in systems can lead to inaccuracy in open-loop control, and reduce the effectiveness of feedback control for the systems.

Actuators made of the smart materials SMAs have been receiving tremendous interest and have broad applications in the past decade because they have some excellent properties such as very good mechanical power-to-mass ratio, noiseless and anticorrosion, and can be conventionally built into structures with the ability to respond to environmental changes to achieve desired goals. Hysteresis existing in SMAs, however, makes the effective use of the SMA actuators quite challenging.

SMAs are materials capable of returning to some previously defined shape or size when they are subject to the appropriate thermal procedure and external force. This phenomenon is referred to as the Shape Memory Effect (SME). It occurs due to a temperature and stress dependent shift in the material's crystalline structure between two different phases called martensite and austenite [1, 19, 36]. Martensite, the low temperature phase, is relatively soft; whereas austenite, the high temperature phase, is relatively hard. The change in SMA's crystalline structure is not a thermodynamically reversible process. In other words, there is energy dissipation due to internal friction and creation of structural defects.

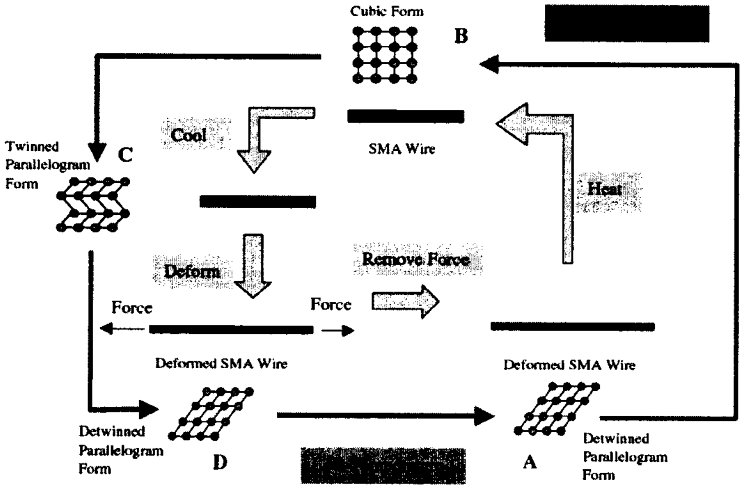


Fig.1.1 Phase transformation of the SMA wire [1]

The hysteresis in SMAs is caused by their phase transformation, in which SMAs convert thermal energy to mechanical work by the change of the lattice structure. The phase transformations of an SMA wire actuator are illustrated in Fig.1.1. Assume that the SMA wire is initially at a low temperature and is in its martensite state (point A in Fig.1.1). Upon heating, the SMA wire will experience a phase transformation to the cubic stronger austenite and the wire will contract in its length (point B in Fig.1.1). Upon cooling, the SMA wire will transfer from austenite to the weaker martensite phase (point C in Fig.1.1). At this stage, the crystal structure of the SMA is in a twinned parallelogram form. In general, its strength in terms of Young's modulus in martensite is three to six times less than its strength in austenite. When an external tension force is applied to the wire, the wire can be easily stretched (point D in Fig.1.1). During its phase transformation, an SMA generates an extremely large force when encountering resistances or experiences a significant change in dimensions when unrestricted. The transformation exhibits a hysteretic effect (see Fig.1.2), in which the transformations on heating and on cooling do not overlap.

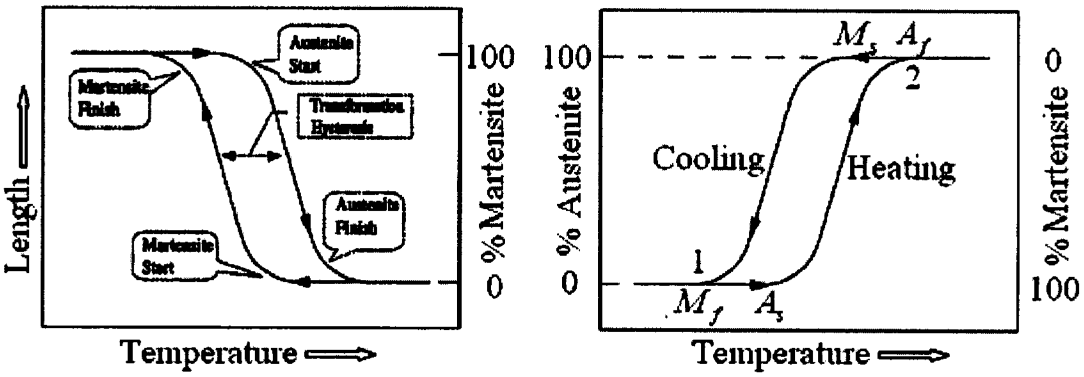


Fig.1.2 The hysteresis loop associated with SMAs [1].

The hysteretic effect, shown in Fig.1.2, adversely affects precise control of SMA actuators and may even cause the system to experience instability. Thus, the hysteresis behavior makes it challenging to develop model and control schemes for SMA actuators, and the compensation of the hysteresis is a major concern during the design of control systems for SMA actuators.

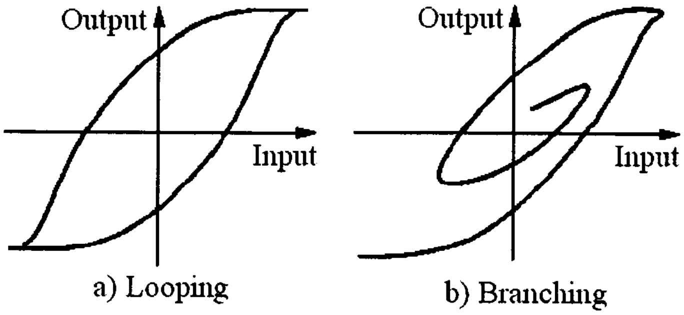
1.2 Fundamentals of Hysteresis

Hysteresis nonlinearities existing in different areas of science are defined by different meanings. To avoid confusion and ambiguity, a mathematical description of scalar hysteresis is abstracted as below.

v(t)
Hysteresis Transducer
u(t)

Fig. 1.3 Hysteresis Transducer

Consider a transducer (see Fig.1.3) that can be characterized by an input $v(t)$ and an output $u(t)$. This transducer is called a hysteresis transducer H if its input-output relationship is a multibranch nonlinearity (see Fig.1.4) for which branch-to-branch transitions occur after input extremes. Particularly, **loops** are created when an input is varied back and forth between two consecutive extremes (see Fig.1.4a). This is not the essence of hysteresis; however, it is a particular case of **"branching,"** which occurs at the reversals of an input (see Fig.1.4b). Generally, **hysteresis** is a multi-branching nonlinearity that occurs when the output of a system lags behind the input.



a) Looping b) Branching

Fig.1.4 Looping and branching in hysteresis

Definition of hysteresis [14]: At any time t, the output $u(t)$ of a system depends not only on its input $v(t)$, but also on its previous trajectory (memory effect). The input–output relationship is invariant with respect to changes in the time scale (rate independence). When a system has memory effect and is rate independent, it is said to have hysteresis.

Rate independent implies the input-output relationship is invariant with respect to changes in time scale (see Fig.1.5).

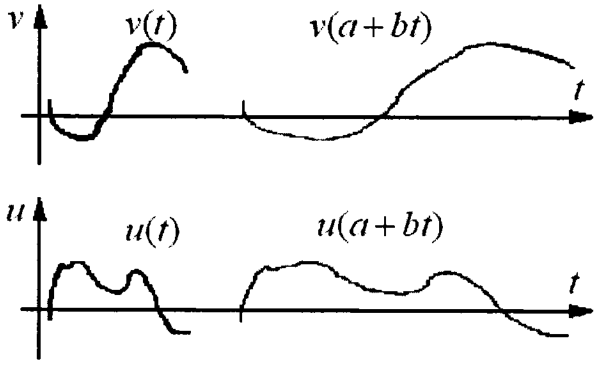


Fig.1.5 Rate independent property of hysteresis

Memory effect implies that the output at a moment of time cannot be determined simply by knowing the input at the moment, the history of the input must also be known.

Memory effect in hysteresis is a kind of "***non-local memory,***" which means that the future output values of the hysteresis $u(t), \forall t \geq t_0$, depend not only on $u(t_0)$ but also on past extremes of the input $v(t)$ [18]. The special case of "***local memory***" implies that the value of the output $u(t_0)$ at some instant and the value of the input $v(t)$ at all subsequent instants in time $t \geq t_0$ uniquely determine the future value of $u(t)$.

Classifications of hysteresis: Hysteresis nonlinearities observed in real systems can be classified into the four following cases:

- Nonsaturated and rate independent hysteresis (e.g., ceramic actuators [22]);
- Saturated and rate independent hysteresis; (e.g., SMA actuators [24]).
- Nonsaturated and rate dependent hysteresis [23];
- Saturated and rate dependent hysteresis.

The class of saturated rate independent hysteresis is the main concern of study in this dissertation. Before detail discussions, some notations concerning saturated rate independent hysteresis are defined. In a saturated hysteresis as shown in Fig.1.6, when the input decreases to v^- , it is said that the system has reached its **negative saturation state**. This occurs when further decreasing in the input yields little or no change in the output. Similarly, it is said that the system has reached its **positive saturation state** as the input increases to v^+. When the input increases from v^- to v^+, the input-output pair (v,u) follows path 1, which is called the **limiting ascending branch** of the hysteresis loop because, in general, (v,u) will never be below this branch of the loop. Similarly, path 2 is called the **limiting descending branch**. The loop constructed by path 1 and 2 is the **major loop**. Path 3 and path 4, called **reversal curves**, occur when the input reverses its direction at a point other than v^- or v^+. The loop closed by paths 3 and 4 is called a **minor loop**.

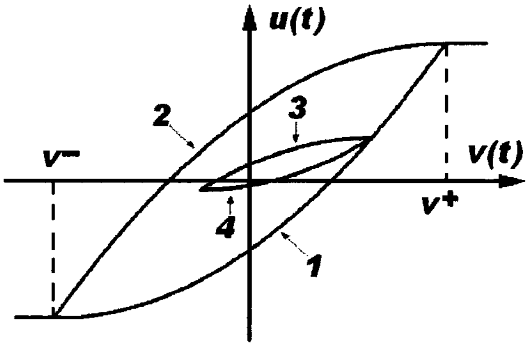


Fig.1.6 Hysteresis graph

1.3 Hysteresis Models

Hysteresis have been found in many physical systems and a number of models have been introduced to describe the phenomenon, including the Duhem model, the Classical Preisach model, the Prandtl-Ishlinskii model, Krasnosel'skii-Pokrovskii (PK) model , and the Stoner-Wohlfarth model, etc. A description of these models can be found in various monographs of Brokate and Sprekels [15], Della Torre [16], Mayergoyz [13] ,Stoner and

Wohlfarth [38], and Visintin [14, 37].

The Duhem model describes hysteresis using two systems of differentiable curves in the input-output plane, one for increasing input and the other for decreasing input respectively [14]. This formulation can be easily modified to confine input-output relationship to a loop-like set. However, it can not precisely describe some complex hysteresis, for example, some hysteresis with saturation states.

The Preisach model was originally introduced by Ferenc Preisach in 1935 [39] as a phenomenal model for hysteresis in ferromagnetic materials. Until the 1970's, a rigorous mathematical treatment had been made by a group of Russian mathematicians from the phenomenological character of the Preisach's model [12]. At present, this model is widely utilized in piezoelectric, magnetostrictive and shape memory alloy actuators. This model describes hysteresis via a linear combination of elementary operators. It is a discontinuous model because its elementary operators are some discontinuous "relays". This model can precisely model a class of hysteresis and reveals many properties of hysteresis. Also, it is the basis of some hysteresis models.

The Prandtl-Ishlinskii model and the Krasnosel'skii-Pokrovskii (PK) model [12] are continuous models because they employ continuous elementary operators such as so-called "play" and "stop". They are more effective models than the Preisach model.

There are many literatures on hysteresis modeling, especially for ferromagnetic materials, but a relatively small portion of them is related to other smart materials, for example, shape memory alloys. Furthermore, the models are usually developed for physical based simulation rather than for stability analysis and control design.

A fundamental idea in dealing with hysteresis is to formulate the mathematical model of hysteresis and use inverse compensation to cancel the hysteretic effect. This idea can be found in [6~11]. Also, there have been a few monographs devoted to the modeling of hysteresis and study of dynamical systems with hysteresis [12~16].

1.4 Control Strategies for Systems with Hysteresis

Tracking control of smart material based actuators is essential in many applications such as vibration controls [2, 3] and robotic applications [4]. Considering the fact that a position sensor is a major cost of a smart actuator system, open-loop control without a position sensor is a preferred approach to reduce the total cost. However, due to the hysteresis, an inherent nonlinear phenomenon associated with smart materials, tracking control of smart actuators is a challenging task [5]. This has motivated researchers to the study of tracking control of smart actuators with hysteresis compensation.

Cru′ z-Hernandez and Hayward [17] have discussed in their paper the use of phasers for compensation of the hysteresis by shifting the phase of the periodic signal in a piezoelectric actuator. Webb *et al.*[18] have presented an adaptive hysteresis model for on-line identification and closed-loop compensation. A neural network controller for tracking control of the SMA wire actuator was introduced by Song *et al* [1]. Hughes and Wen [20] implemented the Preisach Model in the control of a SMA wire actuator to provide bending force to a flexible aluminum beam. Ge and Jouaneh [21] have used a combination of feed-forward controller with a feedback loop (PID) to reduce hysteresis in actuators represented by the Preisach model. Tao *et al.* [7, 29] have used adaptive methods to control plants with hysteretic behavior. They have developed control algorithms to reduce the effects of hysteresis nonlinearities. Other monographs [26, 30~35] also are devoted to control of hysteresis in smart materials.

1.5 Studies and Contribution in the Dissertation

In this dissertation methods for modeling hysteresis of shape memory alloy actuators and control methodologies for systems preceded by the actuators will be studied.

To study the problem of modeling hysteresis in smart material based actuators, three models will be adopted. The first model is the Classical Preisach model, which reveals the properties of the hysteresis of smart actuators, and is the basis of other models to describe the Preisach class of hysteresis. The second is the Krasnosel'skii-Pokrovskii (KP) model, which governs the Preisach class of hysteresis with few elementary operators, and is more suitable for real-time adaptive control. The third is a newly defined hysteresis model, which improves the shortcomings of the Preisach model and the KP model, and is more general and applicable for modeling hysteresis in smart materials than the Preisach model and the KP model.

The problem of hysteresis control is pursued from three perspectives. The first is based on the Preisach model and the theme is to develop accurate and fast inverse control algorithms. The second perspective is optimal control based on the approximate inversion of the KP model. The third is adaptive control through adopting a compensator from the KP model and the newly defined hysteresis model.

The following outlines the contributions of this dissertation. First, this dissertation develops a direct method to implement the Classical Preisach model. This method avoids the time-consuming computation to find the memory interface which is mostly used by existing literatures to model the Preisach class of hysteresis. Also, this method can be applied to implement other models, particularly when a model employs few elements to describe the hysteresis. Second, this dissertation reveals the relationship between the

Preisach model and the KP operator. This relationship proves that the KP model possesses the same properties as the Preisach model, and guarantees suitable application of the KP model in modeling the Preisach class of hysteresis. Third, by examining the shortcomings of the Preisach model and the KP model, a new model is developed, which can describe the reversible parts in real hysteresis after exceeding saturation states. Also, this new model can express reversal branches with arbitrary initial slopes in hysteresis loops; conversely, reversal branches modeled by the Preisach model and the KP model feature zero-initial-slope. Finally, a novel robust adaptive control methodology, which is based on a newly defined elementary compensator of the KP model rather than the inverse KP model, has been established to reduce the hysteresis effect injecting into systems and achieves the trajectory tracking goal of the systems.

The rest of the dissertation is organized as follows. Chapter 2 reviews the definition and some key properties of the Preisach model. Aided by the key properties, simulations for identification, implementation, and open-loop and closed-loop compensations through the Preisach model inversion are conducted. Chapter 3 presents a direct method to implement the Preisach model without adopting memory interface. Related simulations for implementation and compensation are also carried out. This method saves computation cost extremely, and is more suitable for real time control. Chapter 4 reviews the definition of the KP model and defines the elementary compensator of the KP model. It also reveals the relationship between the KP model and the Preisach model which in turn guarantees the KP model has the same properties as the Preisach model. Simulations for identification, implementation, and open-loop and closed-loop compensations through the KP model inversion are conducted. Chapter 5 describes the adaptive inverse KP model for the model reference control for linear systems with unknown actuator hysteresis. Chapter 6 discusses adaptive inverse KP model for the model reference adaptive control for linear systems with unknown input hysteresis. Chapter 7 designs adaptive robust controller for nonlinear systems with unknown input hysteresis using a compensator from the KP model. Related robust stability analysis for robust trajectory tracking has been performed. Chapter 8 newly defines a hysteresis model in order to overcome some shortcomings of the Preisach model and the KP model. Simulations are also conducted to control systems with hysteresis through this modified model. Chapter 9 summarizes the dissertation.

CHAPTER 2

MODELING AND COMPENSATION OF HYSTERESIS BY THE CLASSICAL PREISACH MODEL

In this chapter, the classical Preisach model and its properties, and the method to implement the Preisach model will be explained. In order to cancel hysteresis nonlinearity, an inverse hysteresis model based on the Preisach model will be developed. Also, some calculations and simulations will be performed to verify and to illustrate the method and algorithm presented in this chapter.

2.1 Introduction to the Preisach Operator

2.1.1 Definition of the Preisach Model

The classical Preisach model was first introduced by F. Preisach in his landmark paper [39] published in 1935. Because the Preisach*'s* approach was purely intuitive, which was based on some hypotheses concerning the physical mechanisms of magnetization, it was first regarded as a physical model of hysteresis. The insight of the model was revealed mathematically for better understanding by the Russian mathematician M. Krasnoselskii [12] and his colleagues in the 1970s when they undertook a comprehensive mathematic study of systems with hysteresis. After that, it was gradually realized that the Preisach model contained a new general mathematical idea, and was abstracted from its physical connotation to be expressed in a purely mathematical form.

The purely mathematical description of the Preisach model can be considered as an operator which integrates infinite weighted elementary hysteresis operators $\hat{\gamma}_{\alpha\beta}$ over a 2-dimensional region. Namely, it can be expressed by

$$u(t) = H[v(t)] = \iint\limits_{v^{+} \geq \alpha \geq \beta \geq v^{-}} \mu(\alpha,\beta)\hat{\gamma}_{\alpha\beta}[v(t)]d\alpha d\beta \,. \tag{2.1}$$

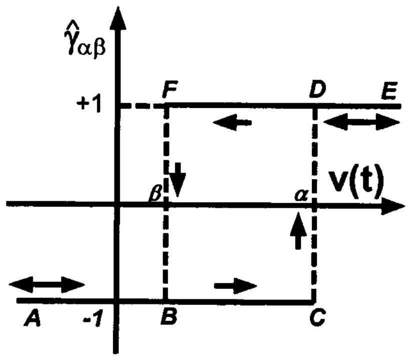


Fig.2.1 An elementary operator (a relay) of Preisach model

Here $H(\cdot)$ is an operator to transform an input $v(t)$ into a hysteresis output $u(t)$; $\mu(\alpha,\beta)$ is a weighting function (densities) that is often referred as the Preisach function. Each of the elementary operators $\hat{\gamma}_{\alpha\beta}[v(t)]$ can be represented by a relay on the input-output diagram (see Fig.2.1). Its parameters α and β correspond to "on" and "off' switching thresholds of input applied on the relay, respectively. From the physical point of view, it is always assumed that α is larger than or equal to β, i.e., $\alpha \geq \beta$, which defines a 2-dimensional region of input. Different sets of parameters (α,β) determine different relays. It is assumed that each relay has only two saturated output values as $\hat{\gamma}_{\alpha\beta}[v(t)] = +1$ and $\hat{\gamma}_{\alpha\beta}[v(t)] = -1$. As the input $v(t)$ monotonically increases from a value less than its lower threshold β towards its upper threshold α, the output of the relay remains in its negative saturation state ($\hat{\gamma}_{\alpha\beta}[v(t)] = -1$). If the input $v(t)$ of the relay continues to increase, the output of the relay switches to its positive saturation state ($\hat{\gamma}_{\alpha\beta}[v(t)] = +1$) when $v(t) = \alpha$, and remains in this state while $v(t) > \alpha$. Thus, the monotonic increases of $v(t)$ forms the ascending branch $ABCDE$ of the relay (see Fig.2.1). Similarly, while the input monotonically decreases, the descending branch $EDFBA$ is traced. It is apparent that each relay represents hysteresis nonlinearity with local memory.

The expression (2.1) of the Preisach model can be interpreted as a parallel connection of infinite weighted relays with positive/negative saturation values as $+1/-1$ (see Fig.2.2). This means that the complex Preisach hysteresis nonlinearity (2.1) is represented as a superposition of simple elementary hysteresis nonlinearities (relays) with local memories; nevertheless, the Preisach hysteresis usually has a non-local memory. It is noticed that the Preisach model has been defined without any reference to a particular physical origin of hysteresis, and it is a phenomenon model with mathematical generality to describe a large number of hysteresis nonlinearities.

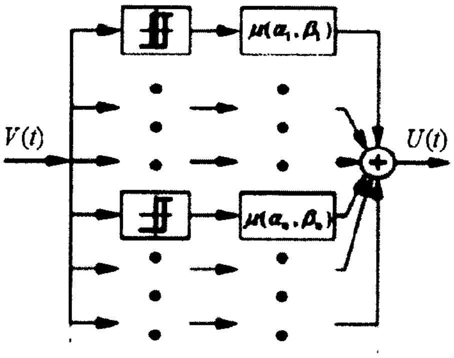


Fig. 2.2 Structural interpretation of the Preisach model

2.1.2 Notations and Geometric Interpretation of the Preisach Model

Preisach Plane T

According to the definition of the Preisach model, every elementary operator has two parameters: α and β, corresponding to "on" and "off" switching thresholds of a relay, and they are always assumed to be $\alpha \geq \beta$. From this fact, a half plane above the line $\alpha = \beta$ in the $\alpha - \beta$ diagram can be constructed. Usually, there exists a limiting input region $[v^-, v^+]$ associated with certain hysteresis nonlinearity. This means that the parameters, α and β, of elementary operators can only vary among the region $[v^-,v^+]$, i.e., $v^- \le \beta \le \alpha \le v^+$, to create hysteresis loops. If the parameters α and β arrive to the boundaries of the input region $[v^-,v^+]$, i.e., $\alpha = \beta = v^-$ or $\alpha = \beta = v^+$, the ascending limiting curve and descending limiting curve will merge together, and the hysteresis nonlinearity degenerates as a step function. Thus, there exist a horizontal line $\alpha = v^+$ and a vertical line $\beta = v^-$ together with the line $\alpha = \beta$ to form a triangular half plane T, which is referred as the Preisach plane of hysteresis (see Fig.2.3). Every point in the triangular domain with a specified pair of coordinates (β,α) corresponds to a particular operator $\hat{\gamma}_{\alpha\beta}[v(t)]$. Furthermore, there is a weighting factor $\mu(\alpha,\beta)$ associated with each point (β,α) on the Preisach plane T.

From the above remarks, it can be realized that the hysteresis only occurs on the Preisach plane T, and the hysteresis gets saturated when the input exceeds the region T . This is guaranteed by setting the weighting factor $\mu(\alpha,\beta)$ to zero for all points outside the Preisach plane T.

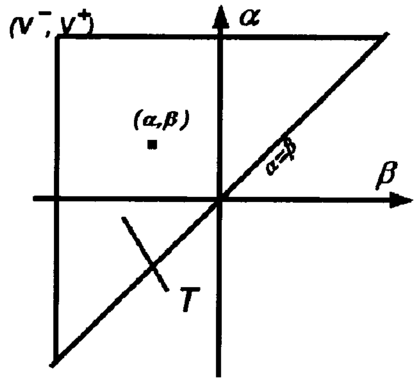


Fig. 2.3 The Preisach plane T

Memory Interface Curves $L(t)$: *The varying of input* $v(t)$ *in the region* $[v^-,v^+]$ *forth* *and back creates the hysteresis loops, and forms a staircase memory interface* $L(t)$ *on its Preisach plane T to divide the region T into two domains, where the relays have two uniform output distributions of* $+1$ *and* -1.

To explain what the memory interface is and how it forms, they can be illustrated by examining the outputs of all relays on the Preisach plane T caused by variations of their input $v(t)$. First, assume that the input $v(t)$ varies starting from a value $v(t_0) \le v^-$. At this time t_0, all elementary operators $\hat{\gamma}_{\alpha\beta}[v(t_0)]$ are subject to the input $v(t_0) \le v^-$, which sets all relays to their negative saturation states, i.e., $\hat{\gamma}_{\alpha\beta}[v(t_0)] = -1$. As the input $v(t)$ monotonically increases to value $v(t_1)$ at time t_1, some relays with "on" switching values $\alpha \le v(t_1)$ are turned into their positive saturation states, i.e., $\hat{\gamma}_{\alpha\beta}[v(t_1)] = +1$ with $\alpha \le v(t_1)$, while others still remain in their negative saturation states, i.e., $\hat{\gamma}_{\alpha\beta}[v(t_1)] = -1$ with $\alpha > v(t_1)$. This increase of input results in a horizontal interface line $L(t): \alpha = v(t_1)$ dividing the Preisach plane T into two domains: $S^+(t_1)$ consisting of points with $\hat{\gamma}_{\alpha\beta}[v(t_1)] = +1$ and $S^-(t_1)$ including points with $\hat{\gamma}_{\alpha\beta}[v(t_1)] = -1$ (see Fig.2.4). If the input $v(t)$ had continued to increase so that $v(t) \ge v^+$, all the relays of the Preisach model would reach their positive saturation states, and the set $S^-(t)$ would degenerate to be nil, while the set $S^+(t)$ would develop to the whole Preisach plane T. At this time, the memory interface $L(t)$ would change as the upper boundary of the Preisach plane T, i.e. , $L(t): \alpha = v^+$.

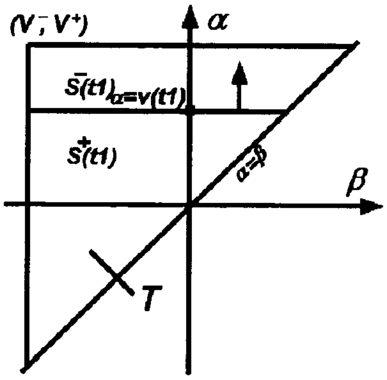


Fig. 2.4 Subdivision of the Preisach plane T by an increasing input $v(t_1)$

Next, assume that input $v(t)$ is monotonically decreasing until it reaches an input value $v(t_2) \leq v(t_1)$ at a moment of time t_2 (see Fig.2.5). At this time, some operators $\hat{\gamma}_{\alpha\beta}[v(t_2)]$ with "off" switching values $\beta > v(t_2)$ change into their negative saturation states. This decrease of the input updates the previous dividing of the Preisach plane T into two new domains: positive saturation domain $S^+(t_2)$ and negative saturation domain $S^-(t_2)$. The memory interface between the two domains $S^+(t_2)$ and $S^-(t_2)$ now consists of a horizontal line segment $a = v(t_1)$ and a vertical line segment $\beta = v(t_2)$, and has a vertex with the coordinates $(v(t_1), v(t_2))$ on it. The vertical line segment moves leftwards, and its motion is specified by the equation $\beta = v(t_2)$. If the input $v(t)$ had continued to decrease so that $v(t) \leq v-$, the motion of the vertical line segment would terminate, and all relays would reach their negative saturation states. At this time, the memory interface $L(t_2)$ would degenerate as the left boundary $L(t): \beta = v-$.

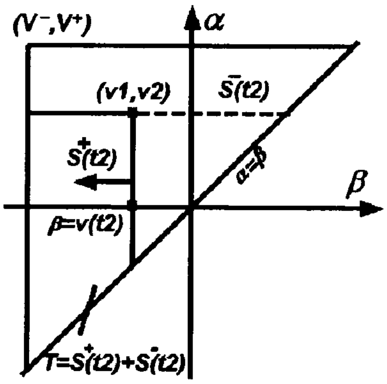


Fig. 2.5 Subdivision of the Preisach plane T by a decreasing input $v(t_2)$ followed an increasing input $v(t_1)$

Continuing assume next that the input increases again until it reaches a new value $v(t_3)$ satisfying condition $v(t_2) \leq v(t_3) \leq v(t_1)$ at an instant of time t_3. The increasing input towards $v(t_3)$ results in the forming of a new horizontal interface line segment $\alpha = v(t_3)$, which moves upwards as the input $v(t)$ continues to increase (see Fig.2.6). At this moment, the memory interface consists of three segments as $a = v(t_1)$, $\beta = v(t_2)$ and $\alpha = v(t_3)$, and a new vertex as $(v(t_3), v(t_2))$.

Continuing again, assume that the input decreases again until it reaches a smaller value $v(t_4)$ satisfying the constraint $v(t_2) \leq v(t_4) \leq v(t_3)$ at the moment of time t_4. This leftward motion adds a new vertex of interface with the coordinates $(v(t_3), v(t_4))$ (see Fig.2.7). At this time, the memory interface consists of four segments: $a = v(t_1)$, $\beta = v(t_2)$, $\alpha = v(t_3)$ and $\beta = v(t_4)$.

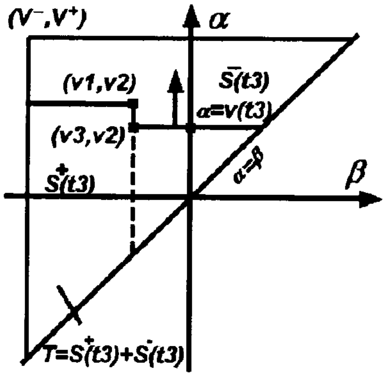


Fig. 2.6 Subdivision of the Preisach plane T by an input sequence $\{ v(t_1), v(t_2), v(t_3) \}$ satisfying $v(t_2) \le v(t_3) \le v(t_1)$

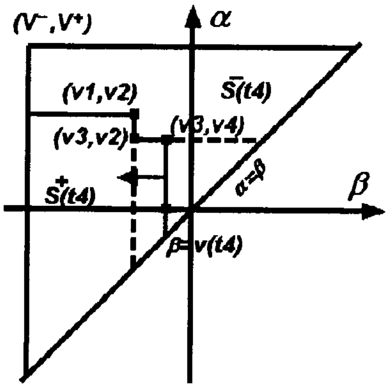


Fig. 2.7 Subdivision of the Preisach plane by an input sequence $\{ v(t_1), v(t_2), v(t_3), v(t_4) \}$

Generalizing the above analysis, the following conclusions can be drawn. At any instant of time t, the triangular Preisach plane T is divided into two sets: $S^+(t)$ consisting of points (α, β) with $\hat{\gamma}_{\alpha\beta}[v(t)] = +1$ and $S^-(t)$ including points (α, β) with $\hat{\gamma}_{\alpha\beta}[v(t)] = -1$. The interface $L(t)$ between $S^+(t)$ and $S^-(t)$ is a staircase whose vertices have α and β coordinates representing respectively the previous local maxima and minima of inputs. The staircase interface $L(t)$ starts from somewhere on the upper or left boundary and stops at somewhere on the line $\alpha = \beta$. This means that the final line segment of the interface $L(t)$ is attached at the line $\alpha = \beta$ and moves along the line as the input $v(t)$ changes. The final line segment is a horizontal one moving upwards as the input increases (see Fig.2.4 and Fig.2.6), and it is a vertical one moving leftwards while the input decreases (see Fig.2.5 and Fig.2.7). The dividing of the Preisach plane T into two domains by the staircase interface is very important because it facilitates the development of a formula to calculation the output value $u(t)$ of the Preisach model, which will be discussed in the following section.

Density Distribution $\mu(\alpha,\beta)$

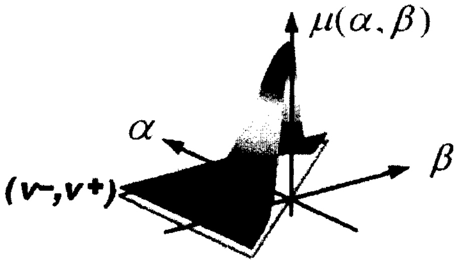


Fig. 2.8 The Preisach function or density distribution

In equation (2.1), the output value $u(t)$ of the hysteresis is expressed as an integral of infinite sets of weighted relays over the Preisach plane T. From the physical point of view, the weight factor $\mu(\alpha,\beta)$ of every relay associated with every point on the Preisach plane T must be a continuous function defined over the plane T (see Fig.2.8). This function is referred to as the Preisach function or the density function of hysteresis. For a particular hysteresis with certain input region $[v^-, v^+]$, the shape and size of the hysteresis loops are only determined by the density distribution $\mu(\alpha,\beta)$. The problem to identify a real hysteresis loop via the Preisach model is to find the density function $\mu(\alpha,\beta)$, and an identification method will be presented in a later section. As long as the input region

$[v^-, v^+]$ and the density function $\mu(\alpha,\beta)$ of the Preisach hysteresis have been specified, its output value $u(t)$ can be calculated by equation (2.1), or its equivalent formula, which will be derived in the next section.

Examining equation (2.1), it can be realized that the density function $\mu(\alpha,\beta)$ must be in a double integral form, which is usually called as a closed form function. However, it is a challenging task to find a closed form density function $\mu(\alpha,\beta)$. Alternatively, a discrete form of density distribution $\mu(\alpha,\beta)$ is sometimes used within required modeling precision.

Formula to Calculate Output Value of Hysteresis

In this section, a formula to calculate the output value $u(t)$ of hysteresis at any instant of time t will be introduced [13].

Facilitated by the definition of memory interface $L(t)$ on the Preisach plane T, at any instant of time t, the integral over the Preisach plane T of equation (2.1) can be divided into two integrals over $S^+(t)$ and $S^-(t)$, respectively, as follows:

$$u(t) = \iint_T \mu(\alpha,\beta)\hat{\gamma}_{\alpha\beta}[v(t)]d\alpha d\beta$$

$$= \iint_{S^+(t)} \mu(\alpha,\beta)\hat{\gamma}_{\alpha\beta}[v(t)]d\alpha d\beta + \iint_{S^-(t)} \mu(\alpha,\beta)\hat{\gamma}_{\alpha\beta}[v(t)]d\alpha d\beta. \qquad (2.2)$$

Since $\quad \hat{\gamma}_{\alpha\beta}[v(t)] = +1$ for points $(\beta,\alpha) \in S^+(t)$ (2.3)

$$\hat{\gamma}_{\alpha\beta}[v(t)] = -1 \text{ for points } (\beta,\alpha) \in S^-(t) \ , \qquad (2.4)$$

and the fact that $u^+ = \iint_T \mu(\alpha,\beta)d\alpha d\beta$, where the subscript T represents 'total', then equation (2.2) can be rewritten as

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

$$= 2\iint_{S^+(t)} \mu(\alpha,\beta)d\alpha d\beta - \iint_T \mu(\alpha,\beta)d\alpha d\beta$$

$$= 2\iint_{S^+(t)} \mu(\alpha,\beta)d\alpha d\beta - u^+ . \tag{2.5}$$

From equation (2.5), it is understood that an instantaneous value of output $u(t)$ depends on dividing the Preisach plane T into positive and negative sets $S^+(t)$ and $S^-(t)$ by a particular staircase interface $L(t)$. The shape of $L(t)$, in turn, depends on the past extremes of inputs because these extremes are the coordinates of the vertices on the staircase interface $L(t)$. Consequently, the history of input of hysteresis before a certain moment t determines the output of hysteresis at the moment. This means that the output of hysteresis at a moment can be memorized by its staircase interface $L(t)$ at the moment. The memory effect of the staircase memory interface $L(t)$ can be verified by the following reasoning.

Assume that two input histories start from the same initial input $v(t_0)$ at initial time t_0 and terminate at the same input $v(t)$ at the last time t, but they have different input histories between time t_0 and t. This assumption leads to two different memory interfaces $L_1(t)$ and $L_2(t)$ (see Fig.2.9). According to equation (2.5), the output values $u_1(t)$ and $u_2(t)$ are obtained, respectively as follows:

$$u_1(t) = \iint_{S_1^+(t)} \mu(\alpha,\beta)d\alpha d\beta - \iint_{S_1^-(t)} \mu(\alpha,\beta)d\alpha d\beta \tag{2.6}$$

and

$$u_2(t) = \iint_{S_2^+(t)} \mu(\alpha,\beta)d\alpha d\beta - \iint_{S_2^-(t)} \mu(\alpha,\beta)d\alpha d\beta . \tag{2.7}$$

They are different output values for the same instant of time even though the two processes start from the identical initial input value $v(t_0)$, and stop at the same input value

$v(t)$ (see Fig.2.10). This means that the output of hysteresis at certain moment t not only depends on the input at that moment but also relies on the input history before that time. This is the property of hysteresis with non-local memory effect. Thus, the expression (2.5) can describe the hysteresis nonlinearities with non-local memories.

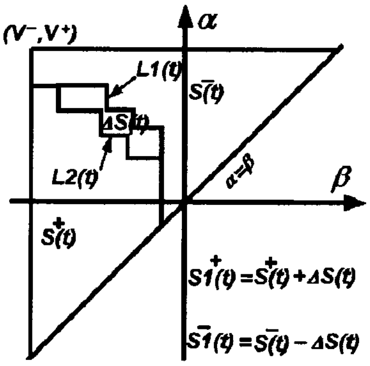


Fig. 2.9 Memory interfaces with same starting inputs and same final inputs

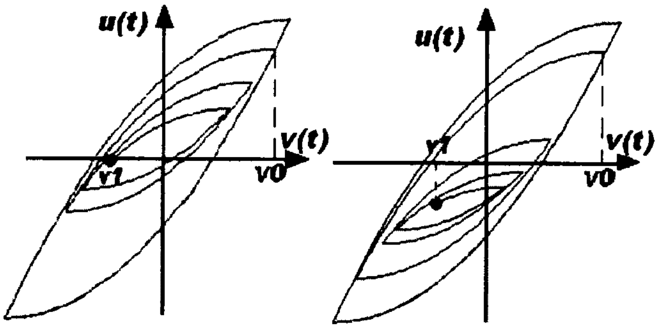


Fig. 2.10 Outputs of different memory interfaces with same starting inputs and same final inputs

Geometric Interpretation

From the above analysis, two memory rules of memory interface $L(t)$ can be summarized by the following statement:

A monotonic increase of input $v(t)$ *leads to a horizontal final line segment of memory interface* $L(t)$ *moving upwards on the Preisach plane* T (see Fig.2.6), *and a monotonic decrease of input* $v(t)$ *results in a vertical final line segment of memory interface* $L(t)$

moving leftwards on the Preisach plane T (see Fig.2.7).

These two different rules result in the formation of the staircase interface $L(t)$, whose vertex coordinates equal to past input extremes.

The formula (2.5) to calculate output values and the above two memory rules for the modification of memory interface $L(t)$, can be interpreted as two independent geometric definitions of the Preisach model. Both definitions are fully equivalent to each other.

2.1.3 Properties of the Preisach Model and Representation Theorem

Having described the geometric interpretation of the Presach model, some main properties of the model can be conveniently revealed as follows:

Symmetry of Output Property

The output value u^+ in the positive saturation state equals to the minus output value u^- in the negative saturation state [13].

Indeed, as input $v(t)$ exceeds the upper limit of the hysteresis input domain, i.e., $v(t) \geq v^+$, all the relays reach their positive saturation states with outputs as $\hat{\gamma}_{\alpha\beta}[v(t)] = +1$. Hence, according to equation (2.5), therefore,

$$u^+ = \iint_T \mu(\alpha,\beta) d\alpha d\beta \tag{2.8}$$

Similarly, in the state of negative saturation, the input $v(t)$ is less than the lower limit of hysteresis input domain as $v(t) \leq v^-$, then all relays shift to negative saturation states with outputs as $\hat{\gamma}_{\alpha\beta}[v(t)] = -1$. This fact results in

$$u^- = -\iint_T \mu(\alpha,\beta) d\alpha d\beta \tag{2.9}$$

From equations (2.8) and (2.9), it is concluded that

$$u^+ = -u^- \tag{2.10}$$

These saturation output values remain constant as u^+ or u^- for any input values above v^+ or below v^-, respectively. In other words, it means that after the limiting ascending branch of the major loop pass the positive saturation state of (v^+,u^+) or the limiting descending branch of the major loop pass the negative saturation state of (v^-,u^-), the hysteresis becomes flat. However, this fact shows one of the shortcomings of the classical Preisach model because it cannot describe reversible components with non-zero slope of hysteresis nonlinearities. This shortcoming can be overcome by introducing a new hysteresis model which will be presented in chapter 8.

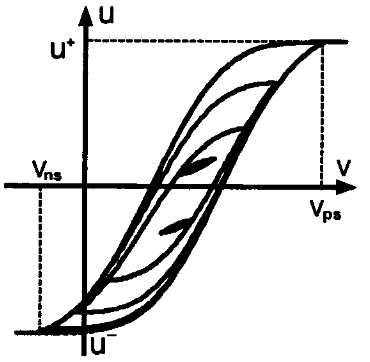


Fig. 2.11 Original coordinate system

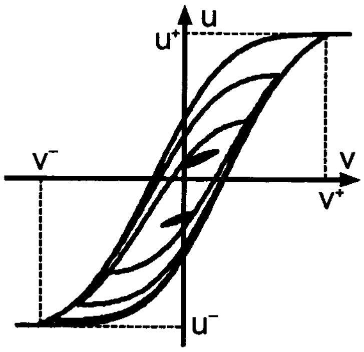


Fig. 2.12 Transformed coordinate system

In reality there are some hysteresis loops whose input values of positive and negative saturation states usually are asymmetrical about the output axis, even though the hysteresis loops are symmetric about the input axis $v(t)$ (see Fig.2.11). For convenience of analysis, coordinate transformations are necessary to change the hysteresis loops to become symmetrical about the output axis $u(t)$ by the following formulas (see Fig.2.12):

$$\begin{cases} v^+ = \dfrac{v_{ps} - v_{ns}}{2} \\ \\ v^- = \dfrac{v_{ns} - v_{ps}}{2} \end{cases}, \qquad (2.11)$$

Here, v_{ps} and v_{ns} are the inputs of positive and negative saturation states.

Wiping-Out Property

Each local input maximum wipes out the vertices of $L(t)$ *whose* $\alpha-$ *coordinates are below this maximum, and each local minimum wipes out the vertices whose* $\beta-$*coordinates are above this minimum* [13].

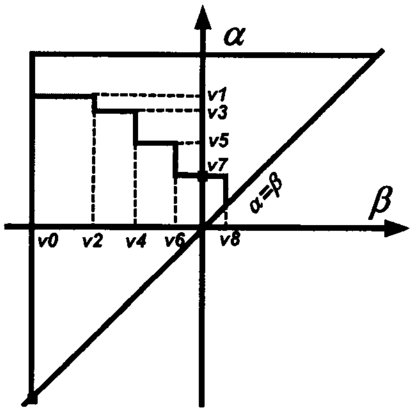


Fig. 2.13 Memory interface line $L(t)$ corresponding to input sequence $\{v_0, v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8\}$

To explain this property clearly, consider a particular input history that is characterized by a finite ascending sequence $\{v_1, v_3, v_5, v_7\}$ of local input maxima and a descending sequence $\{v_0, v_2, v_4, v_6, v_8\}$ of local input minima. A typical $\alpha-\beta$ diagram for this kind of history is shown in Fig.2.13.

Now, assume that the input monotonically increases until it reaches some maximum value v_9 above v_3. This monotonic increase of input results in the forming of a horizontal line segment of interface $L(t)$ that moves upwards until the maximum value v_9 is reached, which modifies the $\alpha-\beta$ diagram as shown in Fig.2.14. It is apparent that all vertices with $\alpha-$ coordinates below v_9 have been wiped out. Equivalently, the input history associated with these wiped out vertices have also been erased. Similarly, the wiping out of vertices occurs for monotonically decrease of input as well. The wiping out occurs due to the further variations of input updating output states of some relays. Indeed, in the above example, as the increasing input approaches to v_9 , all relays with α – coordinates less than v_9 must change their states from negative saturation to positive saturation.

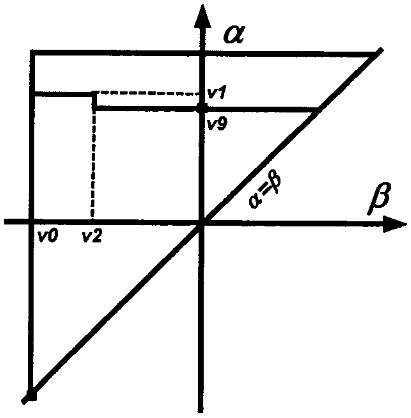


Fig. 2.14 Memory interface line $L(t)$ corresponding to input sequence $\{v_0, v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9\}$

Applying the wiping-out property, an algorithm can be established to determine the memory interface $L(t)$ consisting of global input extremes which are only accumulated by the Preisach model.

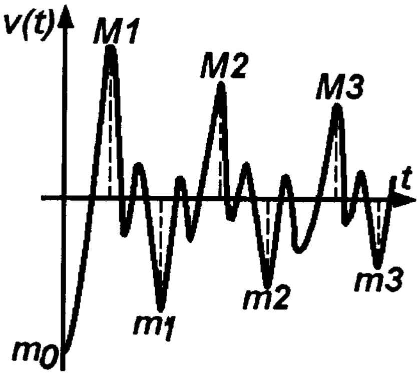


Fig. 2.15 A particular input variation starting from the negative saturation state

Consider a particular input variation initially increasing from v_0 (see Fig.2.15) for the time interval $t_0 \le t \le t'$. The initial input value v_0 was assumed below v^- at the initial instant of time t_0 to ensure all relays start from states of negative saturation. The first global minimum input is $m_0 = v_0 = v(t_0)$ at time $t_0^- = t_0$.

Now, input $v(t)$ first increases and then decreases during the interval $t_0 \le t \le t'$. It is easy to find the global maximum of the input, M_1, at the time t_1^+ as

$$\begin{cases} M_1 = \max v(t) & t \in [t_0, t'] \\ v(t_1^+) = M & t_1^+ \in [t_0, t'] \end{cases} \tag{2.12}$$

It is clear that all previous extremes of inputs between m_0 and M_1 were wiped out by this maximum M_1.

Next, for input $v(t)$ during the period $t_1^+ \le t \le t'$, the global minimum of the input, m_1, at the time t_1^-, can be found as

$$\begin{cases} m_1 = \min v(t) & t \in [t_1^+, t'] \\ v(t_1^-) = m_1 & t_1^- \in [t_1^+, t'] \end{cases} \tag{2.13}$$

It is apparent that all intermediate input extremes between M_1 and m_1 were wiped out by this minimum m_1.

After that, for the interval $t_1^- \le t \le t'$ we can find the global maximum of the input M_2 at the time t_2^+ can be found as

$$\begin{cases} M_2 = \max v(t) & t \in [t_1^-, t'] \\ v(t_2^+) = M_2 & t_2^+ \in [t_1^-, t'] \end{cases} \tag{2.14}$$

It is obvious that all previous input extremes between m_1 and M_2 have been wiped out by this maximum M_2.

As before, for inputs in the interval $t_2^+ \le t \le t'$, the global minimum of the input m_2 at the time t_2^- can be found as

$$\begin{cases} m_2 = \min v(t) & t \in [t_2^+, t'] \\ v(t_2^-) = m_2 & t_2^- \in [t_2^+, t'] \end{cases} \tag{2.15}$$

Continuing the above reasoning, the global maximum M_k and global minimum m_k can be found, respectively, as:

$$\begin{cases} M_k = \max v(t) & t \in [t_{k-1}^-, t'] \\ v(t_k^+) = M_k & t_k^+ \in [t_{k-1}^-, t'] \end{cases} \tag{2.16}$$

and

$$\begin{cases} m_k = \min v(t) & t \in [t_k^+, t'] \\ v(t_k^-) = m_k & t_k^- \in [t_k^+, t'] \end{cases} \tag{2.17}$$

Only the above input extremes are accumulated by the Preisach model, while all intermediate input extremes are wiped out. It can be said that the global maximum M_k and global minimum m_k form an alternating series of dominant maxima and minima.

Based on the above algorithm, an *m. file* program in *Matlab* has been written (see Appendix 1) to calculate the alternating series of dominant extremes for any given input sequence. It will be used to find and draw the memory interface $L(t)$ at any moment of time and to calculate the output value of the hystersis at the moment. This *m. file* will be called as a subroutine by another *m. file* to calculate and draw the hysteresis loops for an arbitrary continuous input.

Congruency Property

All minor hysteresis loops corresponding to back-and-forth variations of inputs between the same two consecutive extreme values are congruent [13].

Let $v_1(t)$ and $v_2(t)$ be two inputs starting from the same instant of time t_0 with two different past histories. When they reach the same extreme v_- or v_+, they have two different memory interfaces as $L_1(t)$ and $L_2(t)$ which divide the Preisach plane into two different sets as $S_1^+(t)/S_1^-(t)$ and $S_2^+(t)/S_2^-(t)$, respectively. They vary back and forth between the same two consecutive extremes v_- and v_+. It can be shown that these periodic input variations result in minor hysteresis loops with the same shape and size.

The proof of the congruency property for the above loops is equivalent to showing that any equal increments of inputs $v_1(t)$ and $v_2(t)$ result in equal increments of outputs $u_1(t)$ and $u_2(t)$. First, assume that both inputs $v_1(t)$ and $v_2(t)$ increase by the same amount: $\Delta v_1 = \Delta v_2 = \Delta v$ starting from the same minimum value v_-. As a result of these increase, the identical triangles ΔT_1 and ΔT_2 are added into the positive sets $S_1^+(t)$ and $S_2^+(t)$ and subtracted from the negative sets $S_1^-(t)$ and $S_2^-(t)$ (see Fig.2.16 and Fig.2.17).

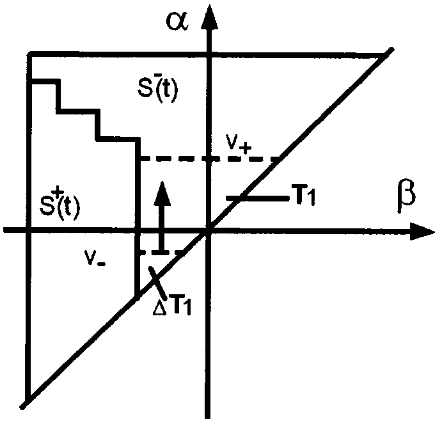


Fig. 2.16 Input with previous memory interface $L_1(t)$ varying inside an interval $[v_-, v_+]$

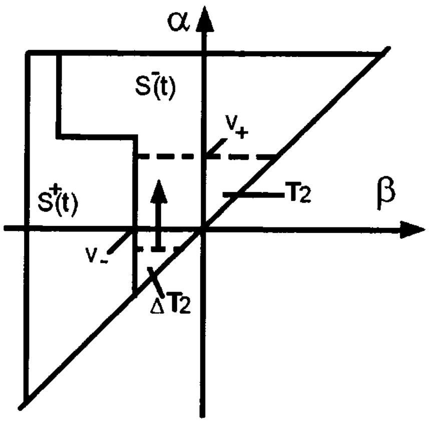


Fig. 2.17 Input with previous memory interface $L_2(t)$ varying inside an interval $[v_-, v_+]$

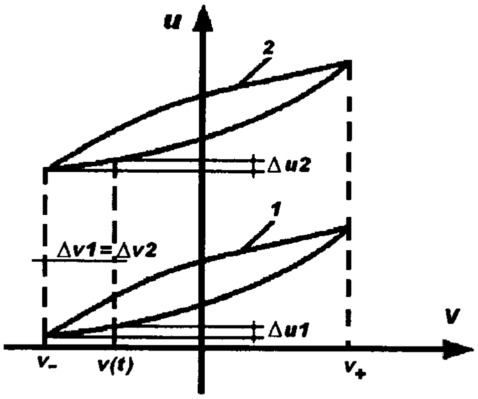


Fig. 2.18 Congruency of two minor loops

By applying equation (2.5), the corresponding output increments due to the two inputs can be calculated as

$$\Delta u_1 = 2\iint_{\Delta T_1} \mu(\alpha,\beta) d\alpha d\beta \tag{2.18}$$

and

$$\Delta u_2 = 2\iint_{\Delta T_2} \mu(\alpha,\beta) d\alpha d\beta\,. \tag{2.19}$$

Since $\Delta T_1 = \Delta T_2$, therefore $\Delta u_1 = \Delta u_2$ (2.20)

Equation (2.20) has been proven for the case when inputs $v_1(t)$ and $v_2(t)$ monotonically increase with the same amount Δv after starting from the same minimum v_-. This equality shows that ascending branches of the above two minor loops are congruent. Similarly, the same result holds when inputs $v_1(t)$ and $v_2(t)$ monotonically decrease with the same amount Δv after achieving the same maximum v_+. This means that the descending branches of the above minor loops are congruent as well. Thus the congruency property of the Preisach model has been proven (see Fig.2.18).

Zero-Initial-Slope Property

All reversal curves of hysteresis loops described by the Preisach model feature zero initial slopes [13].

This statement is based on the following reasoning. Let $u_{\alpha'\beta'}$ be a reversal curve which is traced for a monotonically decreasing input. This curve starts at the point $(v_{\alpha'}, u_{\alpha'})$ which corresponds to the local input maximum $v = \alpha'$. Referring to equation (2.18), the difference $u_{\alpha'} - u_{\alpha'\beta'}$ between two outputs is then given by

$$u_{\alpha'} - u_{\alpha'\beta'} = 2\iint_{T(\alpha',\beta')} \mu(\alpha,\beta) d\alpha d\beta \tag{2.21}$$

Furthermore,

$$u_{\alpha'\beta'} = u_{\alpha'} - 2\iint_{T(\alpha',\beta')} \mu(\alpha,\beta) d\alpha d\beta \; . \tag{2.22}$$

For a fix starting point $(v_{\alpha'}, u_{\alpha'})$, $u_{\alpha'}$ is a constant. Thus, the current slope of this curve at the point $v = \beta'$ can be expressed as

$$\tan\theta(\alpha',\beta') = \frac{\partial u_{\alpha'\beta'}}{\partial\beta'} = -2\frac{\partial \iint_{T(\alpha',\beta')} \mu(\alpha,\beta) d\alpha d\beta}{\partial\beta'} = -2\frac{\partial[\int_{\beta'}^{\alpha}(\int_{\beta}^{\alpha'} \mu(\alpha,\beta) d\alpha) d\beta]}{\partial\beta'}$$

$$= \frac{\partial[\int_{\alpha}^{\beta'}(2\int_{\beta}^{\alpha'} \mu(\alpha,\beta) d\alpha) d\beta]}{\partial\beta'} = 2\int_{\beta}^{\alpha'} \mu(\alpha,\beta) d\alpha \tag{2.23}$$

where $\theta(\alpha',\beta')$ is the angle between the input axis v and the tangent to the reversal curve $u_{\alpha'\beta'}$ at the point $v = \beta'$.

Referring to equation (2.23), for the starting point $v = \alpha'$ of the reversal curve $u_{\alpha'\beta'}$,

$$\lim_{\beta' \to \alpha'} \tan\theta(\alpha', \beta') = 2 \lim_{\beta' \to \alpha'} \int_{\beta'}^{\alpha'} \mu(\alpha, \beta') d\alpha = 0.$$

This property is shown in Fig.2.19, where the slopes $k_1 = k_2 = k_3 = k_4 = 0$. This property is another shortcoming of the Preisach model which will be overcome by introducing a newly defined hysteresis model in the chapter 8.

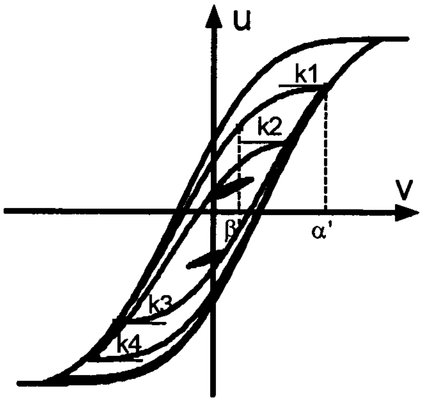


Fig. 2.19 Zero initial slope of hysteresis loops

Representation Theorem

The wiping-out property and the congruency property constitute the necessary and sufficient conditions for a hysteresis nonlinearity to be represented by the Preisach model on the set of piece-wise monotonic inputs [13].

The complete proof can be found in I.D. Mayergoyz's book [13].

2.2 Identification of the Preisach Model

The issue to identify the Preisach model is particularly important in its application to a wide variety of physical systems. From the previous definition of the model it should be understood that determining the output of the Preisach operator is a matter of

- finding the memory interface $L(t)$ on the Preisach plane, which has been discussed in the previous subsections, and

- knowing the density distribution $\mu(\alpha,\beta)$ of the Preisach model.

Thus, in order to use the Preisach model to describe a system with hysteresis, it is necessary to identify the "optimal" parameters $\mu(\alpha,\beta)$ corresponding to data obtained from measurements of hysteresis.

One method to identify the "optimal" densities $\mu(\alpha,\beta)$ is called the "curve fitting method" which is introduced in the following: By assuming that densities $\mu(\alpha,\beta)$ can be expressed by some specific function with a few undetermined parameters, the output values of hysteresis responding to some given inputs can be expressed using the Preisach model with the assumed density $\mu(\alpha,\beta)$. Optimal fitting of the calculated outputs to measured data can determines the parameters of the assumed density function.

For example, in ferromagnetism, the fact that parameter $\mu(\alpha,\beta)$ is seen as a probability distribution of hysterons had motivated Della Torre and his co-authors to assume a particular form for $\mu(\alpha,\beta)$, namely the Gaussian function

$$\mu(\alpha,\beta)=\frac{M_{sat}}{2\pi\sigma_i\sigma_k}\exp\{-\frac{1}{2}[\frac{(\alpha-h_k)^2}{\sigma_k{}^2}+\frac{\beta^2}{\sigma_i{}^2}]\} \qquad (2.24)$$

The beauty of this method is in its simplicity. The measure $\mu(\alpha,\beta)$ has four unknown parameters $(M_{sat},h_k,\sigma_i,\sigma_k)$ which can easily be determined with only a few measurements, and these are standard in experimental magnetism. In fact, only measurements on the limiting branches and one reversal curve are required. Once these parameters are determined the measure $\mu(\alpha,\beta)$ is known over the entire Preisach plane. The curve fitting method can also be used to describe the density function $\mu(\alpha,\beta)$ for hysteresis models of other smart materials such as SMAs.

The second method for approximating the density distribution $\mu(\alpha,\beta)$ is to determine the value of the $\mu(\alpha,\beta)$ at various points in the triangular Preisach plane T and use interpolation to approximate $\mu(\alpha,\beta)$ over the remaining points in the Preisach plane T.

Before moving on to discuss the methods to determine these parameters for $\mu(\alpha,\beta)$, some necessary concepts and formulas are first introduced.

2.2.1 The Discrete Preisach Plane

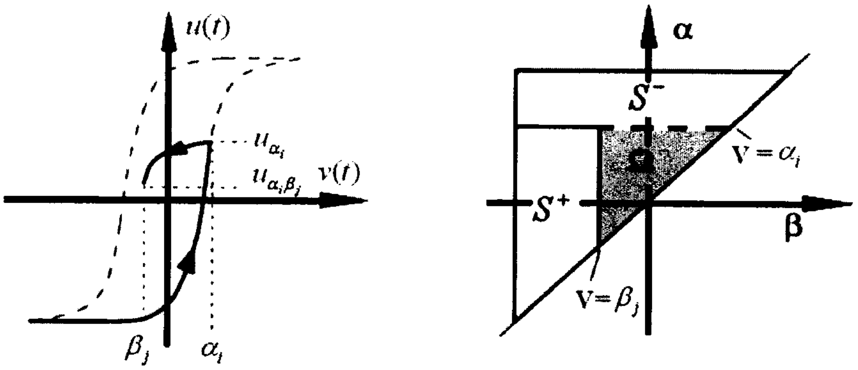


Fig.2.20: A FOD curve and its memory interface line

For any point $p_{ji} = (\beta_j,\alpha_i)$ on the Preisach plane T, it can be considered as a vertex on a memory interface $L(t)$ which is formed by increasing input v from the negative saturation state v^- to a value $v = \alpha_i$ and then decreasing input v to a state as $v = \beta_j$. These variations of input v result in a partition of *the limiting ascending branch* of the hysteresis major loop and a part of *the first-order decreasing curve* (FOD), which attaches on *the limiting ascending branch* at point (α_i,u_{α_i}) in the input-output diagram (see Fig.2.20).

In this manner, different pairs of inputs (β_j,α_i) with $\alpha_i \ge \beta_j$ form different first-order decreasing curves (FOD) inside the hysteresis major loop and divide the Preisach plane T into small cells. By dividing the Preisach plane T uniformly using numbers of horizontal l lines and numbers of vertical l lines respectively, a total number of N nodes on the Preisach plan T as shown in Fig. 2.21 can be created, where

$$N=\frac{(l+2)(l+2)-(l+2)}{2}+(l+2)=\frac{(l+3)(l+2)}{2} \tag{2.25}$$

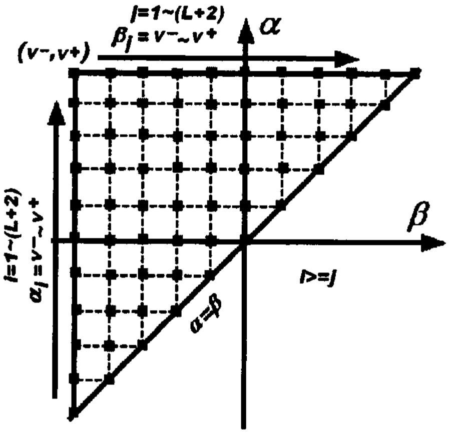
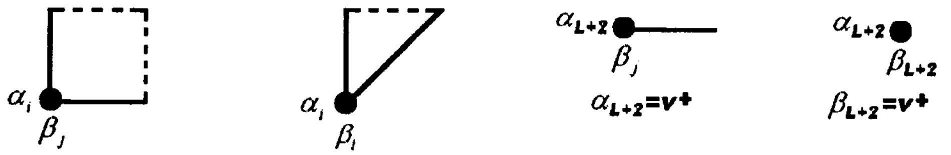


Fig. 2.21 FOD data points in the Preisach plane T

The above dividing of the Preisach plane T also generates four kinds of small cells, as illustrated in Fig.2.22.



Rectangular cell Triangular cell Degenerated cell Point cell

Fig. 2.22 Four kinds of cells in the Preisach plane T

The total number of the small cells also equal to N. Thus, each node represents a left-lower corner of a small cell, and the input pairs (β_j,α_i), coordinates of the ji node, can be defined by following formula:

$$\begin{cases}\beta_j=(j-1)\dfrac{v^+-v^-}{l+1}+v^- \\ \\ \alpha_i=(i-1)\dfrac{v^+-v^-}{l+1}+v^-\end{cases} \quad with \quad i,j=1,2,\ldots,(l+2);\quad i\geq j \tag{2.26}$$

2.2.2 Identification of Density Function by Least Squares Method

In this section the least squares method to determine the optimal density function $\mu(\alpha,\beta)$ of the Preisach model is presented.

To obtain the output $u(t)$, the weighting function $\mu(\alpha,\beta)$ needs to be determined first. This can be done experimentally using a set of first-order descending curves (FOD curves) as shown in Fig.2.20.

With reference to Fig.2.20 and applying equation (2.5), the change in output along the descending branch is obtained as

$$u_{\alpha_i\alpha_i} - u_{\alpha_i\beta_j} = 2 \iint\limits_{S^+ + \Omega_{ij}} \mu(\alpha,\beta)d\alpha d\beta - 2\iint\limits_{S^+} \mu(\alpha,\beta)d\alpha d\beta$$

$$= 2\iint\limits_{\Omega_{ij}} \mu(\alpha,\beta)d\alpha d\beta \tag{2.27}$$

where $u_{\alpha_i\alpha_i}$ refers to the output when the input monotonically increases to α_i from v^- , and $u_{\alpha_i\beta_j}$ refers to the output when the input monotonically decreasing to β_j immediately after the increase of input from v^- to α_i. For all output changes $u_{\alpha\alpha} - u_{\alpha\beta}$ corresponding to all points (β,α) on the Preisach plane T, a formula is obtained as

$$u_{\alpha\alpha} - u_{\alpha\beta} = 2 \iint\limits_{S^+ + \Omega} \mu(\alpha,\beta)d\alpha d\beta - 2\iint\limits_{S^+} \mu(\alpha,\beta)d\alpha d\beta$$

$$= 2\iint\limits_{\Omega} \mu(\alpha,\beta)d\alpha d\beta \tag{2.28}$$

Referring to equation (2.23) and taking partial derivatives of both sides of (2.28) with respect to α and β the value of the weighting function at any point in the Preisach plane T can be determined by the following formula:

$$\mu(\alpha,\beta) = \frac{1}{2}\frac{\partial^2 u_{\alpha\beta}}{\partial\alpha\partial\beta} \tag{2.29}$$

If the output values $u_{\alpha\beta}$ could be identified for all points in T by collecting an infinite number of FOD measures, then the density function $\mu(\alpha,\beta)$ can be determined at these points. With respect to equation (2.29), the surface consisting of all 3-dimensional points $(\beta,\alpha,u_{\alpha\beta})$ should be smooth. This surface, which is called the FOD surface, must be twice differentiable with respect to variables α and β to obtain the weighting function $\mu_{\alpha\beta}$.

To identify the density function $\mu(\alpha,\beta)$ via equation (2.29), however, it is impossible to measure an infinite number of FOD measures to construct the FOD surface. Instead, one can first assume that the FOD surface has a special form governed by a class of functions with some unknown parameters, and then match the candidate surfaces defined by the functions to some measured output values of $u_{\alpha\beta}$ to find these parameters by the least squares method. After obtaining the parameters, the FOD function $u_{\alpha\beta}$ is determined, and then solving the density $\mu(\alpha,\beta)$ is a straight forward application of equation (2.29). Alternatively, one can directly assume that the density function $\mu(\alpha,\beta)$ satisfies a particular class of functions with some parameters required to be determined. And then express the outputs of $u_{\alpha\beta}$ in a function of these parameters by substituting $\mu(\alpha,\beta)$ into equation (2.28). Matching these outputs of $u_{\alpha\beta}$ with the measured output values by the least squares method results in the optimal parameters for the density function $\mu(\alpha,\beta)$. These two methods are discussed in detail as follows:

A. Estimation on the FOD Surface Function to Determine Density Function

Assume that the FOD surface function has a special form as

$$\hat{u}_{\alpha\beta} = f(x_1, x_2, x_3, \ldots, x_n, \alpha, \beta) \tag{2.30}$$

where $x_1, x_2, x_3, \ldots, x_n$ are parameters to be determined by the least squares method. For every point (α_i, β_j), there exists a corresponding point on the FOD surface with a calculated output value as

$$\hat{u}_{\alpha_i\beta_j} = f(x_1, x_2, x_3, \ldots, x_n, \alpha_i, \beta_j) \tag{2.31}$$

and a measured output value as $u_{\alpha_i\beta_j}$.

To find a density function suitable for all points on the Preisach plane, first mesh the plane by l horizontal lines and l vertical lines uniformly resulting in N nodes as in Fig.2.21. Then for every point (α_i, β_j), make a measure of $u_{\alpha_i\beta_j}$ and calculate $\hat{u}_{\alpha_i\beta_j}$ by equation (2.31). There is a square error between these two values for every point as

$$|\hat{u}_{\alpha_i\beta_j} - u_{\alpha_i\beta_j}|^2 = |f(x_1, x_2, x_3, \ldots, x_n, \alpha_i, \beta_j) - u_{\alpha_i\beta_j}|^2 \tag{2.32}$$

The sum of all square errors for all points on the Preisach plane is a function of parameters $x_1, x_2, x_3, \ldots, x_n$, given as

$$\sum_{i=1}^{l+2}\sum_{j=1}^{i} |\hat{u}_{\alpha_i\beta_j} - u_{\alpha_i\beta_j}|^2 = \sum_{i=1}^{l+2}\sum_{j=1}^{i} |f(x_1, x_2, x_3, \ldots, x_n, \alpha_i, \beta_j) - u_{\alpha_i\beta_j}|^2$$

$$= e(x_1, x_2, x_3, \ldots, x_n) \geq 0. \tag{2.33}$$

Equation (2.33) can be expressed in terms of the vectors $X = [x_1, x_2, x_3, \ldots, x_n]$, $F(X) = [f_1(X), f_2(X), \ldots, f_N(X)]$ and $U = [u_1, u_2, u_3, \ldots, u_N]$ as

$$e(x_1, x_2, x_3, \ldots, x_n) = |F(X) - U|^2 \tag{2.34}$$

where N is calculated by (2.25).

Minimizing the function (2.34) yields parameters $X = [x_1, x_2, x_3, \ldots, x_n]$ which leads to the best fitting of the measured data to the surface described by the assumed function. This matching can be implemented by calling an existing *Matlab* function: *lsqnonneg (F(X),U)*. After obtaining the FOD function $\hat{u}_{\alpha\beta} = f(x_1, x_2, x_3, \ldots, x_n, \alpha, \beta)$, it is substituted into equation (2.29) to calculate the density function $\mu(\alpha, \beta)$.

B. Direct Estimation on the Density Function

Different from the method to estimate the FOD function, the second approach is to directly assume that the density function has a special form as

$$\hat{\mu}(\alpha,\beta) = g(x_1', x_2', x_3', \ldots, x_n', \alpha, \beta) \tag{2.35}$$

where $x_1', x_2', x_3', \ldots, x_n'$ are parameters need to be determined by the least squares method. The output vales $\hat{u}_{\alpha\beta}$ for every point on the FOD surface can be expressed as a function of the parameters $X' = [x_1', x_2', x_3', \ldots, x_n']$, also the actual values $u_{\alpha\beta}$ can be measured by experiments. Similar to approach A, the parameters $X' = [x_1', x_2', x_3', \ldots, x_n']$ can be solved by the least squares method.

Remark: the above two procedures are considered curve fitting methods, and there are very restrictive limits on the assumption of the expressions to represent the FOD surface $u_{\alpha\beta}$ or the density function $\mu(\alpha, \beta)$ because they must be in closed form. This implies that the functions must be twice differentiable or in double-integral form. Otherwise, the density function cannot be obtained from the optimal FOD function, or the FOD values cannot be calculated from the assumed density function. This limitation restricts the curve fitting method from being applied to a wide variety of physical systems.

2.2.3 Identification FOD Surface by Interpolation Method

The second method for identification of the Preisach model is to identify the FOD surface

$u_{\alpha\beta}$, rather than to identify the density function $\mu_{\alpha\beta}$. This method is resulted from the fact that the FOD distribution $u_{\alpha\beta}$ describes the effect of $\mu_{\alpha\beta}$ on the hysteresis operator. Also, the interpolation method avoids making twice partial differential of experimentally determined quantities to obtain density function $\mu_{\alpha\beta}$ because the operation of twice partial differential can greatly magnify any error introduced in the collection of data. This fact can be understood from (2.29). One can only measure $u_{\alpha\beta}$ at nodes of the Preisach plane *T*, and then use interpolation method to find $u_{\alpha\beta}$ for any points in the plane *T*. The interpolation method for determining the FOD distribution $u_{\alpha\beta}$ can be explained as bellow.

As long as the experimentally measured data $u_{\alpha_i\beta_j}\,(i \geq j = 1,2,...,l+2)$ for all discrete nodes on the Preisach plane are known, the data $u_{\alpha_i\beta_j}$ of most points which do not coincide with those discrete nodes could be approximated through interpolation. If a point (β,α) is located inside a rectangular cell formed by points of(β_j,α_i), (β_{j+1},α_i), $(\beta_{j+1},\alpha_{i+1})$, and (β_j,α_{i+1}) or a triangle cell with vertices as (α_i,α_i),$(\alpha_{i+1},\alpha_{i+1})$ and(α_i,α_{i+1}), outputs of which are experimentally measured already (see Fig.2.23), the $u_{\alpha\beta}$ of the point (β,α) can be determined by the following formulas:

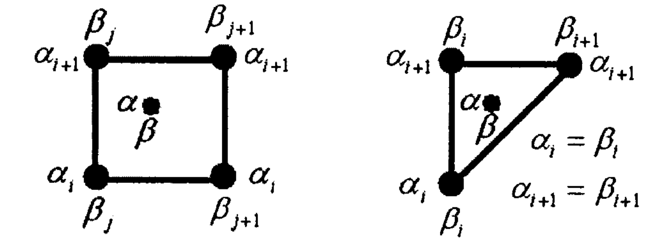


Fig. 2.23 Interpolation the DOF outputs of points

If $(\alpha-\alpha_i)(\alpha-\alpha_{i+1})\le 0$ and $(\beta-\beta_j)(\beta-\beta_{j+1})\le 0$ then the point (β,α) locates inside a rectangle cell with vertices at (β_j,α_i), (β_{j+1},α_i), $(\beta_{j+1},\alpha_{i+1})$ and (β_j,α_{i+1}), the output of the point (β,α) can be interpolated by a formula as

$$u_{\alpha\beta}=\frac{(\alpha-\alpha_i)(\beta-\beta_j)}{(\alpha_i-\alpha_{i+1})(\beta_j-\beta_{j+1})}u_{\alpha_{i+1}\beta_{j+1}}-\frac{(\alpha-\alpha_i)(\beta-\beta_{j+1})}{(\alpha_i-\alpha_{i+1})(\beta_j-\beta_{j+1})}u_{\alpha_{i+1}\beta_j}$$

$$-\frac{(\alpha-\alpha_{i+1})(\beta-\beta_j)}{(\alpha_i-\alpha_{i+1})(\beta_j-\beta_{j+1})}u_{\alpha_i\beta_{j+1}}+\frac{(\alpha-\alpha_{i+1})(\beta-\beta_{j+1})}{(\alpha_i-\alpha_{i+1})(\beta_j-\beta_{j+1})}u_{\alpha_i\beta_j} \qquad (2.36)$$

If $(\alpha-\alpha_i)(\alpha-\alpha_{i+1})\le 0, \alpha_i=\beta_i$ and $\alpha_{i+1}=\beta_{i+1}$, then the point (β,α) locates inside a triangular cell with vertices at $(\alpha_i,\alpha_i),(\alpha_{i+1},\alpha_{i+1})$ and (α_i,α_{i+1}) , therefore

$$u_{\alpha\beta}=\frac{(\alpha_{i+1}-\alpha_i)(\alpha_{i+1}-\alpha)}{(\alpha_{i+1}-\alpha_i)^2}u_{\alpha_i\alpha_i}+\frac{(\alpha_i-\alpha_{i+1})(\alpha_i-\beta)}{(\alpha_{i+1}-\alpha_i)^2}u_{\alpha_{i+1}\alpha_{i+1}}+\frac{(\alpha_{i+1}-\alpha_i)(\alpha-\beta)}{(\alpha_{i+1}-\alpha_i)^2}u_{\alpha_{i+1}\alpha_i} \qquad (2.37)$$

The method described in this section allows for a much more general determination of the parameters and therefore is applicable to a greater number of physical systems. However, it requires the collection of much more data than the curve fitting method described above.

2.3 Numerical Implementation of the Preisach Model

Having the FOD function or the density function, theoretically, the output of the Preisach hysteresis can be directly calculated by equation (2.1) or its equivalent formula (2.5). However, the $\hat{\gamma}_{\alpha\beta}$ in equation (2.1) and knowledge of the integral domain $S^+(t)$ are required to determine the memory interface line segments of $L(t)$. In practice, the output of the Preisach hysteresis are determined through the implementation of numerical implementation methods.

Recall equation (2.5): $u(t) = \iint_{S^+(t)} \mu(\alpha,\beta) d\alpha d\beta - \iint_{S^-(t)} \mu(\alpha,\beta) d\alpha d\beta$

$$= 2\iint_{S^+(t)} \mu(\alpha,\beta) d\alpha d\beta - \iint_{T} \mu(\alpha,\beta) d\alpha d\beta$$

Substituting equation (2.8), $u^+ = \iint_{T} \mu(\alpha,\beta) d\alpha d\beta$, into (2.5) yields

$$u(t) = 2\iint_{S^+(t)} \mu(\alpha,\beta) d\alpha d\beta - u^+$$

Substituting equation (2.10), $u^+ = -u^-$, into the above equation obtains

$$u(t) = 2 \iint_{S^+(t)} \mu(\alpha,\beta) d\alpha d\beta + u^- \qquad (2.38)$$

If the last variation of the input is decreasing, the memory interface $L(t)$ divides $s^+(t)$ into n numbers of trapezoids Q_k with $k = 1,2,...,n$ (see Fig.2.24). Q_n has the right-upper vertex of $(\beta,\alpha) = (m_n, M_n) = (v(t), M_n)$. For each Q_k , referring to Fig.2.20, and considering equation (2.28), $u_{\alpha\alpha} - u_{\alpha\beta} = 2\iint_{\Omega} \mu(\alpha,\beta) d\alpha d\beta$, the result for one single trapezoid is

$$\iint_{Q_k} \mu(\alpha,\beta) d\alpha d\beta = \frac{u_{M_k M_k} - u_{M_k m_{k-1}}}{2} - \frac{u_{M_k M_k} - u_{M_k m_k}}{2} = \frac{u_{M_k m_k} - u_{M_k m_{k-1}}}{2} \qquad (2.39)$$

Summation of all trapezoids underneath the memory interface $L(t)$ in Fig.2.24 gives

$$\iint_{S^+(t)} \mu(\alpha,\beta) d\alpha d\beta = \sum_{k=1}^{n} \frac{u_{M_k m_k} - u_{M_k m_{k-1}}}{2} \qquad (2.40)$$

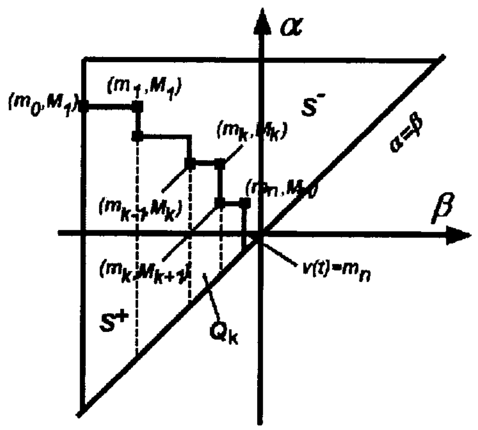


Fig.2.24 Memory interface with increase final input and corresponding trapezoids in the Preisach plane T

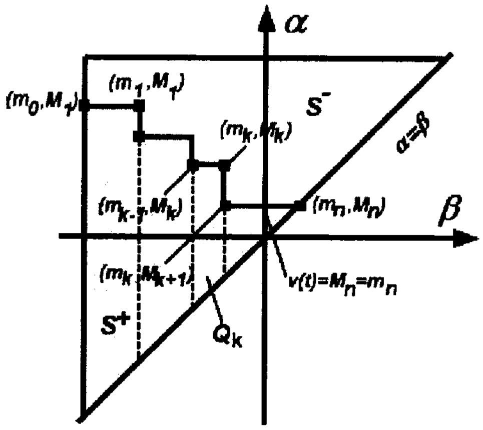


Fig.2.25 Memory interface with a final increase input and corresponding trapezoids in the Preisach plane T

Substituting equation (2.40) into (2.38) yields

$$u(t)=\sum_{k=1}^{n}(u_{M_k m_k}-u_{M_k m_{k-1}})+u^- \tag{2.41}$$

If the last variation of the input is increasing, then in addition to those n trapezoids, $s^+(t)$ has a triangular part Q_n (see Fig.2.25). If it is assumed that input $v(t)$ at the final time has a decreasing trend but have not begun to reduce the area of the triangular part Q_n, then a new final vertex can be defined with coordinates as $(m_n,M_n)=(\beta,\alpha)=(v(t),v(t))$ on the memory interface line. Based on this assumption, equations (2.39) to (2.41) are still valid since $m_n = M_n = v(t)$. Thus, the numerical Preisach model for both cases with final increasing input and final decreasing input can be written as equation (2.41).

With $u_{M_1 m_0} = u_{M_1 v^-} = u^- = -u^+$, some special cases can be verified as follows:

Case1: When $n = 1$, equation (2.41) becomes

$$u(t) = -u^+ + (u_{M_1 m_1} - u_{M_1 m_0}) = -u^+ + u_{M_1 m_1} + u^+ = u_{M_1 m_1} \tag{2.42}$$

Case 2: From the above case 1, if $v(t) = v^+$, then $M_1 = m_1 = v^+$, and thus equation (2.42) becomes

$$u(t) = u_{M_1 m_1} = u_{v^+ v^+} = u^+ \tag{2.43}$$

Case3: From the above case 1, after $v(t)$ increases to an arbitrary value above the input of the negative saturation state v^-, it decreases finally to arrive at the negative saturation state as $v(t) = v^-$, then $m_1 = v^-$, and thus equation (2.42) becomes

$$u(t) = u_{M_1 m_1} = u_{M_1 v^-} = u^- \tag{2.44}$$

Remarks: when a continuous FOD surface function has been identified through matching a candidate function to experimental data, equation (2.41) can directly be used to calculate the output of hysteresis subjected to an input value at the end of any sequence of inputs. As the input is given as a continuous process of time, the corresponding output as a continuous process can be obtained, and furthermore, the hysteresis loops can be draw in the input-output diagram.

However, sometimes, the candidate functions for the FOD surface or density distribution are difficult to predict in advance even though some precise output data on the FOD surface can be measured. In this situation, the numerical implementation of formula (2.41) is still applicable, and is implemented through interpolation methods. The finer the Preisach plane T is divided (the larger l is adopted), then the results are more precise for the calculations of $u_{M_k m_k}$ and $u_{M_k m_{k-1}}$ from equation (2.41) which are obtained through the above two interpolation formulas (2.36) and (2.37). Thus, the hysteresis outputs are more precisely modeled by equation (2.41).

2.4 Hysteresis Compensation through Inverse Presiach Model

2.4.1 Exact Inverse Preisach Model

After obtaining the numerical expression of the hysteresis, its corresponding inverse numerical model can be derived as in the following procedure:

For $\dot{v}(t) \leq 0$, seeking the inverse hysteresis becomes a problem of finding a decreasing input $v(t) = m_n = \beta$ corresponding to an output $u(t)$ and a given fixed input $\alpha = M_n$ so that

$$\begin{cases} u_{\alpha\beta} = u(t) + u^+ + \sum_{k=1}^{n-1} (u_{M_k m_{k-1}} - u_{M_k m_k}) + u_{M_n m_{n-1}} & for \quad n \geq 2 \\ u_{\alpha\beta} = u(t) & for \quad n = 1 \end{cases} \tag{2.45}$$

Equation (2.45) can be derived as following reasoning:

From (2.41), one has

$$u(t) = \sum_{k=1}^{n} (u_{M_k m_k} - u_{M_k m_{k-1}}) + u^- \tag{2.46}$$

For $n = 1$, since $u_{M_1 m_0} = u^-$ and $u_{M_n m_n} = u_{M_1 m_1} = u_{\alpha\beta}$, equation (2.46) is rewritten as

$$u(t) = u_{M_1 m_1} - u_{M_1 m_0} + u^- = u_{\alpha\beta} - u^- + u^- = u_{\alpha\beta} \tag{2.47}$$

For $n \geq 2$, since $u_{M_n m_n} = u_{\alpha\beta}$ and $u^- = -u^+$, equation (2.46) can be rewritten as

$$u(t)=\sum_{k=1}^{n-1}(u_{M_k m_k}-u_{M_k m_{k-1}})+u^{-}+u_{M_n m_n}-u_{M_n m_{n-1}}$$

$$=u_{\alpha\beta}-u^{+}-u_{M_n m_{n-1}}-\sum_{k=1}^{n-1}(u_{M_k m_{k-1}}-u_{M_k m_k}) \qquad (2.48)$$

Rearranging equation (2.48) yields

$$u_{\alpha\beta}=u(t)+u^{+}+u_{M_n m_{n-1}}+\sum_{k=1}^{n-1}(u_{M_k m_{k-1}}-u_{M_k m_k}) \qquad (2.49)$$

Combining equations (2.47) and (2.49) leads to equation (2.45). Thus, the inverse function for decreasing input is obtained as

$$v(t)=u_{\beta}^{-1}(\alpha,u_{\alpha\beta})=\begin{cases} u_{\beta}^{-1}\{M_n,u(t)+u^{+}+\sum_{k=1}^{n-1}(u_{M_k m_{k-1}}-u_{M_k m_k})+u_{M_n m_{n-1}}\} & for \quad n\geq 2,\dot{v}(t)\leq 0 \\ u_{\beta}^{-1}\{M_n,u(t)\} & for \quad n=1,\dot{v}(t)\leq 0 \end{cases} \qquad (2.50)$$

Similarly, for $\dot{v}(t)\geq 0$, the problem of seeking the inverse hysteresis is to find an increasing input $v(t)=\alpha=M_n$ corresponding to a given fixed input $\beta=m_{n-1}$ and an output $u(t)$, so that

$$\begin{cases} u_{\alpha\alpha}-u_{\alpha\beta}=u(t)+u^{+} & for \quad n=1,\dot{v}(t)\geq 0 \\ u_{\alpha\alpha}-u_{\alpha\beta}=u(t)+u^{+}+\sum_{k=1}^{n-1}(u_{M_k m_{k-1}}-u_{M_k m_k}) & for \quad n\geq 2,\dot{v}(t)\geq 0 \end{cases} \qquad (2.51)$$

Equation (2.51) comes from below reasoning:

From (2.41), one has equation (2.46) as

$$u(t)=\sum_{k=1}^{n}(u_{M_k m_k}-u_{M_k m_{k-1}})+u^{-} \qquad (2.46)$$

For $n=1$, since $u_{M_n m_n}=u_{M_1 m_1}=u_{\alpha\alpha}$ and $u_{M_n m_{n-1}}=u_{M_1 m_0}=u_{\alpha\beta}=u_{\alpha v-}=u^{-}$, equation (2.46) can be rewritten as

$$u(t)=u_{M_1 m_1}-u_{M_1 m_0}+u^{-}$$

$$= u_{\alpha\alpha} - u_{\alpha v-} + u^{-}$$

$$= u_{\alpha\alpha} - u_{\alpha\beta} + u^{-} = u_{\alpha\alpha} \tag{2.52}$$

This situation means that the input-output points of hysteresis loop goes along the ascending limiting curve starting from the negative saturation state of the hysteresis.

For $n \geq 2$, equation (2.46) is rewritten as

$$u(t) = \sum_{k=1}^{n-1} (u_{M_k m_k} - u_{M_k m_{k-1}}) + u^{-} + u_{M_n m_n} - u_{M_n m_{n-1}} \tag{2.53}$$

In this case with a final ascending input, since $u^{-} = -u^{+}, u_{\alpha\beta} = u_{M_n m_{n-1}}$, and $u_{\alpha\alpha} = u_{M_n m_n}$ (see Fig.2.25), equation (2.53) becomes

$$u(t) = \sum_{k=1}^{n-1} (u_{M_k m_k} - u_{M_k m_{k-1}}) - u^{+} + u_{\alpha\alpha} - u_{\alpha\beta} \tag{2.54}$$

Combining equations (2.52) and (2.54) leads to equation (2.51). By defining $F_{\alpha\beta} \overset{\Delta}{=} u_{\alpha\alpha} - u_{\alpha\beta}$ and substituting $F_{\alpha\beta}$ into equations (2.51), one has

$$\begin{cases} F_{\alpha\beta} = u(t) + u^{+} & for \quad n = 1, \dot{v}(t) \geq 0 \\ F_{\alpha\beta} = u(t) + u^{+} + \sum_{k=1}^{n-1} (u_{M_k m_{k-1}} - u_{M_k m_k}) & for \quad n \geq 2, \dot{v}(t) \geq 0 \end{cases} \tag{2.55}$$

Thus, the inverse function for increasing input is obtained as

$$v(t) = F_{\alpha}^{-1}(F_{\alpha\beta}, \beta) = F_{\alpha}^{-1}(F_{\alpha\beta}, m_{n-1})$$

$$= \begin{cases} F_{\alpha}^{-1}\{u(t) + u^{+}, m_{n-1}\} & for \quad n = 1, \dot{v}(t) \geq 0 \\ F_{\alpha}^{-1}\{[u(t) + u^{+} + \sum_{k=1}^{n-1} (u_{M_k m_{k-1}} - u_{M_k m_k})], m_{n-1}\} & for \quad n \geq 2, \dot{v}(t) \geq 0 \end{cases} \tag{2.56}$$

This numerical inverse Preisach model, (2.50) and (2.56), can be converted to an *s-function* in *Matlab* for real time control.

2.4.2 Approximated Inverse Preisach Model

The FOD surface sometimes can only be determined by interpolation methods since the function $u_{\alpha\beta}$ can not be assumed as a specific form. Even though the function $u_{\alpha\beta}$ can be expressed by a continuous function, the exact inverse Preisach models, (2.50) and (2.56), sometimes are in implicit forms and can not be solved. Therefore, the exact inverse Preisach model cannot be used to compensate the hysteresis. Instead, an approximate algorithm of the inverse hysteresis model is introduced as in the following:

For a given output $u(\tau)$ at a moment of time τ with known input and output history $v(t<\tau)$, $u(t<\tau)$ of the Preisach hysteresis operator, increase or decrease the input v by steps of idv ,where $i=1,2,\ldots n$, until the calculated output value of the Preisach operator becomes close to the true output ,i.e.,

$$\text{when}\quad \begin{cases} H[v(t<\tau)+ndv]\geq u(\tau) \quad and \quad H[v(t<\tau)+(n-1)dv]\leq u(\tau) \quad for \quad \dot{u}(t)\geq 0 \\ \\ H[v(t<\tau)-ndv]\leq u(\tau) \quad and \quad H[v(t<\tau)-(n-1)dv]\geq u(\tau) \quad for \quad \dot{u}(t)\leq 0 \end{cases} \tag{2.57}$$

thus, interpolate the input $v(\tau)$ for the moment τ as

$$v(t)=\begin{cases} v(t<\tau)+dv\dfrac{u(\tau)-H[v(t<\tau)+(n-1)dv]}{H[v(t<\tau)+ndv]-H[v(t<\tau)+(n-1)dv]} & for \quad \dot{u}(t)\geq 0 \\ \\ v(t<\tau)-dv\dfrac{u(\tau)-H[v(t<\tau)-(n-1)dv]}{H[v(t<\tau)-ndv]-H[v(t<\tau)-(n-1)dv]} & for \quad \dot{u}(t)\leq 0 \end{cases} \tag{2.58}$$

This algorithm can be explained by the following sequence of steps:

1) Pass the present input $v_{present}$ to $v_2=v_{present}$ and the corresponding current output $u_{present}$ to $u_2=u_{present}$.

2) If the process has an increasing trend, namely, $v\geq v_{present}$ and $u\geq u_{present}$ (u increases as v increases, see Fig.2.26),

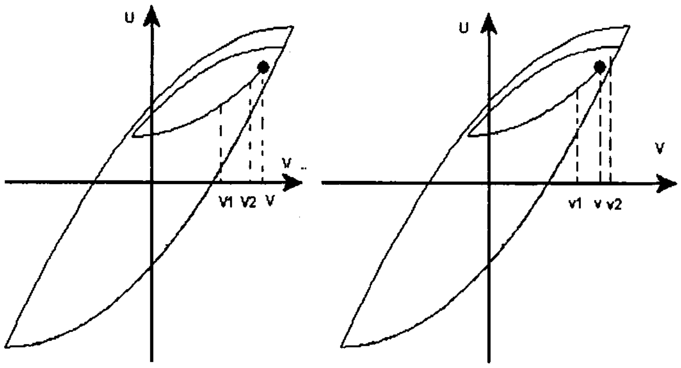


Fig. 2.26 Increasing u

2.1) store the values of v_2 and u_2 into v_1 and u_1 by setting $v_1 = v_2$ and $u_1 = u_2$, and

2.2) update the values of v_2 and u_2 by $v_2 = v_1 + dv$ and $u_2 = H(v_2)$, where the updated output u_2 is calculated by the Preisach operator.

2.3) If $u_2 < u$, the updated output u_2 is still less than the desired output value u, then return to the step 2.1. This means that the calculated output has not yet arrived at the desired output u, the input v has to increase continuously to calculate a larger output approaching to the desired output u.

2.4) When the calculated u_2 becomes larger than or equal to the desired output ($u_2 \geq u$), then interpolate the desired input as

$$v = v_1 + dv \frac{u - u_1}{u_2 - u_1}. \tag{2.59}$$

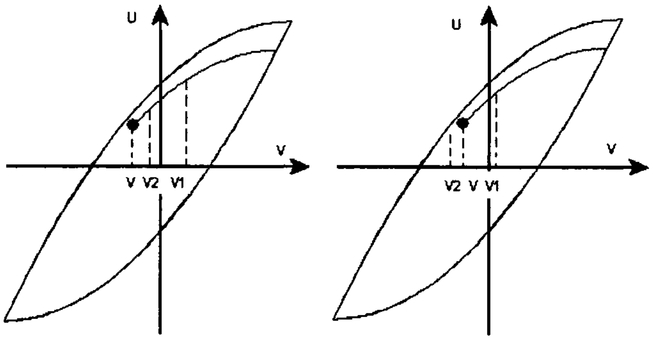


Fig. 2.27 Decreasing u

3) If the process has a decreasing trend, namely, $v \le v_{present}$ and $u \le u_{presen}$

(u decreases as v decreases, see Fig.2.27),

3.1) set $v_1 = v_2$ and $u_1 = u_2$

3.2) calculate and update $v_2 = v_1 - dv$ and $u_2 = H(v_2)$

3.3) if $u_2 > u$, go back to step3.1.

3.4) when the calculated u_2 becomes less than or equal to the desired output

$(u_2 \le u)$, then interpolate the desired input as

$$v = v_1 - dv\frac{u - u_1}{u_2 - u_1}. \tag{2.60}$$

It can be seen that if $v_1 \le v \le v_2$, $v_1 \le v^* \le v_2$ and $dv = | v_2 - v_1 |$, then $| v^* - v | \le dv$. In fact, the estimated v will generally be close to the exact inverse v^* much less than dv. In addition, as $dv \to 0$, then $v \to v^*$. However, to obtain finer outputs of u_1 and u_2 in (2.59) and (2.60) using interpolation equations (2.36) and (2.37), a large number l of dividing lines is required, which in turn increases the computational cost significantly in this inverse method. For example, a number $l = 98$ of dividing lines will create 5050 nodes on the Preisach plane T, resulting in a small dv which must be much less than the interval Δv, i.e.,

$$dv < \Delta v = \frac{v^+ - v^-}{l + 2} = 0.01(v^+ - v^-).$$

For any relative large variation of input $v_{desired} - v_{present}$, it takes many steps to update the input v to exceed the desired input $v_{desired}$, thus, it involves a large cost of computation to make the current point of state $(v_{present}, u_{present})$ to pass the desire point $(v_{desired}, u_{desired})$. Thus, some conclusions can be stated as follows:

The approximated inverse algorithm gives a good but not exact inverse hysteresis, but it is at least as close as dv to the true inverse. The algorithm is computationally expensive for extremely small dv and fine divisions of the Preisach plane T with a large l.

2.4.3 Open-Loop Compensation by Approximate Inverse Presiach Model

Based on the results presented in the above subsections, it is assumed that the Preisach model captures the hysteresis characteristics of an actuator accurately and comprehensively. Consequently, it is possible to pursue model-based compensation of hysteresis of the actuator using an open-loop compensation strategy depicted as in Fig.2.28.

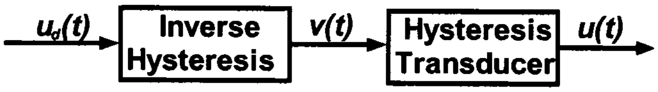


Fig. 2.28 Open loop compensation of hysteresis in SMA actuator

In the open-loop compensation configuration, the desired output from the actuator is the input signal applied to the inverse Preisach model as described by equations (2.1) and (2.59). The inverse hysteresis model predicts the input $v(t)$ of hysteresis that would achieve the desired output $u_d(t)$, and pass this calculated $v(t)$ to the actuator as its input. If the prediction is exact, the output $u(t)$ of the hysteresis will be equal to the desired output $u_d(t)$, and the inverse model will perfectly compensate the hysteresis in the actuator. The exact compensation by the inverse hysteresis is expressed as

$$u(t) = H[v(t)] = H[H^{-1}[u_d(t)]] = u_d(t) \tag{2.61}$$

Due to the inaccurate modeling of the hysteresis and the inverse model, as well as calculation errors, there is a difference between the predicted input to the actuator and the exact required input to the actuator. Therefore, the compensation is not perfect. The inaccurate compensation by the inverse hysteresis is expressed by

$$u(t) = H[v(t)] = H[\hat{H}^{-1}[u_d(t)]] \approx u_d(t) \tag{2.62}$$

This open loop compensation configuration with imperfect hysteresis cancellation can be applied when the plant output is within error tolerance specifications.

2.5 Examples for Identification, Implementation and Compensation of the Preisach Class of Hysteresis

2.5.1 A Two-Wire SMA Actuator

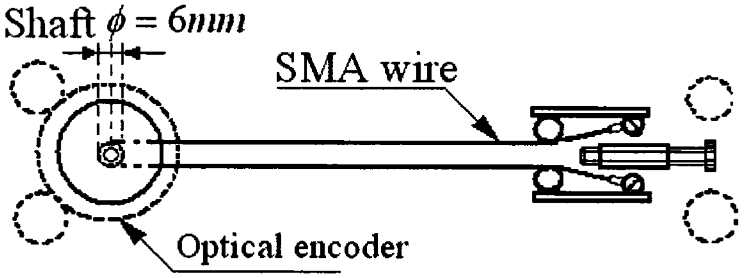


Fig.2.29 A two-wire SMA actuator [35]

In [35], a two-wire SMA actuator had been introduced, and it had been shown that the SMA actuator had the wiping-out property and the congruent property. Thus, it can be modeled through the Preisach hysteresis representation. For quoting the experimental data from [35] to conduct simulations of modeling and compensation of the hysteresis, the actuator is illustrated as Fig. 2.29, and its construct is explained as follows.

The mid-point of a length of SMA wire is anchored at the centre of the 6mm shaft by a set screw. This shaft is equipped with a 2000 count/revolution optical encoder, giving an angular of 0.18 degrees per scale, or less than 10 μm of change in length of the SMA wire. A total of 35cm of ϕ 0.3mm NiTi wire is used to provide two 15cm lengths of active wire. The wires have a common electrical connection at the shaft, and the two "free" ends have electrical lugs crimped over a knot tied in the wire. The lugs provide an electrical connection to the ends of the SMA wire, and also allow the wire ends to be secured to a terminal block. This terminal block is mobile, and can be positioned by means of a screw. In this configuration, one wire shrinks caused by applied current on it, while its opposing wire stretches. It is considered that a reverse current has been applied on the actuator when the opposing wire is heated. Even though the rotation of the actuator is caused by the current applied on one of its wires, the temperature of the SMA wires above ambient is considered as the input of the actuator while the rotation angular degree of the shaft is regarded as the output of the actuator.

2.5.2 Identification of Density Function

Consider the SMA actuator with the following input (temperature ^{o}C) restriction

$$185 \geq \alpha \geq \beta \geq -185$$

inside which the actuator has hysteresis property and beyond which the hysteresis becomes saturated. This domain is equivalent to a triangular Preisach plane defined by

$$T = \{(\alpha,\beta) \mid 185 \geq \alpha \geq \beta \geq -185\}.$$

To construct a FOD surface over the domain, the Preisach plane T is first uniformly divided by $l = 9$ horizontal lines and $l = 9$ vertical lines, respectively. In this way, the plane T is divided into small cells with $N = (l+2)(l+3)/2 = 66$ nodes. For each node, the corresponding pair of coordinates is expressed as (β_j, α_i), where α_i and β_j are the *ith* and *jth* element of the two respective vectors, with a restriction that $i \geq j$

$$\alpha = [-185 \; -148 \; -111 \; -74 \; -37 \quad 0 \quad 37 \quad 74 \quad 111 \quad 148 \quad 185]$$

$$\beta = [-185 \; -148 \; -111 \; -74 \; -37 \quad 0 \quad 37 \quad 74 \quad 111 \quad 148 \quad 185]$$

The measured outputs $u_{\alpha\beta}$ (rotation degree) of the hysteretic actuator for the above input pairs (β_j, α_i) are arranged as shown in the following upward triangular matrix:

$$u_{\alpha\beta} = \begin{bmatrix}
-17.000 & -16.370 & -14.400 & -9.160 & 0.290 & 9.500 & 14.400 & 16.200 & 16.780 & 16.950 & 17.000 \\
-17.000 & -16.380 & -14.450 & -9.320 & -0.050 & 8.980 & 13.780 & 15.550 & 16.110 & 16.280 & NaN \\
-17.000 & -16.410 & -14.580 & -9.690 & -0.880 & 7.700 & 12.270 & 13.950 & 14.480 & NaN & NaN \\
-17.000 & -16.480 & -14.850 & -10.530 & -2.720 & 4.890 & 8.940 & 10.430 & NaN & NaN & NaN \\
-17.000 & -16.600 & -15.350 & -12.010 & -5.990 & -0.130 & 2.980 & NaN & NaN & NaN & NaN \\
-17.000 & -16.750 & -15.970 & -13.880 & -10.120 & -6.460 & NaN & NaN & NaN & NaN & NaN \\
-17.000 & -16.870 & -16.480 & -15.430 & -13.530 & NaN & NaN & NaN & NaN & NaN & NaN \\
-17.000 & -16.950 & -16.770 & -16.320 & NaN & NaN & NaN & NaN & NaN & NaN & NaN \\
-17.000 & -16.980 & -16.910 & NaN & NaN & NaN & NaN & NaN & NaN & NaN & NaN \\
-17.000 & -16.990 & NaN & NaN & NaN & NaN & NaN & NaN & NaN & NaN & NaN \\
-17.000 & NaN & NaN & NaN & NaN & NaN & NaN & NaN & NaN & NaN & NaN
\end{bmatrix}$$

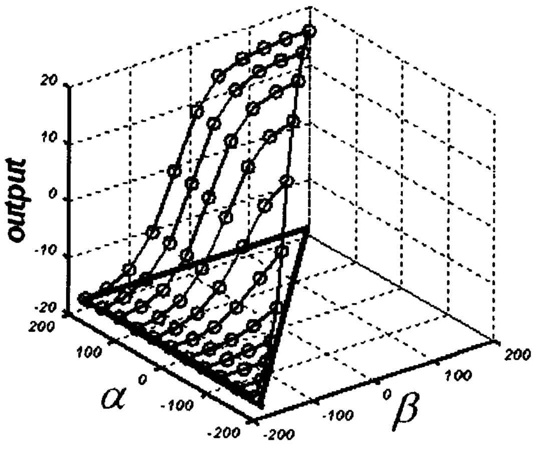


Fig. 2.30 FOD curves on the FOD surface over the Preisach plane T

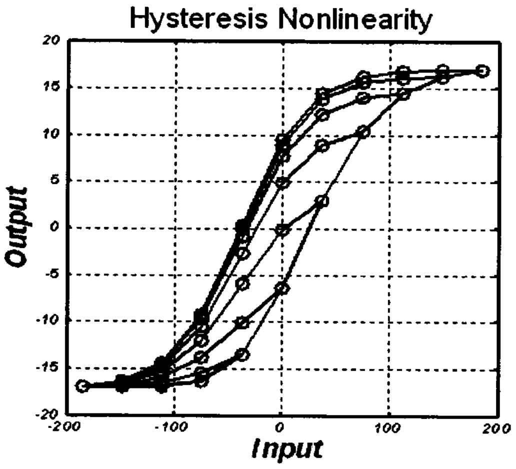


Fig. 2.31 FOD curves in the hysteresis major loop

Using the above matrix $u_{\alpha\beta}$, the FOD surface is constructed, as shown in Fig.2.30. If the FOD surface is projected along the α axis (i.e., onto the output-β plane), then, the region where the hysteresis loops occur can be obtained (see Fig.2.31). The outline of the region is the major hysteresis.

For this FOD surface, from reference [35], the assumed candidate FOD function is given by equation (2.63), where 5 parameters are to be identified.

$$\hat{u}_{\alpha\beta} = x_1 \times \frac{(e^{x_4(v^- + x_5)} - e^{x_4(\beta + x_5)})}{(1+e^{x_2(\alpha+x_3)})(1+e^{x_4(\beta+x_5)})(1+e^{x_4(v^-+x_5)})} + u^- \qquad (2.63)$$

When $\alpha = \beta = v^+$, then $\hat{u}_{\alpha\beta} = \hat{u}_{v^+v^+} = u^+$. Thus, the parameter x_1 can be calculated as

$$x_1 = (u^+ - u^-) \times \frac{(1+e^{x_2(v^+ + x_3)})(1+e^{x_4(v^+ + x_5)})(1+e^{x_4(v^- + x_5)})}{(e^{x_4(v^- + x_5)} - e^{x_4(v^+ + x_5)})} \qquad (2.64)$$

Equation (2.64) shows that the FOD surface function only has 4 independent parameters as x_2, x_3, x_4, and x_5, which can be identified by the least squares method.

Given the following conditions as output of the negative saturation state $u^- = -17$, input of the negative saturation state $v^- = -185$, and input of the positive saturation state $v^+ = 185$, the parameters $[x_2 \;\; x_3 \;\; x_4 \;\; x_5]$ of the surface can be optimized by the Nelder-Mead simplex algorithm [42], as

$$[x_2 \;\; x_3 \;\; x_4 \;\; x_5] = [-0.026 \;\; -16.773 \;\; -0.033 \;\; 38.473]$$

Furthermore, the parameter x_1 can be given by (2.64) as

$$x_1 = 34.724.$$

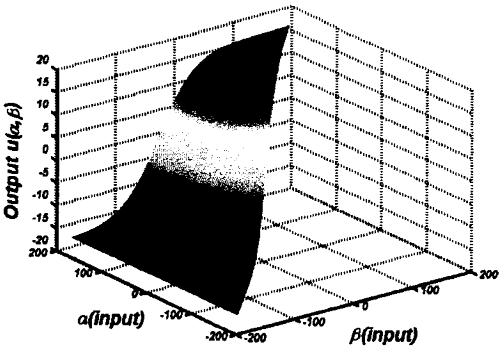


Fig. 2.32 FOD surface

After conducting the parameter identification, the estimated FOD surface function $\hat{u}_{\alpha\beta}$ can be considered as the exact FOD surface function $u_{\alpha\beta}$. The FOD surface function (2.55) describes a smooth FOD surface which is depicted as in Fig.2.32.

Recalling equation (2.29), $\mu(\alpha,\beta)=\frac{1}{2}\frac{\partial^2 u_{\alpha\beta}}{\partial\alpha\partial\beta}$

where $u_{\alpha\beta}$ indicates the FOD surface function (2.63), the corresponding density function can be calculated as [35]

$$\mu_{\alpha\beta}=(x_1-1)x_2x_4\frac{e^{x_4(\beta+x_5)}e^{x_2(\alpha+x_3)}e^{x_4(\beta+x_5)}}{(1+e^{x_2(\alpha+x_3)})^2(1+e^{x_4(\beta+x_5)})^2(1+e^{x_4(v^-+x_5)})}+\frac{1}{2}x_1e^{x_4(v^-+x_5)} \qquad (2.65)$$

By substituting the values of the parameters x_1, x_2, x_3, x_4, x_5 and $v^-=-185$, the density function (2.65) is solved and is shown in Fig.2.33.

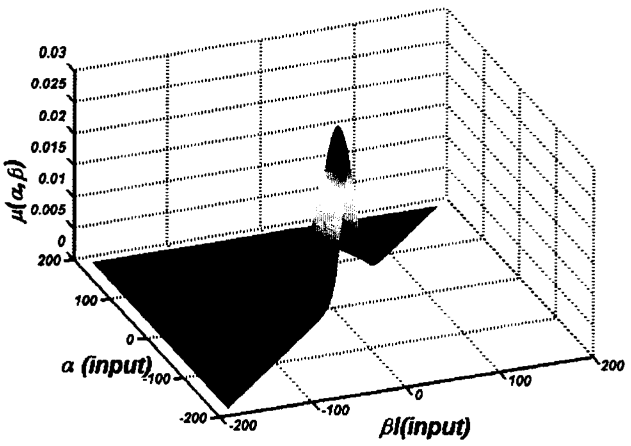


Fig. 2.33 Density distribution $\mu_{\alpha\beta}$

2.5.3 Examples for Numerical Implementation of the Preisach Model

With the FOD surface function $u_{\alpha\beta}$ (2.63) and its corresponding density function $\mu_{\alpha\beta}$ (2.65), it is possible to conduct some calculation to find the memory interface line in the Preisach plane *T*; to obtain the output of hysteresis at any moment; and to draw the hysteresis loops for the above hysteretic actuator subject to some continuous input with different local extrema. The related *m. files* in *Matlab* are written and listed in Appendix 1.

To demonstrate the application of the *m.files,* some special cases with different input histories are studied as following:

Example 1

For the hysteresis of the actuator described by the density function (2.65) and conditions $u^+ = 17, u^- = -17, v^- = -185$ and $v^+ = 185$, it is subjected to a continuous input with a sequence of local input extrema as

$$v_{LE}=[-185\ 185 -185\ 145\ -145\ 145\ -145\ 105\ -105\ 75\ -75\ 45\ -45\ 20\ -30].$$

Between any two adjacent local extremes, it is assumed that there is a linear increase or decrease process joining two input extremes (see Fig.2.34). It should be mentioned that the form of these continuous processes between the adjacent input extremes has no affect on the memory interface and at the output of the hysteresis at the end moment of the process. The process is needed for the calculation of the output values of the hysteresis at any time during the process and to draw the hysteresis loops. The continuous input $v(t)$ lasts for a period of $t = 100$ seconds.

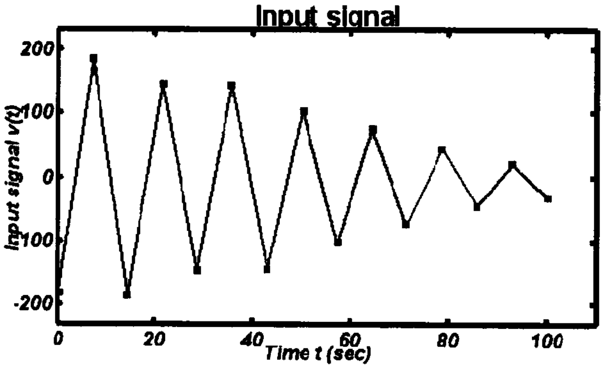
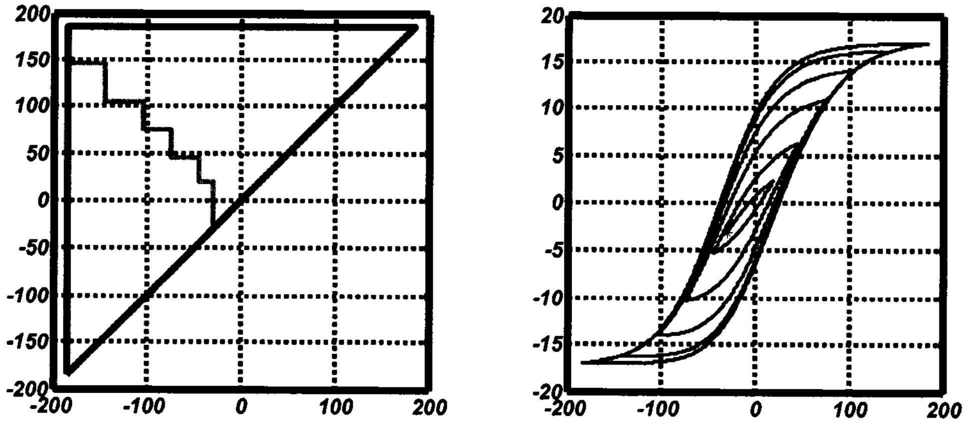


Fig. 2.34 Input signal $v(t)$ with a sequence of local extreme input values

Conveying the above continuous input process to the *m.file* in Appendix 1, the output values at any moment of time along the process can be calculated. In this example, the calculated final output is specifically given as $u(-30) = -3.0468$ when the input $v = -30$. The memory interface (see Fig.2.35a) at the final moment of time is found and drawn in the Preisach plane. The hysteresis loops (see Fig.2.35b) for the entire process is drawn by finding all outputs for every sampling moment with sampling time $t_s = 0.1\sec$. The final point is the last point at the hysteresis loops.



a Memory interface

b Hysteresis loops

Fig. 2.35 Memory interface lines and hysteresis loops for a sequence of local input extrema $v_{LE}=[-185,185,-185,145,-145,145,-145,105,-105,75,-75,45,-45,20,-30]$

From Fig.2.35a, it can be realized that the green staircase is the memory interface $L(t)$ of this special input sequence. Note that the sub-processes $v=[-185 \quad 185 \quad -185]$ and $v=[145 \quad -145 \quad 145 \quad -145]$ does not have interface lines shown on the staircase $L(t)$, because its interface lines have been wiped out. Thus the entire process is equivalent to

$$v_{LE}=[-185 \quad -145 \quad 105 \quad -105 \quad 75 \quad -75 \quad 45 \quad -45 \quad 20 \quad -30\ ].$$

However, the periodical sub-process $v=[-185 \quad 185 \quad -185]$ causes the major loop, and the periodic sub-process $v=[145 \quad -145 \quad 145 \quad -145]$ results in a completed minor loop inside the major loop.

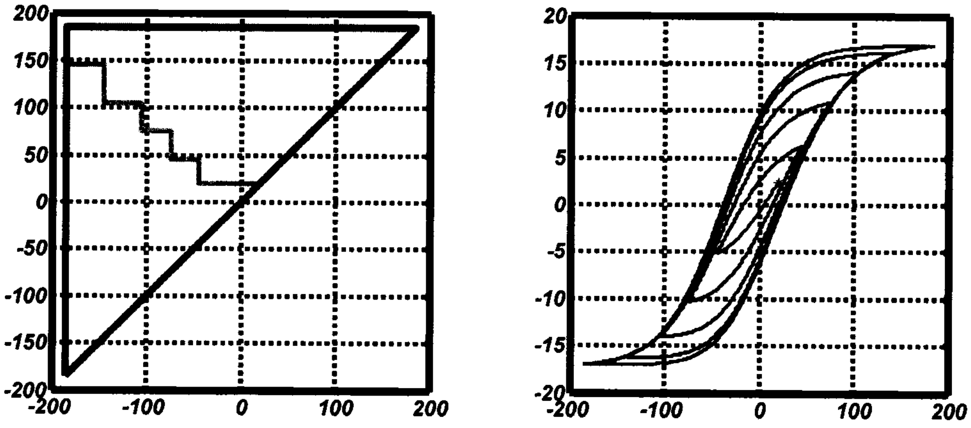
Example 2

For the same hysteresis described as above, suppose that it is subjected to a continuous input with a sequence of local input extrema as

$$v_{LE}=[-185 \ 185 \ -185 \ 145 \ -145 \ 145 \ -145 \ 105 \ -105 \ 75 \ -75 \ 45 \ -45 \ 20 \quad].$$

Also, there is a linear increase or decrease process joining every pair of adjacent input extrema. The simulation time and the sampling time are the same as the example 1.

Performing the simulation through the *m. files* in Appendix 1 obtains the memory interface lines as shown in Fig.2.36a and the hysteresis loops as in Fig.2.36b. The hysteresis loops is created by the last input increase from -50 to 20. And the last point is the final input-output state of $(v,u)=(20,2.4481)$.



a Memory interface

b Hysteresis loops

Fig.2.36 Memory interface lines and hysteresis loops for a sequence of local input extrema $v_{LE}=[-185,185,-185,145,-145,145,-145,105,-105,75,-75,45,-45,20]$

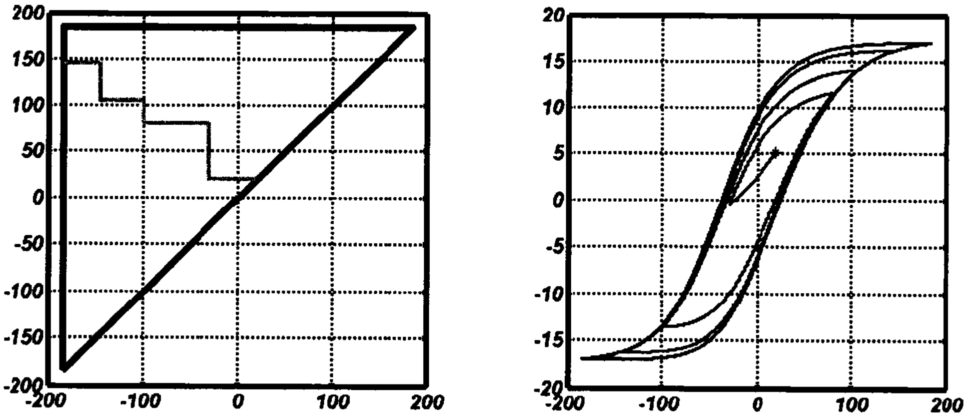
Example 3

Similarly, when the actuator is subjected to a continuous input with a sequence of local extreme input values as

$$v_{LE}=[-185\ \ 185\ \ -185\ 145\ \ -145\ \ 105\ \ -100\ \ 75\ \ -45\ \ 80\ \ -30\ \ 20]$$

then, this input process has the same starting state and final state as in case2, but has a different history from case 3.

Simulation gives the memory interface line for the final moment of time as shown in Fig.2.37a, and the hysteresis loops are drawn as in Fig.2.37b. In Fig.2.37b, the hysteresis loops is created by the last input increase from -30 to 20, and the marked point has an output value as $u(20) = 5.1428$ when $v = 20$. The figure clearly shows that the memory interface and the output at the final moment of time, as well as the hysteresis loops, are different from case2. This is because the two cases have different histories.



a Memory interface

b Hysteresis loops

Fig. 2.37 Memory interface lines and hysteresis loops for a sequence of local input extrema v_{LE} =[-185,185,-185,145,-145,105,-100,75,-45,80,-30,20]

Remark: from the above three simulation cases, the following facts are summarized: each line segment of memory interface lines corresponds to a branching curve inside the major loop. As line segments develop upwards, an increasing branching curve is formed; while line segments move leftwards, a decreasing branching curve is generated. When two adjacent line segments of a hysteresis process are wiped out by later interface lines, a minor loop consisting of an increasing branching curve and a decreasing branch curve is created inside the major loop. Even though two memory interface lines formed through different histories have the same shape, they create different shapes of hysteresis loops with the same outputs at the final moment of time.

2.5.4 An Example for Open-loop Compensation with Inverse Hysteresis

The following example shows how the inverse Preisach model is determined. The objective is to have the hysteresis $u(t) = H[v](t)$ to produce a continuous output as

$$u_d = -8.5(\cos(\frac{\pi}{30}\mathrm{t}) + \cos(\frac{\pi}{30\sqrt{2}}\mathrm{t})) \,, \tag{2.66}$$

The hysteresis input $v(t)$ is unknown, and its value as a function of time is required to obtain the desired output. Thus, the inverse Preisach model must be used to calculate the input $v(t)$. According to equation (2.58) and the inverse algorithm introduced in section 2.4, the Preisach model and its associated approximated inverse model have been software coded and is given in Appendix 2. These two models are cascaded as the open loop compensation configuration of Fig.2.28, where the desired output is treated as an input signal to the inverse hysteresis model. This open-loop compensation configuration can be software executed in a real-time environment for simulation purposes, or in real-time applications to cancel or reduce the hysteresis nonlinearity.

The initial and final simulation time is set as $t = 0:269\sec$, with a sampling time of $t_s = 1\sec$. For this input signal to the inverse model, the minimum of the desired output signal is $\min(u_d) = -17$ when $t = 0$, which means that the actuator starts from its negative saturation state. The maximum of the desired output signal is $\max(u_d) = 14.6635 < u^+ = 17$, which means that the hysteresis does not need to reach its positive saturation state. The simulation results are shown as Fig.2.37. In Fig.2.37, the signal $v(t)$ is calculated by the inverse hysteresis model, it is applied on the actuator as an input signal; the signal $u(t)$ is the real output of the actuator; the input signal $v(t)$ and the real output signal $u(t)$ have the similar tendency of variation; the relationship between the desired output $u_d(t)$ and the real output $u(t)$ of the actuator is the compensation effect; the relationship between the input signal $v(t)$ and output signal $u(t)$ of the actuator is the hysteresis nonlinearity.

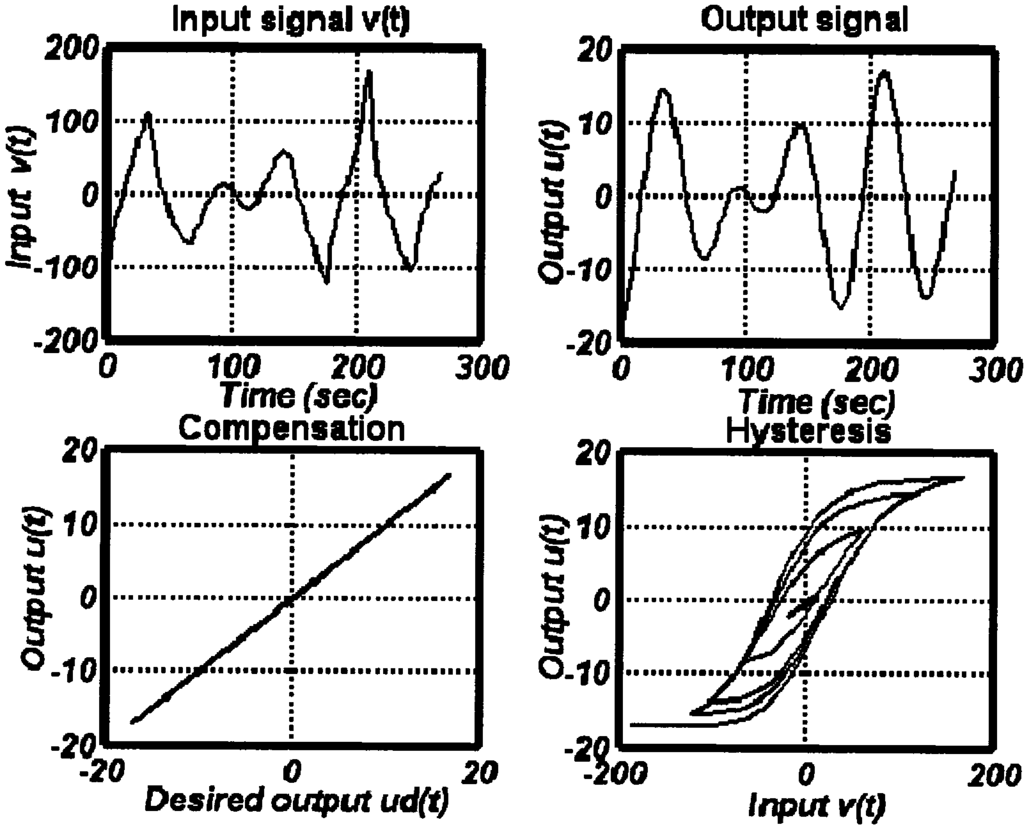


Fig. 2.38 Modeling and compensation of a hysteresis generated by a continuous signal

Remark: from the above simulation results, it is observed that if the hysteresis is modeled with high precision, the inverse hysteresis model can nearly reduce most of the nonlinear effect. However, due to the inaccurate modeling of the hysteresis and hysteretic dynamic properties, the hysteresis model cannot precisely describe the hysteresis nonlinearity. Furthermore, the inverse hysteresis model cannot completely cancel the hysteresis effect. Thus, the open loop compensation method is not enough, and an on-line identification or adaptive parameter identification methodology is required. These will be introduced in forthcoming chapters.

2.6 Summary

In this chapter, the classical Preisach model and its properties have been described and mathematically proven. The implementation method to model the Preisach class of hysteresis by identifying its FOD surface function or density function, and finding the memory interface has been developed. An associated approximated inverse hysteresis model has been derived. Some calculations have been performed to identify the density function, to find the memory interface, to obtain the output of hysteresis, and to verify the properties of the Preisach model. A simulation of open loop hysteresis compensation has been carried out to illustrate the modeling process and reduction effect to the hysteresis.

In this applicable method, the memory interface $L(t)$ is very important because the off-line parameter identification depends on it, the calculation of the hysteresis output depends on it, and the inverse hysteresis operator is based on it. This method is mostly adopted in existing literatures. This method, in some way, simplifies the calculation to obtain output of hysteresis because it divides the entire Preisach plane into two domains with uniform outputs of relays. It is acceptable for off-line identification and simple on-line compensation.

However, for an input processes with an irregular variation in the real time environment, there is extremely large amount of calculation involved to find memory interface line, and to numerically implement the hysteresis model as well as to compensate the hysteresis using the inverse hysteresis operator.

To find memory interface line, the local input extrema are required to be found first, and then according to the wiping-out property of hysteresis, these extrema are sorted out at every moment of time during the process. The more irregular variation of an input processes is applied on the hysteresis, the shorter sampling time is required to be adopted, and then the finer cells are created by dividing the Preisach plane, finally the more calculation is involved to seek the memory interface line.

Also the more irregular variation of an input process is applied on the hysteresis, the more vertices on the memory interface line are formed, and the more narrow strips on the $S^+(t)$ domain are sliced into by these vertices, finally, the more calculation is involved in the numerical implementation. This can be understood from the Fig.2.22 and the equation (2.41).

Since the inverse hysteresis model uses (2.41) several times during every sampling period, the cost of calculation becomes extremely large for the compensation methods based on seeking memory interface line.

On-line adaptive control configurations have more applications in industries. These configurations require more effective methods to capture the dynamics of systems. Therefore, in the next chapter, a direct method will be introduced to implement the Preisach class of hysteresis.

CHAPTER 3

IMPLEMENTATION OF LINEARLY PARAMETERIZED PREISACH HYSTERESIS MODEL WITHOUT USING MEMORY INTERFACE LINES

3.1 A Direct Method to Implement the Preisach Model

In chapter 2, a numerical method procedure was introduced to implement the Preisach model using memory interface lines in the Preisach plane. Due to the extremely large amount of calculations required to sort the dominating input extrema of hystersis at every moment of the entire process, it is very difficult to divide the Preisach plane into a large amount of strips, especially when the staircase shape memory interface line has many vertexes. Also, it is usually difficult to formulate a suitable weighting function $\mu(p_1,p_2)$ which is expressed in double integral form over the Preisach plane T. Thus, it is difficult to implement real-time control of systems with hysteresis using the numerical method.

To avoid the complexity of the numerical method, in this section a direct approach will be discussed. For this method, at every moment of the entire hysteresis process, the output state of every relay will be calculated according to the definition of the relay; then the output of each relay will be weighted with its density factor; and finally all weighted outputs of relays will be summed together to calculate the output of the hysteresis at any moment of time.

3.1.1 Linearly Parameterized Preisach Model

Although the Preisach hysteresis model can be described in integral form by equation (2.1), it is difficult to directly calculate the output of hysteresis, as explained by the above two reasons. Thus, to circumvent these difficulties, the integral form of the Preisach hysteresis model will be transformed into an expression in linearly parameterized style.

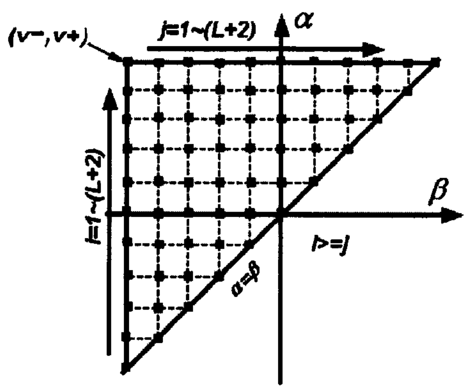


Fig.3.1 Dividing Preisach plane T into small cells

To linearly parameterize the Preisach hysteresis model, first uniformly divide the Preisach plane T with l horizontal lines and l vertical lines into $N=(l+2)(l+3)/2$ small cells (see Fig.3.1). The coordinates (β_j,α_i) of the lower-left nodes of cells can be expressed by

$$\begin{cases} \beta_j=(j-1)\dfrac{v^+-v^-}{l+1}+v^- \\ \\ \alpha_i=(i-1)\dfrac{v^+-v^-}{l+1}+v^- \end{cases} \quad with \quad i,j=1,2,\ldots,(l+2);\quad i\geq j \tag{3.1}$$

After the dividing of the Preisach plane T, the total contribution of relays in each cell to the Preisach hysteresis model (2.1) can be lumped together as the effect of a relay associated with the lower-left node of the cell. In this way, one has

$$\rho_{\alpha_i\beta_j}\hat{\gamma}_{\alpha_i\beta_j}[v(t)]=\iint_{ij^{th}cell}\mu(\alpha,\beta)\hat{\gamma}_{\alpha\beta}[v(t)]d\alpha d\beta \tag{3.2}$$

where $\rho_{\alpha_i\beta_j}$ is called the lumped density of the $ij th$ cell to its lower-left node (β_j,α_i). As l is selected large enough, the dividing of the Preisach plane T becomes very fine, the cells become very small, and the output values $\hat{\gamma}_{\alpha\beta}[v(t)]$ of all relays on the $ij th$ cell can be approximately considered as an identical value as the output value $\hat{\gamma}_{\alpha_i\beta_j}[v(t)]$ of the lower-left node of the $ij\,th$ cell. Thus (3.2) is rewritten as

$$\rho_{\alpha_i\beta_j}\hat{\gamma}_{\alpha_i\beta_j}[v(t)] = \hat{\gamma}_{\alpha_i\beta_j}[v(t)]\iint_{ij^{th}cell}\mu(\alpha,\beta)d\alpha d\beta \tag{3.3}$$

and therefore,
$$\rho_{\alpha_i\beta_j} = \iint_{ij^{th}cell}\mu(\alpha,\beta)d\alpha d\beta \tag{3.4}$$

Define
$$c \overset{\Delta}{=} \iint_T \mu(\alpha,\beta)d\alpha d\beta = \sum_i^{l+2}\sum_{j=1}^{i}\iint_{ij^{\ th}cell}\mu(\alpha,\beta)d\alpha d\beta \tag{3.5}$$

and
$$\mu_{n\alpha_i\beta_j} \overset{\Delta}{=} \mu_{\alpha_i\beta_j} /(\sum_i^{l+2}\sum_{j=1}^{i}\mu_{\alpha_i\beta_j}) \tag{3.6}$$

where c is the total density value of the Preisach hysteresis and $\mu_{n\alpha_i\beta_j}$ are the normalized densities of all lower-left nodes on the Preisach plane T, so that

$$\sum_i^{l+2}\sum_{j=1}^{i}\mu_{n\alpha_i\beta_j} = 1\,. \tag{3.7}$$

Furthermore
$$\sum_i^{l+2}\sum_{j=1}^{i}c\mu_{n\alpha_i\beta_j} = c\sum_i^{l+2}\sum_{j=1}^{i}\mu_{n\alpha_i\beta_j} = c = \sum_i^{l+2}\sum_{j=1}^{i}\iint_{ij^{\ th}cell}\mu(\alpha,\beta)d\alpha d\beta \tag{3.8}$$

Considering (3.4) and (3.8), one has

$$\rho_{\alpha_i\beta_j} = c\mu_{n\alpha_i\beta_j} \tag{3.9}$$

Indeed, as the number of dividing lines $l \to \infty$, each cell degenerates as its lower-left node. Thus, one has $\rho_{\alpha_i\beta_j} = c\mu_{n\alpha_i\beta_j} \to \mu_{\alpha_i\beta_j}$

For an acceptable modeling precision, a large enough l results in equation (3.3), and then the Preisach hysteresis model (2.1) can be linearly parameterized as

$$u(t) = \sum_{i=1}^{l+2}\sum_{j=1}^{i}\iint_{ij^{th}cell}\mu(\alpha,\beta)\hat{\gamma}_{\alpha_i\beta_j}[v(t)]d\alpha d\beta$$

$$= \sum_{i=1}^{l+2}\sum_{j=1}^{i}\hat{\gamma}_{\alpha_i\beta_j}[v(t)]\iint_{ij^{th}cell}\mu(\alpha,\beta)d\alpha d\beta$$

$$= \sum_{i=1}^{l+2} \sum_{j=1}^{i} \rho_{\alpha_i \beta_j} \hat{\gamma}_{\alpha_i \beta_j}[v(t)]$$

$$= \Theta^T F(v) \tag{3.10}$$

where $$F(v)^T = [\hat{\gamma}_{\alpha_1\beta_1}, \hat{\gamma}_{\alpha_2\beta_1}, \hat{\gamma}_{\alpha_2\beta_2}, \hat{\gamma}_{\alpha_3\beta_1}, \cdots \hat{\gamma}_{\alpha_{l+2}\beta_{l+2}}] = [\hat{\gamma}_1, \hat{\gamma}_2, \cdots \hat{\gamma}_i, \cdots \hat{\gamma}_N] \tag{3.11}$$

and $$\Theta^T = [\rho_{\alpha_1\beta_1}, \rho_{\alpha_2\beta_1}, \rho_{\alpha_2\beta_2}, \rho_{\alpha_3\beta_1}, \ldots, \rho_{\alpha_{l+2}\beta_{l+2}}] \overset{\Delta}{=} [\theta_1, \theta_2, \ldots, \theta_i, \ldots, \theta_N] \tag{3.12}$$

are N dimension vectors, respectively, and $\rho_{\alpha_i \beta_j}$ are calculated by (3.5), (3.6) and (3.9).

3.1.2 Simulation Principle

This direct method is to implement the linearly parameterized Preisach hysteresis model

$$u(t) = H[v(t)] = \Theta^T F(v) \tag{3.10}$$

The finer the Preisach plane T is divided, the larger the vectors Θ and $F(v)$ are obtained, and the more precisely the linearly parameterized Preisach model approaches the original Preisach hysteresis model.

To calculate the output of hysteresis using (3.10), the lumped densities $\rho_{\alpha_i \beta_j}$ in the parameter vector Θ are required to be known. The lumped densities $\rho_{\alpha_i \beta_j}$ are related to the density function or distribution $\mu_{\alpha\beta}$ of the hysteresis by equations (3.5), (3.6) and (3.9). Thus, it is still required to identify the densities $\mu_{\alpha\beta}$ of the hysteresis in advance. It can be carried out by the least squares method to fit an assumed density function to all measured output data on the FOD surface, or by the interpolation method. These methods have been systemically presented in chapter 2.

Output values $\hat{\gamma}_{\alpha_i \beta_j}$ of relays, which can only be two alternative values as +1 and -1, are the elements of the vector $F(v)$. Theoretically, they can be determined by the definition of the relay. To model and simulate systems with the Preisach class of hysteresis, one has to construct the elementary operator (relay) of the Preisach model according to its definition. Fortunately, there is an elementary nonlinear operator (relay) in the library of the simulation software Simulink for real time simulation or real-time control. Thus, it is not necessary to write an s-function to describe the relay of the Preisach model. Instead, the existing relay in the Simulink package can be conveniently used to express the Preisach relays, and to establish a model for real-time control of systems with hysteresis.

To express the "relays" in the Preisach model with the simulation "relays" in Simulink, the relay defined in the Preisach model and the relay in the library of the Simulink package are examined first as below.

The "relay" defined in the Preisach model has two input thresholds, α and β, with a restriction that $\alpha \geq \beta$, and has two saturation states with output values as -1 and $+1$. This means that the output of the relay will switch to $+1$ as the input increases above the upper threshold α, and the output of the relay will switch back to -1 when the input decreases below the lower threshold β (see Fig.2.1). There is no restraint on the symmetry of α and β about the output axis $\hat{\gamma}_{\alpha\beta}$. On the other hand, the "relay" defined in the Simulink package is similar to the "relay" defined in the Preisach model. It also has two asymmetric input thresholds α and β satisfying the condition $\alpha \geq \beta$. But, unlike the two fixed output states "+1" and "-1" defined in the Preisach model, the Simulink "relay" has two user defined parameters as "***output when on***" and "***output when off***' (see Fig.3.2).

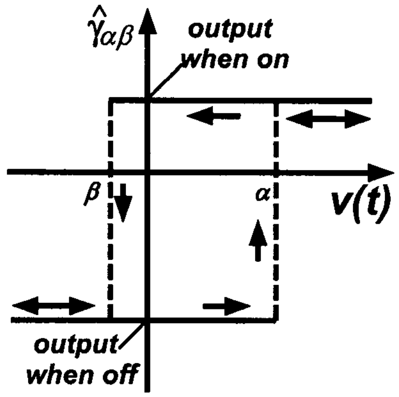


Fig.3.2 A relay in the Simulink package

3.1.3 Simulation of a Relay

To define the parameters of the Preisach "relay" with respect to the Simulink package, the parameters α and β are set as $\alpha = 2$ and $\beta = -2$ in the pop-up menu of "*Block Parameters: Relay*", and the output states "***the output when on***" and "***the output when off***' are assigned values as "+1" and "-1", respectively. The input signal applied to the simulation block "relay" is chosen as $v(t) = -4\cos(t)$. This completes the modeling of the Preisach "relay" in the Simulink environment. Running the simple simulation model, the output-input relation of the block "relay" can be obtained. The relation is identical to the elementary operator defined in the Preisach model with two thresholds as $\alpha = 2$ and $\beta = -2$. The simulation model and the results are shown as in Fig.3.3 and Fig.3.4, respectively.

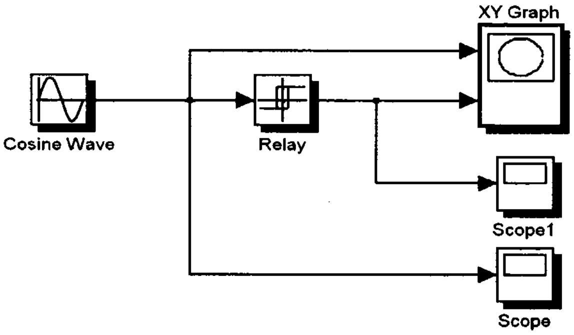


Fig.3.3 Simulation model of the Peisach "relay"

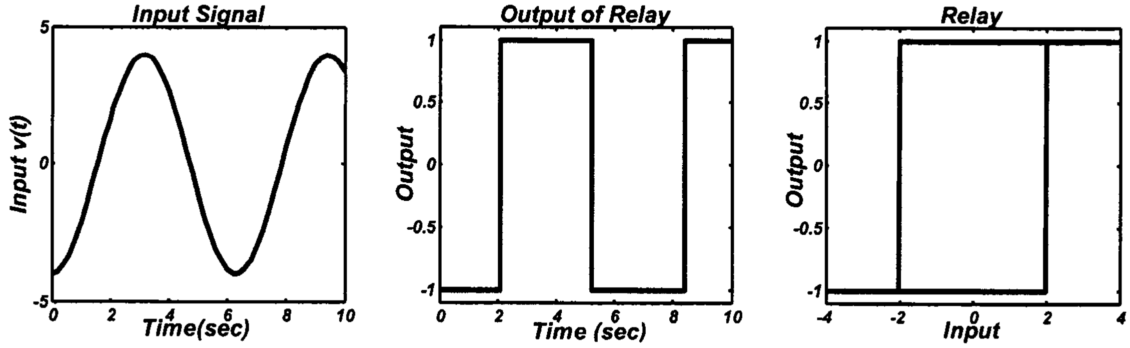


Fig. 3.4 Results of the simulation of the Preisach "relay"

3.1.4 Simulation of a Rough Hystersis

After modeling a relay in Simulink, a rough hysteresis can be modeled as below by superimposing a few weighted relays.

Assume there are three points, $p_1(-3,-2)$, $p_2(-3.5,3)$ and $p_3(1.5,2)$, on the Preisach plane T to represent three relays, (β_1,α_1), (β_2,α_2) and (β_3,α_3) (see Fig.3.5). The hysteresis input range is assumed as $v(t)\in[v^-,v^+]=[-4,4]$. Also densities of the three relays are assumed as $\mu_1=0.3$, $\mu_2=0.45$, $\mu_3=0.25$, respectively. The parameters $\alpha_1,\alpha_2,\alpha_3$; β_1,β_2,β_3; and the densities μ_1,μ_2,μ_3 of the three relays are expressed in three vectors as

$$\alpha=[\alpha_1,\alpha_2,\alpha_3]=[-2,\quad 3,\quad 2]$$

$$\beta=[\beta_1,\beta_2,\beta_3]=[-3,\quad -3.5,\quad 1.5]$$

and

$$\mu=[\mu_1,\mu_2,\mu_3]=[0.3,\quad 0.45,\quad 0.25].$$

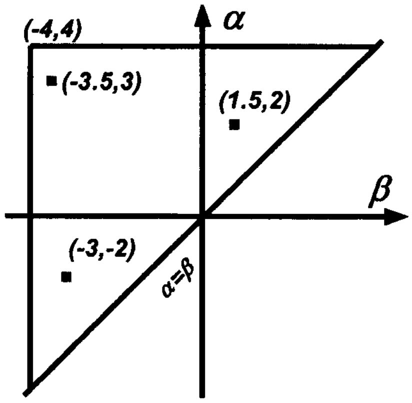


Fig.3.5 Relays on the Preisach plane T

The three vectors α, β, μ are entered into the workspace of MatLab in advance, and the relay parameters in the Simulink package refer to the values of the three variables α, β, μ by calling their values in the MatLab workspace. The parameters "***output when on***" and "***output when off***" of these relays are assigned values as "+1" and "-1", respectively. In this way, the "relay" in the Simulink package becomes a vector consisting of three relays, and the output is also a vector. As this output vector is made as an inner product with the density vector Θ, which is obtained by equations (3.5), (3.6), (3.9) and (3.12), a scalar output is obtained as the output of the hysteresis at any instant of time. When the same continuous input signal $v_1(t) = -4\cos(t)$ is applied to the vector relay, a rough hysteresis is formed. The simulation model and the corresponding results are shown as in Fig.3.6 and Fig.3.7.

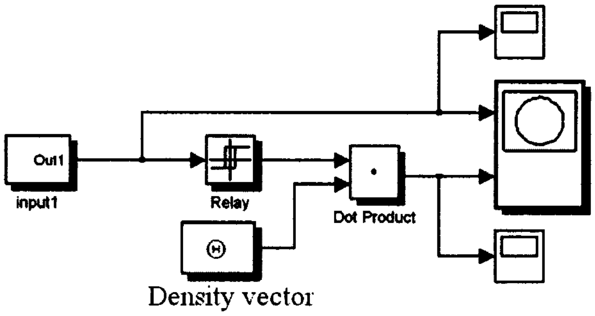


Fig.3.6 Simulation model with vector relay

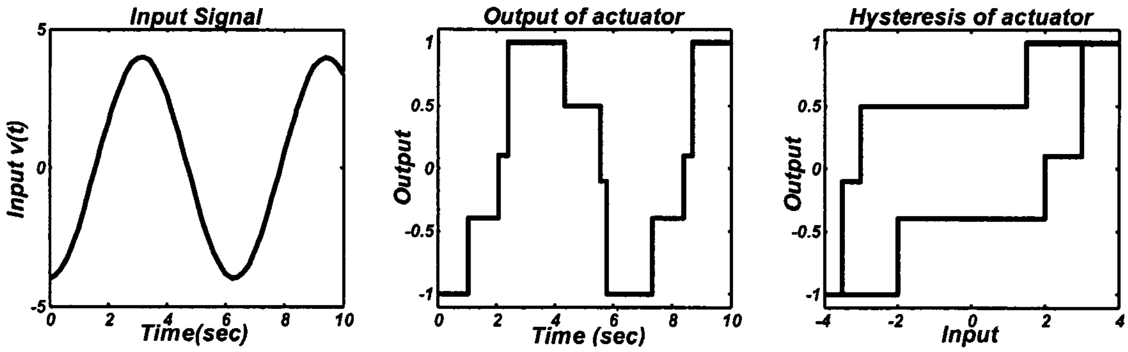


Fig.3.7 Simulation results for a rough hysteresis

3.1.5 Simulations of Smooth Hystersis Loops

In the above simulation model, if the dividing of the Preisach plane T becomes finer to generate smaller cells, and a continuous density function is divided in the same manner to create a vector Θ consisting of lumped densities of all these small cells, a smoother hysteresis can be produced. For example, if $l = 100$ horizontal lines and $l = 100$ vertical lines are used to uniformly discretize the Preisach plane T into $N = (l+2)(l+3)/2 = 5253$ small cells, N relays corresponding to N left-lower nodes of N small cells are obtained. In this case, the density function is assumed as $\mu(\alpha, \beta) = 1/5253$, which defines a uniform density distribution over the Preisach plane T. For an input signal $v(t) = -6e^{-0.055t}\cos(t)$, the simulation results of the model in Fig.3.6 are shown as in Fig.3.8. Due to the exponential decay of the amplitude of the input signal $v(t)$, the output of the hysteresis cannot always reach its negative and positive saturation states. When the local input extrema cannot exceed the hysteresis input region $[v^-, v^+] = [-4, +4]$, the minor loops of hysteresis are formed inside the major loop.

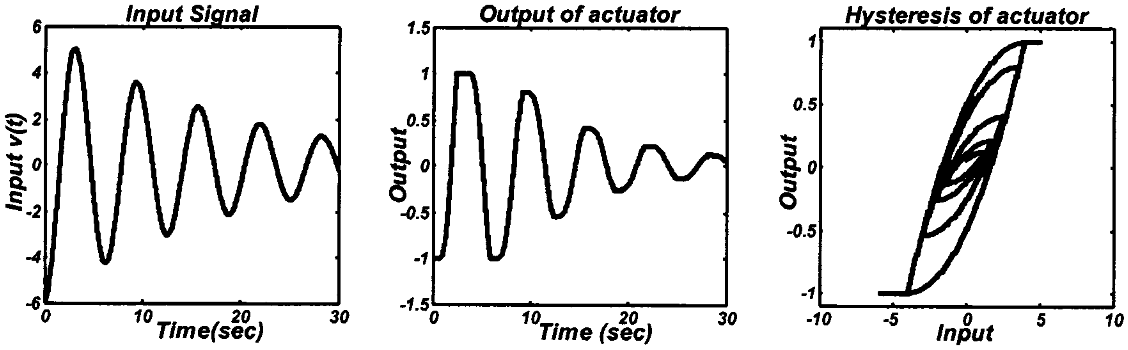


Fig.3.8 Hystersis formed with normalized uniform density distribution

When the input signal applied to the hysteresis remains as $v(t) = -6e^{-0.055t}\cos(t)$, but the density function changes as a normalized bi-variant normal distribution defined by

$$\mu(\alpha_i,\beta_j) = \mu'(\alpha_i,\beta_j)/\sum_{i=1}^{l+2}\sum_{j=1}^{i}\mu'(\alpha_i,\beta_j) \tag{3.13}$$

where $l = 100$, $i \geq j = 1,2,\ldots,l+2$, and

$$\mu'(\alpha_i,\beta_j) = \frac{1}{2\pi\sigma^2}e^{\frac{-1}{2\sigma^2}((\alpha_i+\alpha_0)^2+(\beta_j+\beta_0)^2)} \tag{3.14}$$

with $\sigma = 1.8$, $\alpha_0 = -0.1$ and $\beta_0 = 0.6$, a smooth hysteresis but with a different loop shape from that of Fig.3.8 is formed. The simulation results are shown as Fig.3.9. Note that the density vector $\Theta^T = [\theta_1,\theta_2,\ldots,\theta_{5253}]$ is also calculated by equations (3.5), (3.6), (3.9) and (3.12).

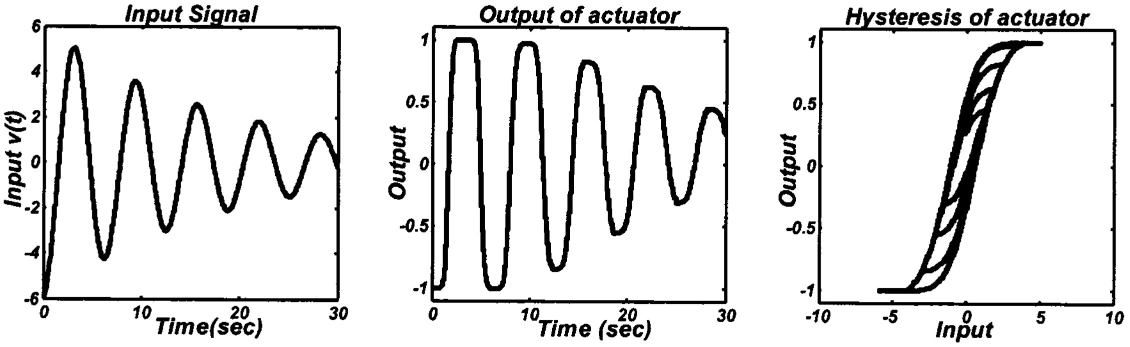


Fig.3.9 Hystersis formed with a normalized bi-variant density distribution

For the above two cases, in the beginning of the process, the local input extrema exceeded the hysteresis input region $[v^-, v^+] = [-4,4]$ which resulted in the saturation states to be reached, and the major loops were formed. As the amplitude of the input signal exponentially decayed so that local input extrema remained inside the hysteresis input region$[v^-, v^+] = [-4,4]$, the minor loops were traced inside the major loops of the hysteresis. The shapes of the hysteresis loops in these two cases were different just because a uniform density distribution was used in one case, while a more complex density function was chosen in the other case.

Remark: for an actuator with a certain hysteresis input region $[v^-, v^+]$ to create a hysteresis nonlinearity, the fineness of dividing the Preisach plane T determines the smoothness of the simulated hysteresis loops; the amplitude of the input signal $v(t)$ to the actuator determines whether the minor loops and the major loop appear or not; the density distribution of the relays influences the shape of the hysteresis loops; and the sum of all the density factors determines the outputs of the hysteresis saturated states. In the above three examples, the summation of densities are all specially chosen as 1. Thus all these hysteresis have the same saturated outputs ($u^- = -1$ and $u^+ = 1$).

Compared with the method introduced in chapter 2, which implements the hysteresis through the memory interface line, this direct method is simpler and involves less calculation. In this method, the output of every relay is assigned directly and individually, and the calculation just depends on the complexity of the hysteresis operator (relay) and the fineness of the dividing of the Preisach plane T. No mater how complex the input signal is applied on the system, the strictly defined operators (relays) can directly and promptly give the output values for every point in the Preisach plane T. Consequently, the calculation for the hysteresis output is straightforward, i.e., it is a summation of weighted outputs of all relays. This method avoids the necessity to find the memory interface line at every moment during the entire process and involving the complicated formula (2.41). This useful and simple method will be applied in forthcoming chapters where even more complex operators are employed to describe hysteresis.

3.2 Open Loop Hysteresis Compensation

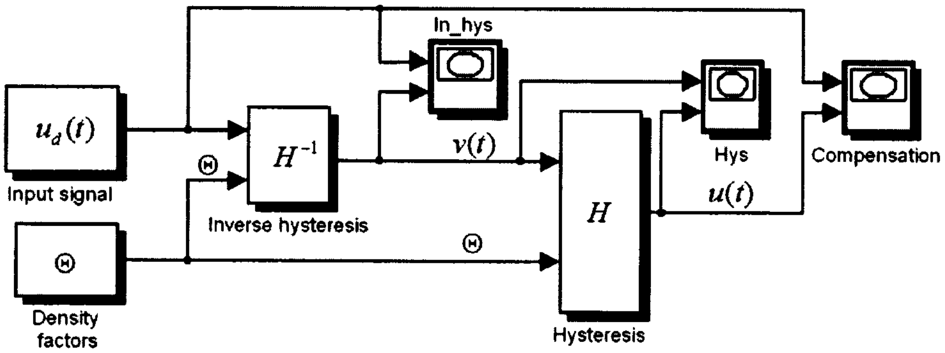


Fig.3.10 Open loop compensation using the approximated inverse Preisach model

According to the algorithm of the approximated Preisach inverse model introduced in subsection 2.4.2 of chapter 2, a simulation subsystem can be established in Simulink as the "*inverse hysteresis operator*" which will be used in the open loop compensation configuration, as shown in Fig.3.10. In the figure, the 'inverse hysteresis' block represents the "*inverse hysteresis operator*" subsystem described in subsection 2.4.2. Figures 3.11 to 3.13 show the internal programming of this block.

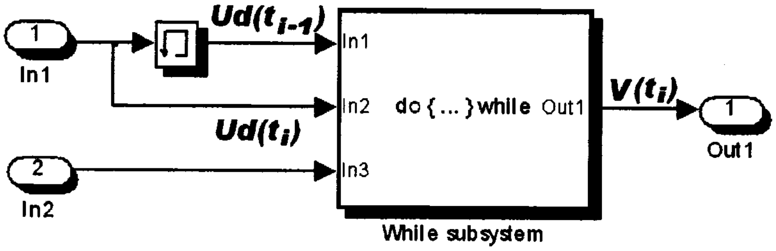


Fig.3.11 Inverse hysteresis subsystem of Fig.3.10

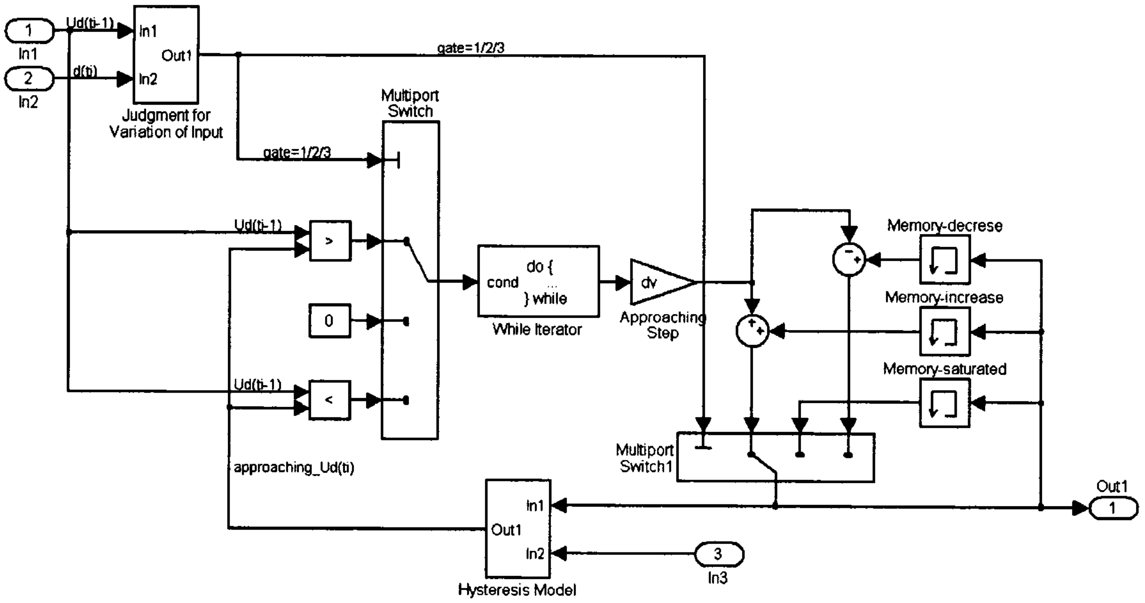


Fig.3.12 While subsystem in the inverse hysteresis subsystem of Fig.3.11

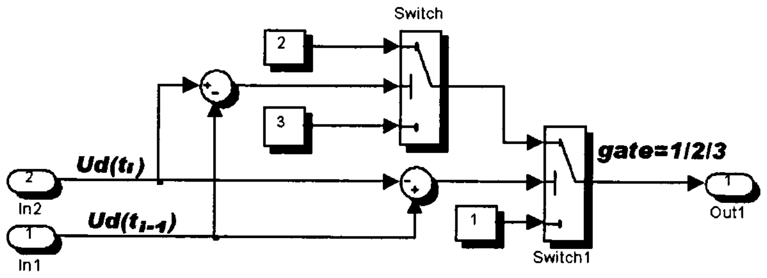


Fig.3.13 Subsystem of adjustment for variation of input of Fig.3.12

In the subsystem "*inverse hysteresis*" (see Fig.3.11), the block "*memory*" is used to the previous sampled input value $u_d(t_{i-1})$, $i=1,2,...n$ is the sampling sequence. $u_d(t_{i-1})$ and the current sampled input value $u_d(t_i)$ are fed to the "*while subsystem*" through the ports "in1" and "in2" to calculate the variation trend of the input $u_d(t)$. This adjustment is done by the subsystem "*judgment for variation of input*" (see Fig.3.13). The subsystem "*while subsystem*" (see Fig.3.12) acts as a "*while loop*" for approaching the final value of $v(t)$ for a desired $u_d(t)$.

To judge the variation of input $u_d(t)$, in Fig.3.13, the previous sampled input value$u_d(t_{i-1})$ is stored, and then the current sampled input value $u_d(t_i)$ is subtracted by the $u_d(n-1)$ to generates the judgment conditions as

$$\begin{cases} gate = 1 & if\ u_d(t_{i-1}) < u_d(t_i) \\ gate = 2 & if\ u_d(t_{i-1}) = u_d(t_i) \\ gate = 3 & if\ u_d(t_{i-1}) > u_d(t_i) \end{cases}$$

The value of the "*gate*" is used to control the channels of the "*multi-port switch*" and the "*multi-port switch 1*" in the subsystem "*while subsystem*" to allow the corresponding data flow into the "*while iterator*" and the "*hysteresis model*". The block "*while iterator*" functions as the approaching algorithm, and the "*multi-port switch 1*" controls the approaching direction. The "*hysteresis model*" includes the "*vector relay*" and "dot production" in Fig.3.6. The signal from port "*in3*" of the "*while subsystem*" in Fig.3.13 is the density vector of the hysteresis model. The signal from "*out1*" of the "*while subsystem*" in Fig.3.13 is the calculated $v(t)$, which is passed to the hysteretic actuator as its input.

Referring to Fig.3.12, for example, as the input value $u_d(t)$ is increasing at a moment of time t_i, the judgment of $u_d(t_i) - u_d(t_i - 1) > 0$ decides $"gate = 1"$, and corresponding branch $approaching\_u(t_i) > u_d(t_i - 1)$ is allowed to go through "*multi-port switch*" and to active the do-while loop, "*while iterator*". Meanwhile, the previous calculated candidate $v(t_{i-1})$ from "*memory_increase*" is allowed to go through "channel 1" of the "*multi-port switch1*". And then, $v(t_{i-1})$ is updated as a new candidate of $v(t_i)$ by increasing a step dv. This updated candidate $v(t_i)$ is then entered to the subsystem *"hysteresis operator" to* calculate a new $approaching\_u(t_i)$. The "*while_iterator*" loop continuous to calculate an integrate increasing by 1 so that the candidate value of $v(t_i)$ increases by an approaching step dv while the logical condition $u(t_i) > approaching\_u(t_i)$ is still valid. Similarly, when the input to inverse hysteresis $u_d(t_i)$ changes to decrease, both the *"multi-port switch"* and the *"multi-port switch 1"* open their *"channel 3"* to find the approximated $v(t_i)$. When the input to inverse hysteresis $u_d(t_i)$ arrives the positive or negative saturation states, i.e., $u_d(t_i) = u_d(t_{i-1})$, the *"while_iterator"* loop is skipped, the previous calculated candidate $v(t_{i-1})$ from *"memory_saturated"* is directly pass through the "*multi-port switch 1*" as the approximated $v(t_i)$.

Assume a hysteresis with a hysteresis input region $[v^-, v^+] = [-4,4]$. If the density distribution of the hysteresis is exactly known as equations (3.13) and (3.14) with $\sigma = 1.8$, $\alpha_0 = -0.1$ and $\beta_0 = 0.6$, the Preisach model $u(t) = H[v(t)]$ represented by equation (3.10) and the corresponding approximated inverse Preisach model $v(t) = H^{-1}[u(t)]$ from subsection 2.4.2 can be constructed. The open loop compensation configuration (see Fig.2.28) can be used to eliminate the nonlinearity of the hysteresis through cascading these models. Fig.3.10 is the simulation model for the open loop compensation. In this simulation model, the input signal applied to the inverse hysteresis model is set as the desired output $u_d(t)$ from the hysteresis, which is given by

$$u_d(t) = -e^{-0.02t} \cos t \tag{3.15}$$

so that the approximated inverse Preisach model is expressed as

$$v(t) = H^{-1}[u_d(t)]. \qquad (3.16)$$

When the exactly known densities are used in both models, the open loop compensation configuration results in a complete cancellation of the nonlinearity of the hysteresis because the output of hysteresis is operated as

$$u(t) = H[v(t)] = H\{H^{-1}[u_d(t)]\} = u_d(t) \qquad (3.17)$$

The simulation results are depicted as Fig.3.14.

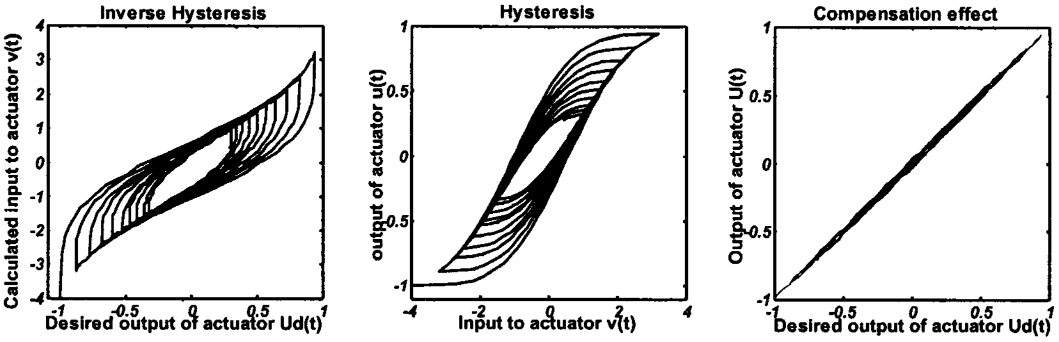


Fig.3.14 Exact compensation results

From the simulation results (Fig.3.14), it is realized that the compensation of the hysteresis is almost perfect even though there still exists a small error. This error is caused by using the approximated inverse hysteresis rather than the exact inverse hysteresis. The error can be considered as a computation error. If the approaching step dv of the approximated inverse algorithm is chosen as a very small constant as $dv \approx 0$, the computation error approaches zero.

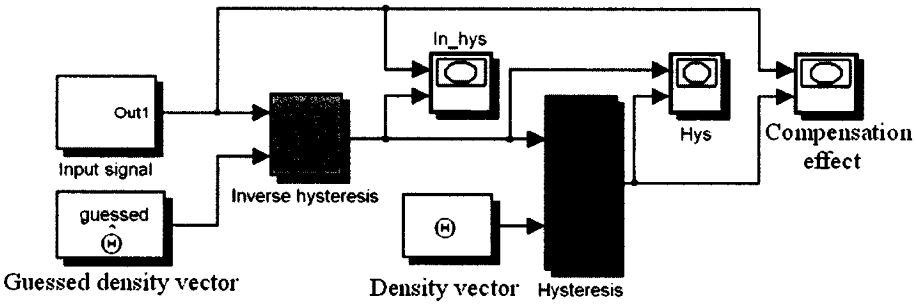


Fig.3.15 Approximated compensation using the approximated inverse Preisach model

Due to dynamics existing in the hysteresis and inaccurate identification of the densities, the density function cannot be exactly known. Unlike the exact open loop compensation, estimated densities are always applied to the inverse model to reduce the hysteresis nonlinearity. The simulation model for open loop compensation is shown as in Fig.3.15, where the guessed density vector $\hat{\Theta}$ is calculated by equations (3.5), (3.6), (3.9), and (3.12) ~ (3.14) with $\sigma = 1.6$, $\alpha_0 = -0.1$ and $\beta_0 = 0.6$; while the real density vector Θ is given as that used in the previous example of exact compensation configuration. The input signal (desired output from hysteresis) $u_d(t)$ and other conditions remain the same as that in the previous example. The compensation effect is illustrated as Fig.3.16.

Apparently, there is relative large compensation error in this approximated compensation scheme. Compared with the output amplitude of the saturation state $|u^-|=|u^+|=1$, the maximum relative error is about 13% of the output amplitude (see Fig.3.17). This error is caused by the inaccurate identified densities and the calculation error in the inverse model.

To reduce this compensation error, it is necessary to implement an adaptive approach to modify the guessed density vector $\hat{\Theta}$. This compensation configuration will be discussed in the next subsection.

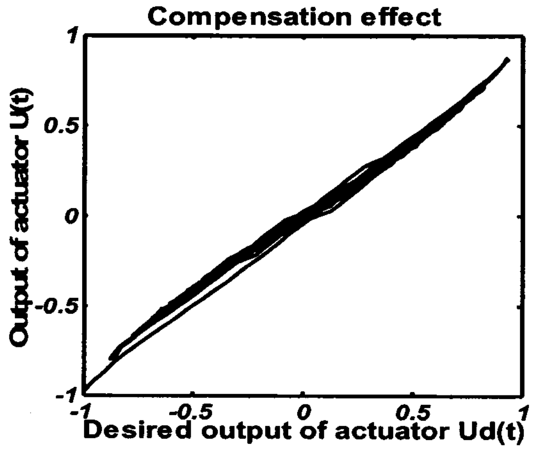


Fig.3.16 Approximated compensation results

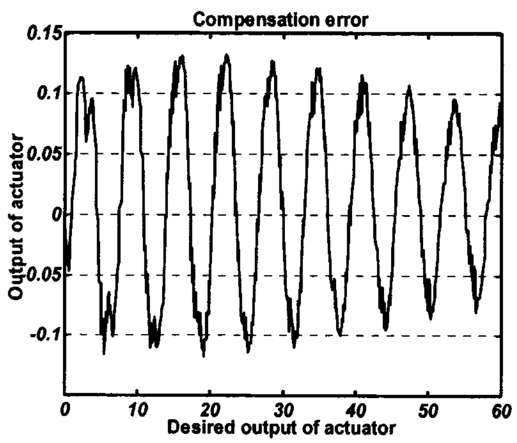


Fig.3.17 Compensation error

3.3 Adaptive Compensation to Input Hysteresis of Linear Systems

3.3.1 Design of a Gradient Adaptive Controller

For a hysteretic actuator, it can be described by a nonlinear operator $u(t) = H[v(t)]$ mapping the input $v(t)$ of the actuator into its output $u(t)$. Expressed in vector form $H(v) = \Theta^T F(v)$, where Θ is the density vector of relays and $F(v)$ is a vector of outputs of the relays, then the hysteresis model is suitable for the gradient adaptive control method. Let the output $u(t)$ of the actuator be connected to a linear plant whose characteristics are assumed be totally known. Namely, the linear plant is assumed can be totally described by a transfer function $G_p(s)$. Rather than measuring the actuator output $u(t)$ because of sensor inconvenience and cost, instead the output y of the plant is measured and is used for the design of an adaptive controller to reduce the hysteresis effect.

Assume the real hysteretic actuator can be exactly described by the linearly parameterized Preisach model $H(v) = \Theta^T F(v)$ with an exact density vector Θ and a vector $F(v)$ containing outputs of adequate relays. For a fixed dimension of the vector $F(v)$, some limited knowledge $\hat{\Theta}$ regarding the vector Θ can be predicted in advance through testing the actuator; where $\hat{\Theta}$ is called the estimated density vector. For an input $v(t)$ applied to the actuator which is modeled by an estimated Preisach model $\hat{H}(v) = \hat{\Theta}^T F(v)$, the calculated output $\hat{u}(t)$ is different from the real output $u(t)$ of the actuator. Consequently, the calculated output $\hat{y}(t)$ of the modeled plant $\hat{G}_p$, which is considered to be an exact model of the real plant G_p without any modeling errors ($\hat{G}_p = G_p$), is different from the measured output $y(t)$ of the actual plant. Therefore, the actual plant output $y(t)$ and the modeled plant output $\hat{y}(t)$ can be expressed, respectively, by

$$y = H(v)G_p = \Theta^T F(v)G_p \tag{3.18}$$

$$\hat{y} = \hat{H}(v)\hat{G}_p = \hat{\Theta}^T F(v)G_p \ . \tag{3.19}$$

There exists an error between the two outputs as

$$\begin{aligned} e(t) &= y(t) - \hat{y}(t) \\ &= [H(v) - \hat{H}(v)]G_p \\ &= (\Theta - \hat{\Theta})^T F(v)G_p \\ &= \tilde{\Theta}^T F(v)G_p \end{aligned} \tag{3.20}$$

where $\tilde{\Theta} \overset{\Delta}{=} \Theta - \hat{\Theta}$ is a vector consisting of density errors.

Based on the estimated Preisach model $\hat{H}(v) = \hat{\Theta}^T F(v)$, an inverse hysteresis $\hat{H}^{-1}$ which has the same estimated density vector $\hat{\Theta}$ can be constructed. In order to cancel or reduce the hysteresis nonlinearity, the inverse hysteresis $\hat{H}^{-1}$ is cascaded with the actuator $H(v)$ as the Fig.3.18. Due to the inaccurate $\hat{\Theta}$, the inverse hysteresis model $\hat{H}^{-1}$ cannot perfectly cancel the actuator hysteresis $H(v)$. If the parameters $\hat{\Theta}$ are estimated more accurately, the compensation effect is more acceptable. Thus, it is required to construct an adaptive controller (see Fig.3.18) to update the estimated $\hat{\Theta}$ so that it approaches the real Θ. Please be noted that $H(v) = \Theta^T F(v)$ in Fig.3.18 represents the real actuator for simulation.

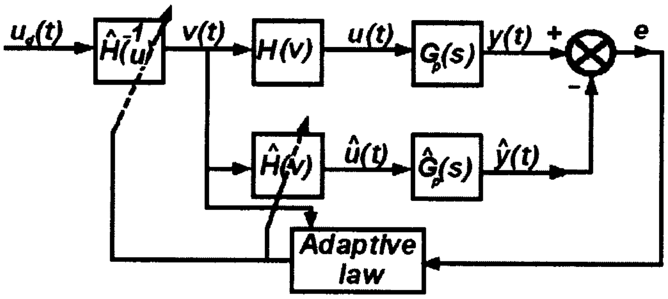


Fig.3.18 Gradient adaptive compensation configuration

The output error between the real system and modeled system is simply described by (3.20). To ensure signal boundedness, a normalized estimation error [27] is introduced as

$$\varepsilon = \frac{e}{m^2} = \frac{y - \hat{y}}{m^2} = \frac{\tilde{\Theta}^T F(v) G_p}{m^2}, \tag{3.21}$$

where,
$$m^2 = 1 + n_s = 1 + (F(v)G_p)^T (F(v)G_p). \tag{3.22}$$

and n_s is the normalizing signal designed so that

$$\frac{F(v)G_p}{n_s} \in L_\infty$$

As in section 4.34 of [27], typical choice for n_s are $n_s = [F(v)G_p]^T [F(v)G_p]$, $n_s = [F(v)G_p]^T P [F(v)G_p]$ for $P = P^T > 0$, etc.

Consider a quadratic cost function given by

$$J(\hat{\Theta}) = \frac{\varepsilon^2 m^2}{2} = \frac{(\tilde{\Theta}^T F(v) G_p)^2}{2m^2} = \frac{(y - \hat{\Theta}^T F(v) G_p)^2}{2m^2} \quad (3.23)$$

The minimizing trajectory for the cost function (3.23) is given by the negative of the gradient of $J(\hat{\Theta})$ as

$$\begin{aligned}
\dot{\hat{\Theta}} &= -\Gamma \nabla J(\hat{\Theta}) \\
&= \Gamma \frac{(y - \hat{\Theta}^T F(v) G_p) F(v) G_p}{m^2} \\
&= \Gamma \frac{(\tilde{\Theta}^T F(v) G_p)}{m^2} F(v) G_p \\
&= \Gamma \varepsilon F(v) G_p \qquad (3.24)
\end{aligned}$$

Equation (3.24) results in the adaptive update law as

$$\dot{\hat{\Theta}} = \Gamma \varepsilon F(v) G_p \qquad (3.25)$$

where, $\Gamma = \Gamma^T > 0$ is a positive scaling matrix of adaptive gains. The larger Γ is chosen, the faster $\hat{\Theta}$ converges to its true value Θ.

So far the choice of parameters $\hat{\Theta}^T = [\hat{\theta}_1, \hat{\theta}_2, ..., \hat{\theta}_i, ..., \hat{\theta}_N]$ is non-restrictive. If one has an estimate from a prior knowledge on the maximum achievable output, u^+, then clearly any individual $\hat{\theta}_i$ would never be greater than u^+, otherwise,

$$u(t) = \sum_{i=1}^{N} \theta_i \hat{\gamma}_i = \sum_{i=1}^{N} \theta_i (+1) > u^+ \text{ for } v(t) > v^+$$

and

$$u(t) = \sum_{i=1}^{N} \theta_i \hat{\gamma}_i = \sum_{i=1}^{N} \theta_i (-1) < -u^+ = u^- \quad \text{for} \quad v(t) < v^-$$

which conflict with the positive and negative saturation states. Therefore, a safe upper bound of $\max(\theta_i) = u^+$ must be used to define a set Q for $\hat{\Theta}$ as

$$\hat{\Theta} \in Q \overset{\Delta}{=} \{\hat{\theta} \in R^n : u^+ - \hat{\theta}_i \geq 0, \qquad i = 1,2,\ldots,n\} \qquad (3.26)$$

3.3.2 Simulation Studies

For simulation purposes, the real hysteretic actuator in the configuration of Fig.3.18 is represented by the Preisach model

$$H(v) = \Theta^T F(v)$$

The accurate density distribution of the hysteresis model is rearranged into a one-dimensional vector as

$$\Theta^T = [\theta_1, \theta_2, \ldots, \theta_i, \ldots, \theta_N] = [\rho_{11}, \rho_{21}, \rho_{22}, \rho_{31}, \ldots, \rho_{(l+2)(l+2)}]$$

where $l = 100$, $i \geq j = 1,2,\ldots,l+2$, $N = (l+2)(l+3)/2 = 5253$, and

$$\rho_{ij} = \mu_n(\alpha_i, \beta_j) = \mu(\alpha_i, \beta_j) / \sum_{i=1}^{l+2} \sum_{j=1}^{i} \mu(\alpha_i, \beta_j) \qquad (3.27)$$

with

$$\mu(\alpha_i, \beta_j) = \frac{1}{2\pi\sigma^2} \exp[\frac{-1}{2\sigma^2}((\alpha_i + \alpha_0)^2 + (\beta_j + \beta_0)^2)] \quad (3.28)$$

In (3.28) the parameters α_0, β_0, and σ are chosen as $\alpha_0 = -0.1$, $\beta_0 = 0.6$ and $\sigma = 1.8$.

The known linear plant is described by the transfer function

$$G_p(s) = \frac{4}{s+4} \qquad (3.29)$$

with initial condition $y(t_0) = x(t_0) = -1$.

The desired input $u_d(t)$ to the plant $G_p(s)$, which is also treated as the input of the inverse hysteresis, is defined by

$$u_d(t) = -e^{-0.02t} \cos t \qquad (3.30)$$

The estimated density vector $\hat{\Theta}^T = [\hat{\theta}_1, \hat{\theta}_2, ..., \hat{\theta}_i, ..., \hat{\theta}_N]$ is calculated by equations (3.27) and (3.28) with the same parameters except $\sigma = 1.6$ which defines a different density distribution.

The inverse hysteresis $\hat{H}^{-1}(u_d)$ is calculated by the approximate inverse algorithm introduced in subsection 2.4.2 of chapter 2 based on the estimated hysteresis $\hat{H}(v)$ with the estimated density vector $\hat{\Theta}^T$.

The adaptive control law is as given in equation (3.25). The above conditions for the simulation are listed in table3.1 and table 3.2.

Table 3.1 Actual hysteresis and estimate hysteresis

Characteristic	Actual hysteresis	Estimated hysteresis (initial estimate)
Discretization number	$l = 100, i \geq j = 1,2,...,l+1$	$l = 100, i \geq j = 1,2,...,l+1$
Number of nodes	$N = (l+2)(l+3)/2 = 5253$	$N = (l+2)(l+3)/2 = 5253$
Hysteresis input region	$v(t) \in [v^-, v^+] = [-4, +4]$	$v(t) \in [v^-, v^+] = [-4, +4]$
Density distribution	$\mu(\alpha_i, \beta_j) = \frac{1}{2\pi\sigma^2} e^{\frac{-1}{2\sigma^2}((\alpha_i+\alpha_0)^2+(\beta_j+\beta_0)^2)}$	$\hat{\mu}(\alpha_i, \beta_j) = \frac{1}{2\pi\sigma^2} e^{\frac{-1}{2\sigma^2}((\alpha_i+\alpha_0)^2+(\beta_j+\beta_0)^2)}$
	$\alpha_0 = -0.1, \beta_0 = 0.6,$ and $\sigma = 1.8$	$\alpha_0 = -0.1, \beta_0 = 0.6,$ and $\sigma = 1.6$
	$\rho_{ij} = \mu(\alpha_i, \beta_j) / \sum_{i=1}^{l+1} \sum_{j=1}^{i} \mu(\alpha_i, \beta_j)$	$\hat{\rho}_{ij} = \hat{\mu}(\alpha_i, \beta_j) / \sum_{i=1}^{l+1} \sum_{j=1}^{i} \hat{\mu}(\alpha_i, \beta_j)$

Table 3.2 Characteristic of plant

Input signal	$u_d(t) = -e^{-0.02t}\cos t$
Plant transfer function	$G_p(s) = 4/(s+4)$
with initial condition	$y(t_0) = x(t_0) = -1$
Adaptive law	$\dot{\Theta} = \Gamma \varepsilon F(v) G_p$ with $\Gamma = 15$

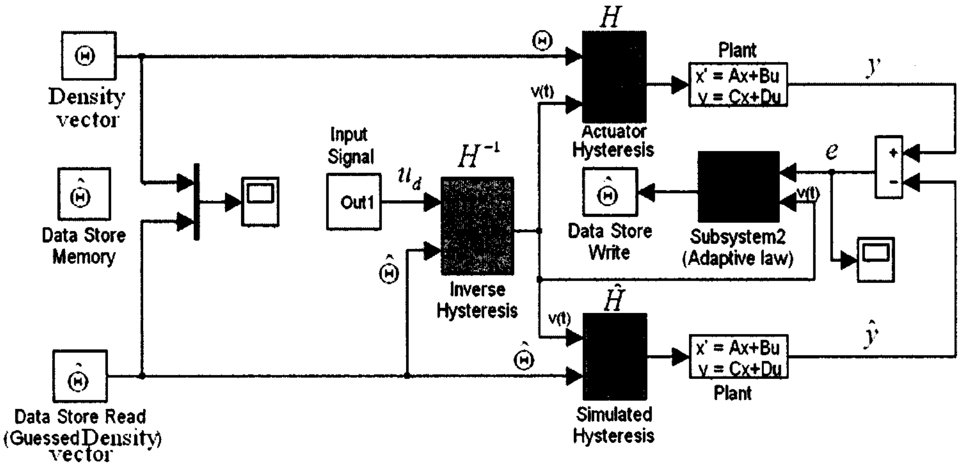


Fig.3.19 Adaptive compensation of hysteresis of actuator with hidden output

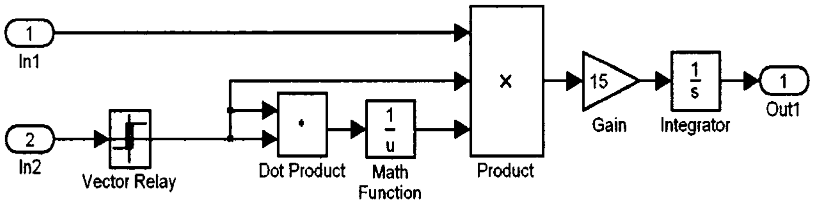


Fig.3.20 Subsystem 2 in Fig.3.19

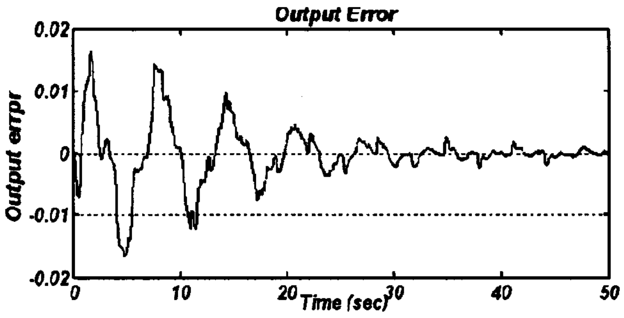


Fig.3.21 Output error of a linear plant with input hysteresis through compensation by a gradient adaptive inverse hysteresis

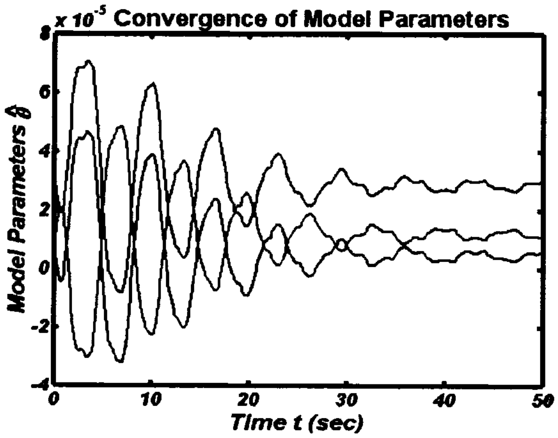


Fig.3.22 Convergence of estimated density vector $\hat{\Theta}$

The Simulink simulation model is shown Fig.3.19. The adaptive controller (see Fig.3.20) is a subsystem of the simulation model. The adaptive gain Γ is chosen through iterative simulation. The larger Γ is chosen, the faster $\hat{\Theta}$ converges to its true value Θ. But too large Γ causes much oscillation in the control system. Here, the adaptive gain is chosen as $\Gamma = 15$. Performing the simulation obtains the output error of the system as shown in Fig.3.21. The convergence to zero of the output error means that the estimated model density vector $\hat{\Theta}$ approaches to the real Θ to completely cancel the hysteresis effect of the actuator. The convergences of three adaptive densities are shown in Fig.3.22. In this chapter the convergence analysis is postponed and will be described in chapter 5.

However, for an unknown or partially unknown plant, this adaptive compensation configuration is not enough to guarantee that the plant will produce a desired output or will be able to track a reference model. For this situation, other types of adaptive robust controllers are needed. Also, in the above gradient adaptive compensation configuration, the linearly parameterized Preisach hysteresis model has a parameter vector Θ of extremely large dimension to be updated, for example $N = 5253$. This is the critical shortcoming of the model which sometimes makes the real-time adaptive control systems not implementable. Thus, a more effective hysteresis model is required to be developed.

3.4 Summary

This chapter discusses a direct method to model the Preisach class of hysteresis through the linearly parameterized Preisach hysteresis model. Simulation principles for relay and smooth hysteresis loops have been presented systematically. Online model parameter identification and the adaptive inverse compensation method through the gradient adaptive law have also been explained. Some simulations have been conducted for verifying the introduced methods. The methods for modeling hysteresis and for the compensation of hysteresis have established foundations for more complicated hysteresis elementary operators which will be introduced in the forthcoming chapters.

CHAPTER 4

MODELING AND COMPENSATION OF HYSTERESIS BY PARAMETERIZED KP HYSTERESIS MODEL

M. A. Krasnosel'skii and A. V. Pokrovskii had presented a hysteresis operator called the KP model in reference [12]. The method can be adopted to represent input hysteresis of systems with saturations. The KP model has the same integral form of weighted elementary operators as the Preisach model. But, unlike the Preisach relay being a discontinuous function on the Preisach Plane and only having two saturation output states (+1,-1), the elementary operator of the KP model which is referred to as the KP kernel is a continuous function on the Preisach plane, and has minor loops within its major loop. The KP kernel is a more compact elementary operator than the Preisach relay because it can be considered as a local memory hysteresis to be expressed as an integral of weighted relays using the Preisach representation [13]. The relationship of two elementary operators guarantees that the KP model has the same properties as the Preisach model. These advantages enable the KP model as a more effective mathematical model to formulate the Preisach class of hysteresis.

4.1 Krasnosel'skii–Pokrovkii (KP) Model

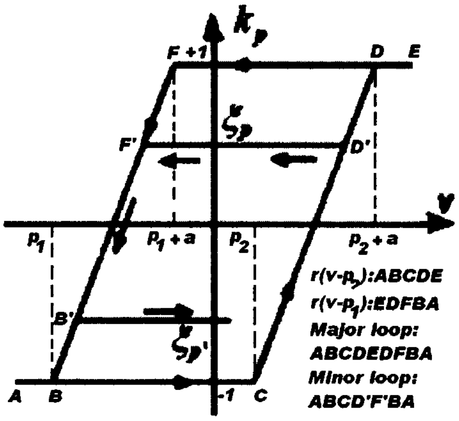


Fig.4.1 KP kernel

As presented in reference [40, 41], the Krasnosel'skii – Pokrovkii (KP) hysteresis model can be expressed as an integral of kernels (see Fig.4.1) over a specific domain by

$$u(t) = H[v](t) = \int_P k_p[v, \xi_p](t)\mu(p)dp \tag{4.1}$$

where $v(t)$: input to the hysteresis;

$u(t)$: output of hysteresis;

$H(\cdot)$: an operator to transform the input $v(t)$ into the output $u(t)$;

P : Preisach Plane (see Fig.4.2) over which hysteresis occurs. It is defined by

$$P = \{p(p_1, p_2) \in R^2 : v^+ - a \geq p_2 \geq p_1 \geq -v^-\} \tag{4.2}$$

where v- and v^+ represent input values of negative and positive saturation states of hysteresis, respectively. a is the rise-constant of the kernel k_p which will be explained later. The range of $v(t) \in [v^-, v^+]$ represents the hysteresis input domain. If the input exceeds the range the output of hysteresis remains at its saturation states. Please note that Preisach plane is notated as P rather than T as in Chapter 3 in order to find the relation between the Preisach model and KP model;

k_p : kernel function which gives output values of each KP kernel defined by a pair of parameters $p(p_1, p_2) \in P$ (see Fig.4.1) as it is subjected to the input $v(t)$;

$\xi_p(t)$: a variable to memory the previous extreme output of kernel parameterized by p;

$\mu(p)$: Density of the kernel k_p, which is utilized to weight the output of the kernel

k_p. Each point $p(p_1,p_2)$ in the Preisach plane P is associated with a kernel k_p and has its specific density value $\mu(p_1,p_2)$. The function to describe densities $\mu(p_1,p_2)$ of all points in the Preisach plane P is called the density function or density distribution of the KP model. Please note that the density function of the KP model is different from the density of the Preisach hysteresis model for modeling a particular hysteresis.

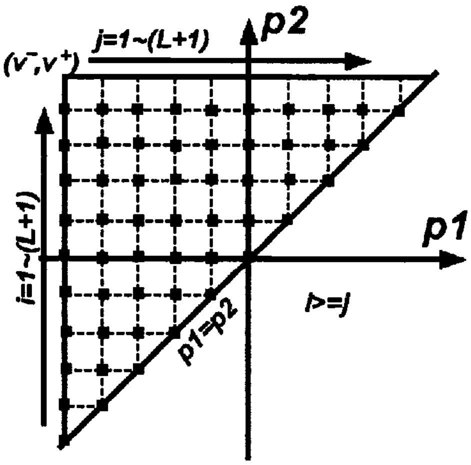


Fig.4.2 The Preisach plane P

The integral KP model given by equation (4.1) can be interpreted as a parallel connection of an infinite number of weighted kernels (see Fig.4.3).

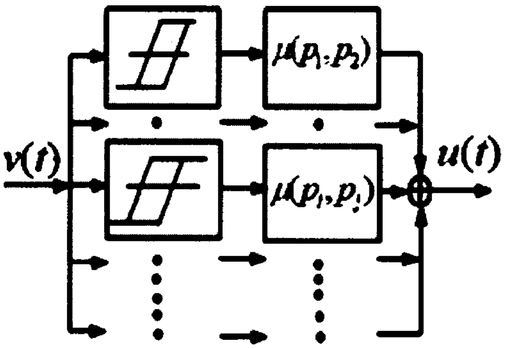


Fig.4.3 Parallel connection of weighted kernels

The kernel function $k_p[v,\xi_p](t)$ (see Fig.4.1) with parameters (p_1,p_2,a) is expressed by

$$k_p[v,\xi_p](t)=\begin{cases}\max\ \{\xi_p(t),r(v(t)-p_2)\} & for\ \ \dot{v}\geq 0\\ \\ \min\ \{\xi_p(t),r(v(t)-p_1)\} & for\ \ \dot{v}\leq 0\end{cases}\tag{4.3}$$

where the value of the memory variable $\xi_p(t)$ depends on the kernel k_p and is updated whenever the rate of $v(t)$changes sign. For example, if the input $v(t)$ starts from a value less than v^-, i.e., $v(t_0)<v^-$,both initial values of $\xi_p(t_0)$and output$k_p[v(t_0),\xi_p(t_0)]$ equal -1. As the input $v(t)$monotonically increases from $v(t_0)$to a value$v(t_1)$ and then tends to decrease at timet_1 , $\dot{v}(t)$switches sign at the timet_1, i.e.,$sign(\dot{v}(t_1^+))=-sign(\dot{v}(t_1^-))$. Each $\xi_p(t_1)$ updates to$k_p[v(t_1),\xi_p(t_0)]$ and retains the value until the input $v(t)$ tends to increase again at a timet_2, i.e.,

$$sign\ \ (\dot{v}(t_2^+))=-sign\ \ (\dot{v}(t_2^-))=-sign\ \ (\dot{v}(t_1^+)).$$

Thus, the memory variable $\xi_p(t)$ is expressed as

$$\xi_p(t)=\begin{cases}-1 & if\ \ t=t_0\\ \\ k_p[v(t),\xi_p(t_{i-1})] & if\ \ t=t_i>t_{i-1}\ \ and\ \ sign\ \ (\dot{v}(t^+))=-sign\ \ (\dot{v}(t^-))\\ \\ \xi_p(t_{i-1}) & if\ \ t_i\geq t>t_{i-1}\ \ and\ \ sign\ \ (\dot{v}(t^+))=sign\ \ (\dot{v}(t^-))\end{cases}\tag{4.4}$$

for $i=1,2,....$ representing the ith turning point. Equation (4.4) will be made clearer by some case studies presented below.

In equation (4.3), the boundary functions $r(v(t)-p_1)$ and $r(v(t)-p_2)$ form the major loop $(ABCDEDFBA)$ of the kernel k_p between -1 and +1(see Fig.4.1). Any other loops which locate inside the major loop and are not enveloped by the boundaries $r(v(t)-p_1)$ and $r(v(t)-p_2)$ are called minor loops of the kernel k_p , for example, $(ABCD'F'BA)$. It can be seen that the width of the kernel is determined by the switching input values, p_1 and p_2, while the transfer slope between -1 and $+1$ is determined by the rise-constant a. The ridge function $r(v(t))$ shown in Fig.4.4 is given by

$$r[v](t) = \begin{cases} -1 & v(t) < 0 \\ -1 + 2v(t)/a & 0 \le v(t) \le a \\ +1 & v(t) > a \end{cases} \tag{4.5}$$

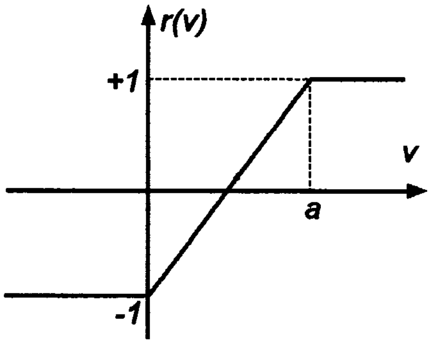


Fig.4.4 Ridge function of KP kernel k_p

Considering equations (4.3), (4.4) and (4.5), the KP kernel can be explained as followings:

Case A: As the input $v(t)$ keeps increasing from a value less than p_2 after the initial time $t = 0$, ($v(t) \le p_2$, $\dot{v}(t) \ge 0$), according to equation (4.4), the memory state $\xi_p(t) = -1$ because the hysteresis has not changed the inputs variation sign from the beginning. Referring to equation (4.5), one has $r_2 = r(v(t) - p_2) = -1$ since $v(t) - p_2 \le 0$. From equation (4.3), the output of the kernel is calculated as

$$k_p[v, \xi_p](t) = \max\ \{\xi_p(t), r(v(t) - p_2)\}$$

$$= \max(-1,-1) = -1 \quad for \quad \dot{v}(t) \ge 0 \tag{4.6}$$

Case B: As the input $v(t)$ passes the value p_2 and keeps increasing, ($v(t) > p_2, \dot{v}(t) \geq 0$), the kernel hysteresis moves along its ascending ridge $r_2 = r(v(t) - p_2)$. Since $v(t) > p_2$, from the ridge function (4.5), one has

$$r(v(t) - p_2) = -1 + 2 \cdot [v(t) - p_2]/a > -1. \tag{4.7}$$

The memory state $\xi_p(t) = -1$ because the input $v(t)$ remains increasing. From equations (4.3) and (4.7), the output of the kernel is obtained as

$$\begin{aligned} k_p[v, \xi_p](t) &= \max\{\xi_p(t), r(v(t) - p_2)\} \\ &= \max(-1, -1 + 2 \cdot [v(t) - p_2]/a) \\ &= -1 + 2 \cdot [v(t) - p_2]/a \quad for \quad \dot{v}(t) \geq 0 \end{aligned} \tag{4.8}$$

Case C: Specially, the kernel hysteresis will get its positive saturation state when the input $v(t)$ passes the value $p_2 + a$. That means the output of the kernel equals +1. This fact can also be determined by equations (4.3), (4.4) and (4.5). Since $v(t) - p_2 > a$, one has $r_2 = r(v(t) - p_2) = 1$. From equation (4.3), the output of the kernel is calculated as

$$\begin{aligned} k_p[v, \xi_p](t) &= \max\{\xi_p(t), r(v(t) - p_2)\} \\ &= \max(-1, +1) = +1 \, for \quad \dot{v}(t) \geq 0 \end{aligned} \tag{4.9}$$

Case D: If the input $v(t)$ changes its variation trend to decrease after increasing to a value $v(t')$ between p_2 and $p_2 + a$, (i.e., $p_2 \leq v(t') \leq p_2 + a$, $\dot{v}(t) \leq 0$ for $t \geq t'$), according to equations (4.4) and (4.8), the memory state $\xi_p(t) = -1 + 2 \cdot [v(t') - p_2]/a$ for $t \geq t'$ because the new memory state will be updated to store the previous output of kernel at the instant t'.

Since $p_2 \leq v(t') \leq p_2 + a$ and $\dot{v}(t) \leq 0$, one has

$$\begin{cases} -1 \le -1 + 2 \cdot [v(t') - p_2]/a \le 1 \\ \\ p_1 \le v(t') - (p_2 - p_1) \le p_1 + a \end{cases} \tag{4.10}$$

Considering equation (4.5), one has

$$\begin{cases} r(v(t) - p_1) = -1 & \text{if} \quad v(t) < p_1 \\ \\ r(v(t) - p_1) = -1 + 2 \cdot [v(t) - p_1]/a & \text{if} \quad p_1 \le v(t) \le p_1 + a \\ \\ r(v(t) - p_1) = +1 & \text{if} \quad v(t) > p_1 + a \end{cases} \tag{4.11}$$

From equation (4.3), the output of the kernel is calculated as

$$k_p[v, \xi_p](t) = \min\,\{\xi_p(t), r(v(t) - p_1)\} \quad \text{since } \dot{v}(t) \le 0 \text{ for } t \ge t'$$

$$= \min\{-1 + 2 \cdot [v(t') - p_2]/a, \quad r(v(t) - p_1)\}$$

Substituting equation (4.11) into the above equation yields

$$k_p[v,\xi_p](t) = \begin{cases} \min\,\{-1+2\cdot[v(t')-p_2]/a\ ,\ +1\} & \text{if } v(t)\in(p_1+a, v(t')] \\ \\ \min\,\{-1+2\cdot[v(t')-p_2]/a\ ,\ -1+2\cdot[v(t)-p_1]/a\} & \text{if } v(t)\in(v(t')-(p_2-p_1), p_1+a] \\ \\ \min\,\{-1+2\cdot[v(t')-p_2]/a\ ,\ -1+2\cdot[v(t)-p_1]/a\} & \text{if } v(t)\in[p_1, v(t')-(p_2-p_1)] \\ \\ \min\,\{-1+2\cdot[v(t')-p_2]/a\ ,\ -1\} & \text{if } v(t)<p_1 \end{cases} \tag{4.12}$$

Equation (4.12) is defined over four domains while the descending boundary (4.11) is defined over three domains. The domains of equation (4.12) are formed by the intersecting of the descending boundary, $r(v(t) - p_1) : EDFBA$, and the horizontal line of memory state, $k_p = \xi_p(t) = -1 + 2 \cdot [v(t') - p_2]/a$ (see Fig. 4.5).

Considering equation (4.10), equation (4.12) yields

$$k_p[v,\xi_p](t)=\begin{cases} -1+2\cdot[v(t')-p_2]/a & \text{if} \quad v(t)\in(p_1+a,v(t')] \\ -1+2\cdot[v(t')-p_2]/a & \text{if} \quad v(t)\in(v(t')-(p_2-p_1),p_1+a] \\ -1+2\cdot[v(t)-p_1]/a & \text{if} \quad v(t)\in[p_1,v(t')-(p_2-p_1)] \\ -1 & \text{if} \quad v(t)<p_1 \end{cases} \tag{4.13}$$

which defines a descending curve $D'FBA$ of a minor loop.

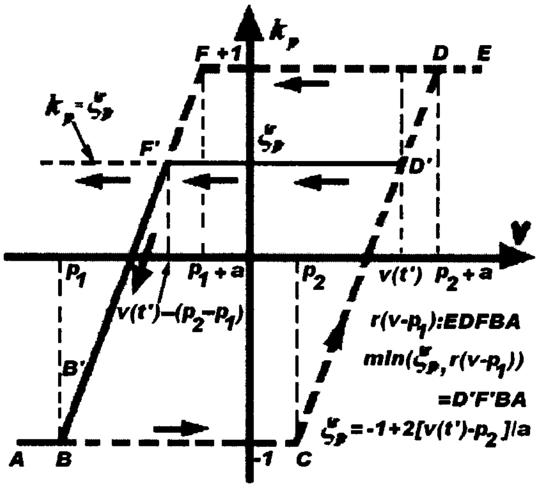


Fig. 4.5 Formation of the descending curves of minor loops

Case E: Specially, if the input $v(t)$ changes its variation trend to decrease after increasing to positive saturation state $v(t') \geq p_2 + a$, the memory state $\xi_p(t) = +1$. Consequently, equation (4.12) changes as

$$k_p[v,\xi_p](t)=\begin{cases} \min(+1,+1) & \text{if} \quad v(t)\in(p_1+a,v(t')] \\ \min(+1,-1+2\cdot[v(t)-p_1]/a) & \text{if} \quad v(t)\in(v(t')-(p_2-p_1),p_1+a] \\ \min(+1,-1+2\cdot[v(t)-p_1]/a) & \text{if} \quad v(t)\in[p_1,v(t')-(p_2-p_1)] \\ \min(+1,-1) & \text{if} \quad v(t)<p_1 \end{cases}$$

$$= \begin{cases} +1 & if \ \ v(t) \in (p_1 + a, v(t')] \\ -1 + 2 \cdot [v(t) - p_1]/a & if \ \ v(t) \in [p_1, p_1 + a] \\ -1 & if \ \ v(t) < p_1 \end{cases} \tag{4.14}$$

More specially, if the input $v(t)$ returns to decrease after increasing to a value $v(t')$ which was still in negative saturation state $v(t') \le p_2$, the memory state $\xi_p(t) = -1$. Consequently, equation (4.12) changes as

$$k_p[v,\xi_p](t) = \begin{cases} \min(-1,+1) & if \ \ v(t) \in (p_1 + a, v(t')] \\ \min(-1,-1+2\cdot[v(t)-p_1]/a) & if \ \ v(t) \in (v(t') - (p_2 - p_1), p_1 + a] \\ \min(-1,-1+2\cdot[v(t)-p_1]/a) & if \ \ v(t) \in [p_1, v(t') - (p_2 - p_1)] \\ \min(-1,-1) & if \ \ v(t) < p_1 \end{cases}$$

$$= -1 \qquad for \quad v(t) < v(t') \le p_2 \tag{4.15}$$

Case F: By the same reasoning, the ascending curves of minor loops can be formed if the input $v(t)$ returns to increase after decreasing to a value $v(t'')$. In this case, it is only considered when the input $v(t)$ changes its variation trend to increase after decreasing to a value $v(t'')$ between p_1 and $p_1 + a$, *i.e.*, $p_1 \le v(t'') \le p_1 + a, \ \dot{v}(t) \ge 0, \ for \ \ t \ge t''$. According to equations (4.3), (4.4) and (4.5), the memory state $\xi_p(t) = -1 + 2 \cdot [v(t'') - p_1]/a$ for $t \ge t''$ because the new memory state will be updated to store the previous output of kernel at the instant t''.

Since $p_1 \le v(t'') \le p_1 + a$, and $\dot{v}(t) \ge 0$, one has

$$\begin{cases} -1 \le -1 + 2 \cdot [v(t'') - p_1]/a \le 1 \\ p_2 \le v(t'') + (p_2 - p_1) \le p_2 + a \end{cases} \tag{4.16}$$

Considering equation (4.5), one has

$$\begin{cases} r(v(t)-p_2)=-1 & \textit{if} \quad v(t)<p_2 \\ r(v(t)-p_2)=-1+2\cdot[v(t)-p_2]/a & \textit{if} \quad p_2 \le v(t) \le p_2+a \\ r(v(t)-p_2)=+1 & \textit{if} \quad v(t)>p_2+a \end{cases} \tag{4.17}$$

From equation (4.3), the output of the kernel is calculated as

$$\begin{aligned} k_p[v,\xi_{p'}](t) &= \max\left\{\xi_{p'}(t), r(v(t)-p_2)\right\} \quad \text{since } \dot{v}(t)\ge 0 \text{ for } t \ge t'' \\ &= \max\left\{-1+2\cdot[v(t'')-p_1]/a, \quad r(v(t)-p_2)\right\} \end{aligned}$$

Substituting equation (4.17) into the above equation yields

$$k_p[v,\xi_p](t)=\begin{cases} \max(-1+2\cdot[v(t'')-p_1]/a,-1) & \textit{if} \quad v(t)\in(v(t''),p_2] \\ \max(-1+2\cdot[v(t'')-p_1]/a,-1+2\cdot[v(t)-p_2]/a) & \textit{if} \quad v(t)\in(p_2,v(t'')+(p_2-p_1)) \\ \max(-1+2\cdot[v(t'')-p_1]/a,-1+2\cdot[v(t)-p_2]/a) & \textit{if} \quad v(t)\in[v(t'')+(p_2-p_1),p_2+a] \\ \max(-1+2\cdot[v(t'')-p_1]/a,+1) & \textit{if} \quad v(t)\ge p_2+a \end{cases} \tag{4.18}$$

Equation (4.18) is defined over four adjacent domains while the descending boundary $r(v(t)-p_1):EDFBA$ is defined over three domains (see equation (4.17)). The four domains of equation (4.18) are formed by the intersecting of the descending boundary and the horizontal line of memory state, i.e., $k_p=\xi_{p'}(t)=-1+2\cdot[v(t'')-p_1]/a$ (see Fig. 4.6).

Considering equation (4.16), equation (4.18) changes as

$$k_p[v,\xi_{p'}](t)=\begin{cases} -1+2\cdot[v(t'')-p_1]/a & \textit{if} \quad v(t)\in(v(t''),p_2] \\ -1+2\cdot[v(t'')-p_1]/a & \textit{if} \quad v(t)\in(p_2,v(t')+(p_2-p_1)) \\ -1+2\cdot[v(t)-p_2]/a & \textit{if} \quad v(t)\in[v(t')+(p_2-p_1),p_2+a] \\ +1 & \textit{if} \quad v(t)\ge p_2+a \end{cases} \tag{4.19}$$

which defines an ascending curve $B'C'DE$ of a minor loop.

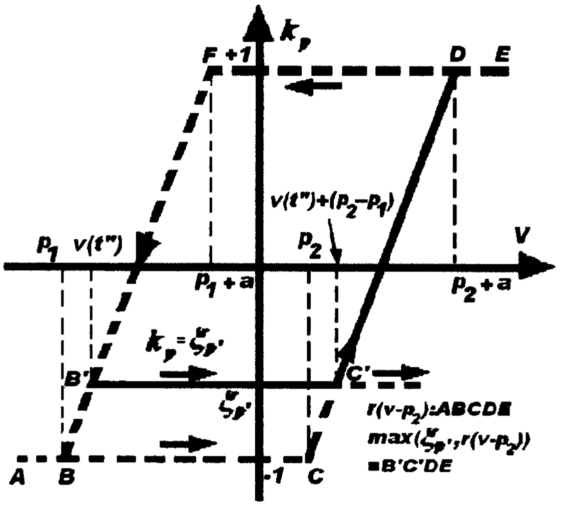


Fig. 4.6 Formation of the ascending curves of minor loops

4.2. Definition of Compensator of KP Operator

Referring to the definition of the KP kernel function $k_p[v,\xi_p](t)$, a related elementary compensator $m_p[v,\psi,\varsigma_p](t)$ (see Fig.4.7) can be defined so that

$$k_p[v,\xi_p](t)+m_p[v,\psi,\varsigma_p](t)=v(t) \tag{4.20}$$

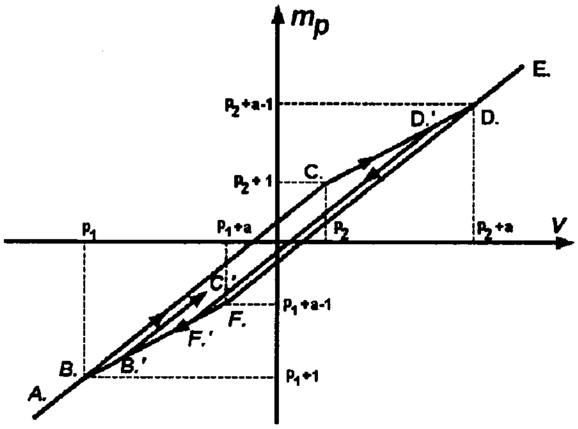


Fig.4.7 Compensator m_p related to k_p

The mathematical definition of the elementary compensator $m_p[v,\psi,\varsigma_p](t)$ related to the kernel function $k_p[v,\xi_p](t)$ is be expressed by

$$m_p[v,\psi,\varsigma_p](t)=\begin{cases}\min\ \{v(t)-\psi(t)+\varsigma_p(t),\ r_c[v(t)-p_2]+p_2+1\}\ \ for\ \dot{v}\geq 0\\ \\ \max\ \{v(t)-\psi(t)+\varsigma_p(t),\ r_c[v(t)-p_2]+p_2+1\}\ \ for\ \dot{v}\leq 0\end{cases}\tag{4.21}$$

where $(\psi(t),\varsigma_p(t))$ is the coordinate of a turning point, for example such as points $D_\bullet'$ and $B_\bullet'$ in Fig.4.7. The coordinate is defined by

$$\psi(t)=\begin{cases}v(t_0)\quad if\quad t=t_0\\ \\ v(t_i)\quad if\quad t=t_i>t_{i-1}\ \ and\ \ sign\ \ (\dot{v}(t^+))=-sign\ \ (\dot{v}(t^-))\\ \\ v(t_{i-1})\quad if\quad t_i\geq t>t_{i-1}\ \ and\ \ sign\ \ (\dot{v}(t^+))=sign\ \ (\dot{v}(t^-))\end{cases}\tag{4.22}$$

and

$$\varsigma_p(t)=\begin{cases}v(t_0)+1 & if\ \ t=t_0\\ \\ m_p[v(t_i),\psi(t_{i-1}),\varsigma_p(t_{i-1})] & if\ \ t=t_i>t_{i-1}\ \ and\ sign\ (\dot{v}(t^+))=-sign\ (\dot{v}(t^-))\\ \\ \varsigma_p(t_{i-1}) & if\ \ t_i\geq t>t_{i-1}\ \ and\ sign\ (\dot{v}(t^+))=sign\ (\dot{v}(t^-)\end{cases}\tag{4.23}$$

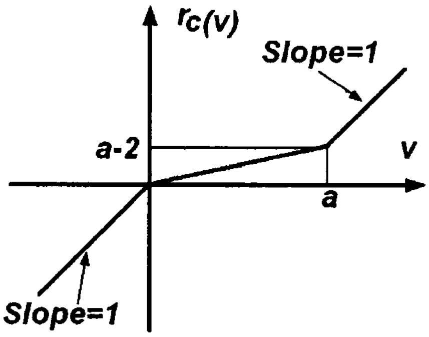


Fig.4.8 Ridge function of compensator m_p

It should be noted that the horizontal coordinates $\psi(t)$ of each turning point of different kernels can have the same value, but the vertical coordinates $\varsigma_p(t)$ of each turning point are kernel related. The ridge function $r_c[v](t)$ (see Fig.4.8) is defined by

$$r_c[v](t) = \begin{cases} v(t) & v(t) < 0 \\ (1-2/a)v(t) & 0 \le v(t) \le a \\ v(t)-2 & v(t) > a \end{cases} \tag{4.24}$$

The formation principle of inner loops in the compensator is similar with that of the KP kernel except for the different choice of the operator "*max*" or "*min*" in (4.3) and (4.21). This can be explained as follows with reference to Fig.4.9. As the input $v(t)$ changes its variation trend from increasing to decreasing, $i.e., from\ \dot{v}(t) \ge 0\ \ to\ \ \dot{v}(t) \le 0$, the curves $(A.B.F.'D.'D.E.)$ of the inner loop of the compensator are above the decreasing boundary $r_c(v(t)-p_1)$: $A.B.F.D.E.$, but the curves $(ABF'D'DE)$ of the inner loop of the KP kernel are below its decreasing boundary $r(v(t)-p_1)$: $ABFDE$. Thus, the "*max*" operator is taken in equation (4.21) but the "min" operator is selected in (4.3). Similarly, the "*min*" operator is taken in equation (4.21) but the "max" operator is selected in (4.3) if the input $v(t)$ changes its variation trend from decreasing to increasing $i.e., from\ \dot{v}(t) \le 0\ \ to\ \ \dot{v}(t) \ge 0$.

The weighting factor $\mu(p) = \mu(p_1, p_2)$ is the same density function as that of the kernel k_p. Thus, to describe a hysteresis, either kernel k_p or compensator m_p is used in the KP model, and the same density function is used.

There exists a relationship between the vertical coordinates $\varsigma_p(t_i)$ and $\xi_p(t_i)$ of the i th turning point of a compensator m_p and it corresponding kernel k_p as

$$\varsigma_p(t_i) + \xi_p(t_i) = v(t_i) \tag{4.25}$$

With these relationships (4.25) and the definitions of elementary compensator $m_p[v,\psi,\zeta_p](t)$ and its kernel $k_p[v,\xi_p](t)$, it can be easy proven that the relationship between the compensator and its related kernel is mathematically expressed by equation (4.20) and graphically related as shown in Fig.4.9.

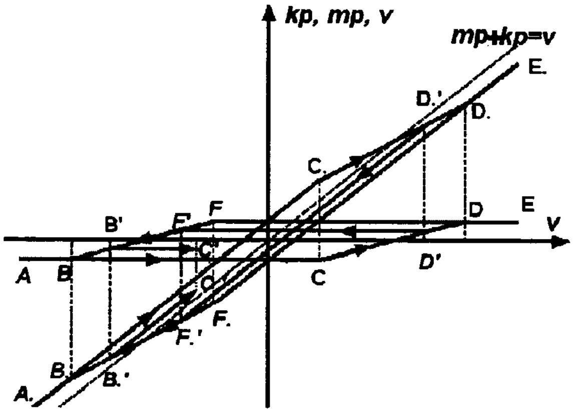


Fig.4.9 Relationship between k_p and m_p

From equation (4.20) which relates the compensator m_p to the kernel k_p, thus the KP hysteresis model (4.1) can be alternatively described by an integral of compensators over the hysteresis plane P as

$$u(t) = H[v](t) = v(t)\int_P \mu(p)dp - \int_P m_p[v,\zeta_p,\psi_p](t)\mu(p)dp$$

$$= cv(t) - \int_{v^-}^{v^+}\int_{v^-}^{p_2} m_{p_1p_2}[v,\zeta_{p_1p_2},\psi_{p_1p_2}](t)\cdot\mu(p_1,p_2)dp_1dp_2 \quad (4.26)$$

with $c \overset{\Delta}{=} \int_P \mu(p)dp$ being a constant for a particular hysteresis with a certain density distribution $\mu(p)$ over the region P.

4.3 Relationship between KP Model and Preisach Model

The KP kernel shown in Fig.4.1 is a hysteresis with local memory effect; i.e., it has a set of inner curves which consist of line segments parallel to the input axis $v(t)$. Referring to equation (4.5), the output of the inner curves are calculated as

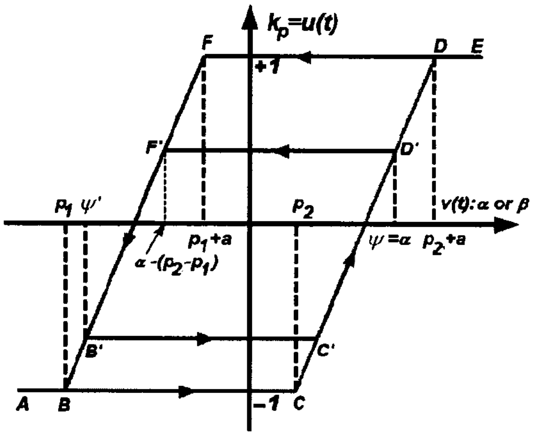


Fig.4.10 KP kernel as a rate-independent local memory hysteresis

$$u(t) = k_p = \begin{cases} 2(\psi - p_2)/a - 1 & \psi - (p_2 - p_1) \le v(t) \le \psi \\ 2[v(t) - p_1]/a - 1 & p_1 \le v(t) \le \psi - (p_2 - p_1) \end{cases} \quad \text{for} \quad \dot{v}(t) \le 0 \quad (4.27a)$$

$$u(t) = k_p = \begin{cases} 2(\psi' - p_1)/a - 1 & \psi' \le v(t) \le \psi' + (p_2 - p_1) \\ 2[v(t) - p_2]/a - 1 & \psi' \le v(t) \le p_2 + a \end{cases} \quad \text{for} \quad \dot{v}(t) \ge 0 \quad (4.27b)$$

where ψ is the input coordinate $v(t)$ of the turning point from increase to decrease while ψ' is the input value of turning point from decrease to increase. Thus, the output of the inner curves are completely characterized by the parameters p_1, p_2, a, and $v(t)$. It can be proved that KP kernel can be described with the Preisach representation [13] as

$$k_p[v](t) = \iint_T \mu(\alpha,\beta)\hat{\gamma}_{\alpha\beta}[v](t)d\alpha d\beta, \qquad (4.28)$$

where $T = \{(\alpha,\beta) \in R^2 : v^+ \ge \alpha \ge \beta \ge v^-\}$ denotes the Preisach plane, $\hat{\gamma}_{\alpha\beta}$ are relays characterized by the parameters as α, β and ± 1, with α and β denoting the increasing and decreasing switching inputs of the relays respectively. Without loss of generality $v^{-} = p_1$ and $v^{+} = p_2 + a$ are assumed be symmetric about output axis $u(t)$, i.e., $v^{+} = p_2 + a = -v^{-} = -p_1$. The weighting function $\mu(\alpha, \beta)$ can be calculated by using formula (2.29) [13]

$$\mu(\alpha, \beta) = \frac{1}{2} \frac{\partial^2 u_{\alpha\beta}}{\partial\alpha\partial\beta} \tag{2.29}$$

Please note that $u_{\alpha\beta}$ in equation (4.29) is the output of points in the first order descending curve (*FOD*), for example, a *FOD* curve is shown as the segments $DF'B$ in Fig.4.10.

From Fig.4.10 It is clear that for any α such that $p_2 < \alpha < p_2 + a$ the corresponding *FOD* curve $u_{\alpha\beta}$ consists of two parts: a particular inner curve $D'F'$ along which $u_{\alpha\beta}$ remains constant for a given α, and a part of the limiting descending branch $F'B$ on which $u_{\alpha\beta}$ varies its value while (see Fig.4.11)

$$(\alpha, \beta) \in \Omega \overset{\Delta}{=} \{(\alpha, \beta) | p_1 \le \beta \le p_1 + a \quad and \quad \beta + p_2 - p_1 \le \alpha < p_2 + a\}.$$

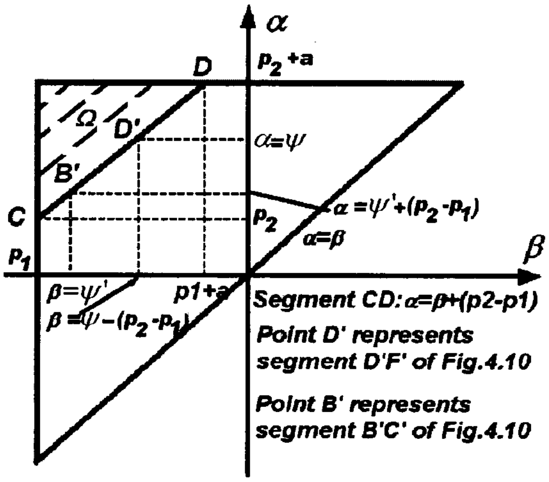


Fig. 4.11 Support region in the Preisach plane
(α and β are the parameters of the Preisach model)

From the above remarks and equation (4.27a), one finds

$$k_p = u_{\alpha\beta} = \begin{cases} 2(\beta - p_1)/a - 1 & if \ (\alpha,\beta) \in \Omega \\ 2[\alpha - (p_2 + a)]/a - 1 & if \ (\alpha,\beta) \notin \Omega \end{cases} \tag{4.29}$$

Please note that $\psi = \alpha$ and $v(t) = \alpha$ in equation (4.27a) when $\dot{v}(t) \le 0$. Taking the partial derivative of equation (4.29) with respect to β gives

$$\frac{\partial k_p}{\partial \beta} = \frac{\partial u}{\partial \beta} = \frac{\partial u_{\alpha\beta}}{\partial \beta} = \begin{cases} \frac{2}{a} & if \ (\alpha , \beta) \in \Omega \\ 0 & if \ (\alpha , \beta) \notin \Omega \end{cases} \tag{4.30}$$

Equation (4.30) means that for a given α, $\partial k_p / \partial \beta$ has an uniform value $2/a$ when $(\alpha,\beta) \in \Omega$ and an uniform value 0 when $(\alpha,\beta) \notin \Omega$. The change of $\partial k_p / \partial \beta$ is the steepest when $\beta = \alpha - (p_2 - p_1)$ (see Fig.4.11).

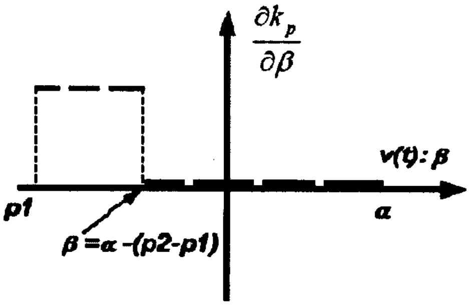


Fig.4.12 Function $\partial k_p / \partial \beta$

In Fig.4.12, an α corresponds to a $\beta = \alpha - (p_2 - p_1)$. Thus, the partial derivative of equation (4.30) with respect to α gives

$$\frac{\partial^2 k_p}{\partial \alpha \partial \beta} = \frac{\partial^2 u_{\alpha\beta}}{\partial \alpha \partial \beta} = \begin{cases} \frac{2}{a} \delta(\alpha - \beta - p_2 + p_1) & if \ p_2 \le \alpha \le p_2 + a \\ 0 & otherwise \end{cases} \tag{4.31}$$

where $\delta(\alpha - \beta - p_2 + p_1)$ is a delta function defining when $\alpha = \beta - p_2 + p_1$. Consequently,

$$\mu_{\alpha\beta} = \frac{1}{2} \frac{\partial^2 u_{\alpha\beta}}{\partial \alpha \partial \beta} = \begin{cases} \frac{1}{a} \delta(\alpha - \beta - p_2 + p_1) & if \ p_2 \le \alpha \le p_2 + a \\ 0 & otherwise \end{cases} \tag{4.32}$$

In equation (4.32), $\mu_{\alpha\beta}$ has nonzero value when $p_2 \le \alpha \le p_2 + a$, which corresponds to the domain Ω shown as in Fig.4.11. Thus, one has

$$u(t) = \iint_T \mu_{\alpha\beta}\hat{\gamma}_{\alpha\beta}[v](t)d\alpha d\beta$$

$$= \iint_{p_2+a \ge \alpha \ge \beta \ge p_1} \mu_{\alpha\beta}\hat{\gamma}_{\alpha\beta}[v](t)d\alpha d\beta$$

$$= \iint_\Omega \mu_{\alpha\beta}\hat{\gamma}_{\alpha\beta}[v](t)d\alpha d\beta$$

Substituting equation (4.32) into the above equation gives

$$u(t) = \frac{1}{a}\int_{p_1}^{p_1+a}\int_{p_2}^{p_2+a} \delta(\alpha - \beta - p_2 + p_1)\hat{\gamma}_{\alpha\beta}[v](t)d\alpha d\beta \qquad (4.33)$$

From Fig.4.11, segment *CD* is described by $\alpha = \beta + p_2 - p_1$. Thus, if α is considered as a variable while β is considered as a constant, one has

$$\delta(0) = \delta(\alpha - \beta - p_2 + p_1)$$

Similarly, if β is considered as a variable while α is considered as a constant, one has

$$\delta(0) = \delta(\beta - \alpha + p_2 - p_1)$$

Finally, one has $d\alpha = d\beta$ and $\delta(\alpha - \beta - p_2 + p_1) = \delta(\beta - \alpha + p_2 - p_1) = \delta(0)$

Considering $\int_{p_2}^{p_2+a} \delta(\alpha - \beta - p_2 + p_1)\hat{\gamma}_{\alpha\beta}[v](t)d\alpha = \begin{cases} 1 & if\ \alpha = \beta + p_2 - p_1 \\ 0 & otherwise \end{cases}$, equation (4.33) can be rewritten as

$$u(t) = \frac{1}{a}\int_{p_2}^{p_2+a}\int_{p_1}^{p_1+a} \delta(\beta - \alpha + p_2 - p_1)\hat{\gamma}_{\alpha,(\alpha-p_2+p_1)}[v](t)d\beta d\alpha$$

Finally, $$u(t) = k_p[v](t) = \frac{1}{a}\int_{p_2}^{p_2+a} \hat{\gamma}_{\alpha,(\alpha-p_2+p_1)}[v](t)d\alpha \qquad (4.34)$$

Equation (4.34) is the Preisach representation of the KP model. To verify that the equation can describe all the branches of the KP model, let the input $v(t)$ increase from p_2 to p, where $p \in (p_2, p_2 + a)$. By substituting $v(t) = p$ into equation (4.34), one obtains

$$k_p[v](t) = \frac{1}{a}\int_{p_2}^{p_2+a} \hat{\gamma}_{\alpha,(\alpha-p_2+p_1)}(p)d\alpha$$

All points on the support segment CD above the line $\alpha = \psi$ have $\hat{\gamma}_{\alpha,(\alpha-p_2+p_1)}(\psi) = -1$ and other points on the support segment CD below the line $\alpha = \psi$ have $\hat{\gamma}_{\alpha,(\alpha-p_2+p_1)}(\psi) = +1$. According to equation (4.34), one has

$$u(t) = k_p(t) = \frac{1}{a}(\int_{p_2}^{\psi} d\alpha - \int_{\psi}^{p_2+a} d\alpha)$$

$$= \frac{1}{a}[(\psi - p_2) - (p_2 + a - \psi)]$$

$$= -1 + \frac{2}{a}(\psi - p_2), \quad \psi \in (p_2, p_2 + a) \tag{4.35}$$

From Equation (4.35), one can see that equation (4.34) indeed describes the KP model when input $v(t) = \psi$ increases between p_2 and $p_2 + a$.

If it is assumed that the input decreases to $v(t) = \beta = \psi - p_2 + p_1$ (point F') in Fig.4.10, the support segment CD in Fig. 4.11 is still divided into two short segments DD' and $D'C$. This means that the output value of the KP kernel remains as

$$u(t) = k_p(t) = -1 + \frac{2}{a}(\psi - p_2), \ \beta \in (\psi, \psi - p_2 + p_1) \tag{4.36}$$

After decreasing to point F' in Fig.4.10, let the input keep decreasing to point B' where $v(t) = \beta = \psi'$. Please note that $\psi' \in (\psi - p_2 + p_1, p_1)$. The dividing point will slip down along the support segment DC in Fig.4.11 from point D' towards point B'. According to equation (4.34), the output of the KP kernel can be calculated as

$$u(t)=k_p(t)=\frac{1}{a}(\int_{p_2}^{\psi\ '+(p_2-p_1)}d\alpha-\int_{\psi\ '+(p_2-p_1)'}^{p_2+a}\ d\alpha)$$

$$=\frac{1}{a}((\psi\ '+(p_2-p_1)-p_2-(p_2+a)+\psi\ '+(p_2-p_1)$$

$$=-1+\frac{2}{a}(\psi\ '-p_1),\quad \psi\ '\in(\psi-p_2+p_1,p_1) \tag{4.37}$$

From equation (4.37), one can see that equation (4.34) indeed describes the descending curve of the KP element. Finally, equations (4.35) ~ (4.37) prove that the KP hysteresis model can be described by the Preisach model in integral form of relays as given by equation (4.34). From Fig.4.11, it is clear that by varying p_1, p_2 and α the support segment of the KP kernel in the representation of The Preisach model can be freely moved around in the half-plane $v^+ \geq p_2 \geq p_1 \geq v^-$. This fact and equation (4.34) suggest that the equivalent representation for the Preisach model can be obtained as a superposition (parallel connection) of infinite number of KP hysteresis elements. They also prove that the KP model possesses the *wiping-out* and *congruent* properties.

Even though the Preisach representation in integral form of weighted relays is more elementary in terms of its operators, the KP kernel provides more information of the nonlinearity than the relay because every KP kernel can be expressed as an integration of infinite weighted relays. The KP model has the memory effect to record all previous extremes of the process. This provides KP model to use less elements to calculate more complicated hysteresis, and it is not necessary to find the interface line of the process in the triangular Preisach plane.

4.4 Linearly Parameterized KP Model

Although the KP hysteresis model can be described in integral form as in equation (4.1), there are still some difficulties to directly calculate the hysteresis output. First, it is very difficult to formulate a suitable weighting function $\mu(p_1, p_2)$ which is double-integrable over the Preisach plane *P*; second, it is impossible to describe the outputs of the kernel $k_p[v, \xi_p](t)$ as continuous functions due to the nonlinearity of the kernel. Thus, the KP hysteresis model should be implemented by numerical method.

To numerically implement the KP model, one can transform (4.1) into linearly parameterized form by dividing the Preisach plane *P* into a mesh. For an acceptable rough approximation, the Preisach plane *P* can be uniformly divided by l horizontal lines and l vertical lines into $N = (l+2)(l+1)/2$ small cells (see Fig.4.2) with coordinates (v_i, v_j) of their lower-left nodes expressed by

$$\begin{cases} v_i = v^- + (i-1)\Delta v' \\ v_j = v^- + (j-1)\Delta v' \end{cases} \tag{4.38}$$

where
$$i \geq j, \quad i, j = 1, 2, \ldots, l+1 \tag{4.39}$$

and
$$\Delta v' = \frac{v^+ - v^-}{l+1} \tag{4.40}$$

is the interval of the divided input $v(t)$, which is selected as

$$\Delta v' = a \tag{4.41}$$

so that the kernels overlap cell by cell to create the smoothest hysteresis curves (see Fig.4.13).

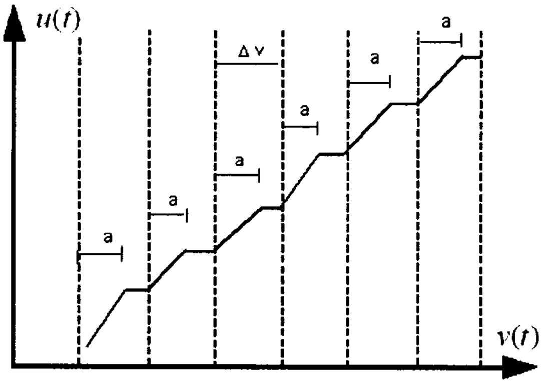


Figure 4.13 Possible Curve form KP Model when $a < \Delta v'$

After the dividing of the Preisach plane P, the total contribution of kernels of each cell to the *KP* hysteresis model (4.1) can be lumped together as the effect of the kernel associated with the lower-left node of the cell. One has

$$k_{p_{ij}}(t)\cdot\rho_{ij} = \int_{\text{the ijth cell}} k_p(t)\cdot\mu_p dp \tag{4.42}$$

where $\rho_{p_{ij}}$ is called the lumped density of a cell to its lower-left node (v_i,v_j). As l is selected large enough, the dividing of the Preisach plane P becomes very fine, the cells become very small, and the output values $k_p(t)$ of all kernels on each cell can be considered as identical to the values $k_{p_{ij}}(t)$ of the lower-left node of the cell. Thus (4.42) is rewritten as

$$k_{p_{ij}}(t)\cdot\rho_{ij} = \int_{\text{the ijth cell}} k_p(t)\cdot\mu_p dp \approx k_{p_{ij}}(t)\cdot \int_{\text{the ijth cell}} \mu_p dp \tag{4.43}$$

Finally,
$$\rho_{ij} \approx \int_{\text{the ijth cell}} \mu_p dp \tag{4.44}$$

Define
$$c \overset{\Delta}{=} \int_P \mu(p)dp = \sum_i^{l+1}\sum_{j=1}^{i} \int_{\text{the ijth cell}} \mu_p dp \tag{4.45}$$

and $$\bar{\mu}_{ij} = \mu_{p_{ij}} / (\sum_{i}^{l+1} \sum_{j=1}^{i} \mu_{p_{ij}}) \tag{4.46}$$

where $\bar{\mu}_{ij}$ is the normalized densities of all lower-left nodes on the Preisach plane P, so

that $$\sum_{i}^{l+1} \sum_{j=1}^{i} \bar{\mu}_{ij} = 1 . \tag{4.47}$$

Since $$\sum_{i}^{l+1} \sum_{j=1}^{i} c\bar{\mu}_{ij} = c \sum_{i}^{l+1} \sum_{j=1}^{i} \bar{\mu}_{ij}$$

substituting equation (4.47) into the above equation gives

$$\sum_{i}^{l+1} \sum_{j=1}^{i} c\bar{\mu}_{ij} = c .$$

Considering the definition (4.45) gives

$$\sum_{i}^{l+1} \sum_{j=1}^{i} c\bar{\mu}_{ij} = \sum_{i}^{l+1} \sum_{j=1}^{i} \int_{the\ ijth\ cell} \mu_p dp \tag{4.48}$$

Substituting equation (4.44) into (4.48) yields

$$\sum_{i}^{l+1} \sum_{j=1}^{i} c\bar{\mu}_{ij} = \sum_{i}^{l+1} \sum_{j=1}^{i} \rho_{ij} ,$$

which implies

$$\rho_{ij} \approx c\bar{\mu}_{ij} \tag{4.49}$$

Indeed, if the number of dividing lines $l \to \infty$, then each cell degenerates as its lower-left node, and one has $\rho_{ij} \approx c\bar{\mu}_{ij} \to \mu_{p_{ij}}$.

For an acceptable modeling precision, a selected large enough l results in equation (4.42), and then the KP hysteresis model (4.1) can be linearly parameterized as

$$u(t) = \sum_{i=1}^{l+1} \sum_{j=1}^{i} \int_{the\ ijth\ cell} k_p(t) \cdot \mu_p dp = \sum_{i=1}^{l+1} \sum_{j=1}^{i} k_{p_{ij}}(t) \cdot \rho_{ij} + d(v(t)) = \Gamma^T K + d(v(t)) \tag{4.50}$$

where $$K^T = [k_{p_{11}}, k_{p_{21}}, k_{p_{22}}, k_{p_{31}}, k_{p_{32}}, \ldots, k_{p_{(l+1)(l+1)}}] \tag{4.51}$$

and $$\Gamma^T = [\rho_{11}, \rho_{21}, \rho_{22}, \rho_{31}, \rho_{32}, \ldots, \rho_{(l+1)(l+1)}] \tag{4.52}$$

are N dimension vectors respectively, and ρ_{ij} are calculated by (4.45), (4.46) and (4.49). The term $d(v(t))$ in (4.50) is modeling error by the linearly parameterization of the model. If the number of dividing lines increases to extremely large value, the term $d(v(t))$ can be considered as zero.

Alternatively, the linearly parameterized KP hysteresis model (4.50) can also be expressed using compensator m_p as

$$\begin{aligned} u(t) &= cv(t) - \sum_{i=1}^{l+1}\sum_{j=1}^{i} \int_{the\ ijth\ cell} m_p(t)\cdot \mu_p dp \\ &= cv(t) - \sum_{i=1}^{l+1}\sum_{j=1}^{i} m_{p_{ij}}(t)\cdot \rho_{p_i p_j} + d(v(t)) \\ &= cv(t) - \Gamma^{T} M + d(v(t)) \end{aligned} \tag{4.53}$$

where $$M^{T} = [m_{p_{11}}, m_{p_{21}}, m_{p_{22}}, m_{p_{31}}, m_{p_{32}} \ldots, m_{p_{(l+1)(l+1)}}] \tag{4.54}$$

is a N dimension vector consisting of output values of all compensators associated with all lower-left nodes of all small cells.

Equations (4.50) and (4.53) are called the linearly parameterized *KP* hysteresis models. Although they can be modeled respectively, these two models are dependent and compensative to each other since they have the same parameter vector Γ^T (4.52). The relationship between two models is expressed as

$$u(t) = cv(t) - \Gamma^T M + d(v(t)) = \Gamma^T K + d(v(t)). \tag{4.55}$$

In (4.53), the term $cv(t)$ is a proportional part of the input $v(t)$, and it means that the hysteretic actuator can be considered as a linear transducer with an input-output error of hysteresis directly described by the term $\Gamma^T M$. This linearly parameterized representation by equation (4.53) is critical for nonlinear robust adaptive control design because the linear term $cv(t)$ can be used to design the controller while the term $\Gamma^T M$ can be considered as dynamics of the plant. Namely, the hysteretic actuator can be treated as a linear transducer while the hysteretic error $\Gamma^T M$ can be considered as plant dynamics.

4.5 Implementation of the Linearly Parameterized KP Model

Governed by the linearly parameterized KP model by equations (4.50~4.52), a hysteresis system is completely characterized by its positive/negative saturation states $(v^+,u^+)/(v^-,u^-)$, number l of dividing lines, and the density distribution μ_{ij} over the Preisach plane P when the system is subjected to a certain input signal $v(t)$. In (4.51), the kernels are characterized by $p_{ij} = (p_i, p_j)$ pair and rise-time a, and these parameters can be calculated by using equations (4.39) and (4.41). The positive/negative saturation states $(v^+,u^+)/(v^-,u^-)$ can be observed from experiment results. The number of l dividing lines, which determines the precision of approximation, can be chosen upon the requirement of modeling precision. The density distribution μ_{ij} which is the key parameter to determine the shape of the hysteresis loop can only be identified by matching experiment data using the method to be discussed in forthcoming section.

Based on the definition of a kernel function of the KP model (equations (4.3~4.5), an s-function, "***Kernel***", in *MatLab* to calculate the outputs of each kernel as it is subject to an input signal $v(t)$ is written and listed in Appendix 3. The summation of all weighted outputs of all "***Kernels***" is calculated by a custom defined s-function "***Hysteresis***", which is also listed in Appendix 3.

In the s-function "***Kernel***", the input $v(t)$ is sampled into a series of input values forming a V vector with a sampling time $t_s = 0.01\sec$. At every sampling moment there is a corresponding V vector consisting of input values as its elements from the beginning of the process to the moment. As the time marches forward, the V vector is updated, and every new updated V vector is used to calculate the outputs of all kernels. Furthermore, all outputs of kernels are multiplied by the corresponding densities, and then are summed together to generate the output value $u(t)$ of the system at the sampling moment. Thus, with the same size as the V vector, an output vector U which consists of the calculated outputs $u(t)$ at every sampling time is formed. As the time continues, the applied inputs $v(t)$ and the calculated outputs $u(t)$ of the system form the hysteresis loops.

One fact need to be mentioned is that the V vector only has one element at the beginning of the process $(t = 0)$. In order to determine whether the sign of input variation, $sign\ \ (\dot{v}(t))$, is switched at the initial time $(t = 0)$, two elements equaling to $v(t = 0)$ are added into the V vector. This makes the V vector to have three elements with the same value $v(t = 0)$ at the first sampling moment $(t = 0)$. Meanwhile, two elements $(-1,-1)$ are also added at the head of output vector U. This ensures the input V vector and the output U vector have the same size.

In order to conduct simulation by running this s-function "***Hysteresis***" to calculate the output of a hysteresis, the density distribution μ_{ij} of the hysteresis must be first offered to the s-function "***Hysteresis***".

For simplicity of explanation, the lumped densities ρ_{ij} in (4.50) and (4.52) of a particular hysteresis system are directly assumed as

$$\rho_{ij} = \frac{\mu(p_i, p_j)}{\sum_{i=1}^{l+1}\sum_{j=1}^{i}\mu(p_i, p_j)} \tag{4.56}$$

where $$\mu(p_i, p_j) = c_1 \exp(c_2((p_i - \bar{p}_1)^2 + (p_j - \bar{p}_2)^2)) \tag{4.57}$$

with constants $$c_1 = \frac{1}{2\pi\sigma^2},\ c_2 = -\frac{1}{2\sigma^2},\ \sigma = 0.6 \text{ and } \bar{p}_1 = \bar{p}_2 = 0, \tag{4.58}$$

and $$i \geq j \quad i, j = 1,2,\ldots,l+1.$$

Equation (4.57) expresses a bi-variant normal density function, where the constant σ specifies the spread of the distribution, and $\bar{p}_1$ and $\bar{p}_2$ are the center of the distribution. Equation (4.56) is used to calculate the lumped densities $[\rho_{ij}]_{(l+1)\times(l+1)}$ by normalizing the density values $\mu(p_i, p_j)$ so that

$$\sum_{i=1}^{l+1}\sum_{j=1}^{i}\rho_{ij} = c = 1 \tag{4.59}$$

Here the normalized densities given by equations (4.56) and (4.57) are used to perform the simulation. The normalized densities ρ_{ij} with $l = 4, \sigma = 0.6$ for $v^- = -2$ and $v^+ = 2$ are listed in table 4.1, and the normalized density distribution ρ_{ij} over the P plane with $l = 4, \sigma = 0.6$ for the same hysteresis region $v \in [v^-, v^+] = [-2, +2]$ is shown in Fig.4.14.

Table 4.1 Normalized densities with $l = 4, \sigma = 0.6$ and $v \in [v^-, v^+] = [-2, +2]$

ρ_{ij}	β j=1~5				
α i=1~5	0.0002	0.0076	0.0448	0.0448	0.0076
	0.0013	0.0448	0.2650	0.2650	0
	0.0013	0.0448	0.2650	0	0
	0.0002	0.0076	0	0	0
	0.0000	0	0	0	0

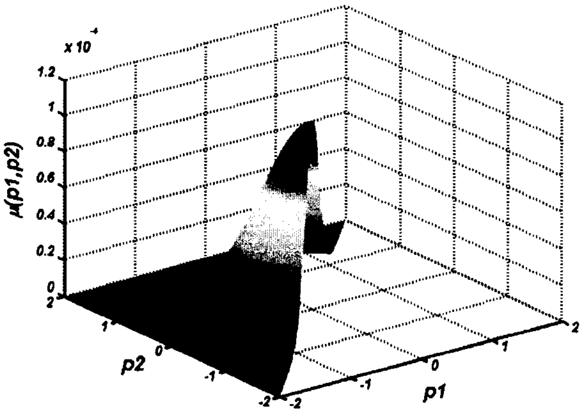


Fig. 4.14 Normalized density distribution over the P plane with $l = 99, \sigma = 0.6$ and $v \in [v^-, v^+] = [-2, +2]$

In the table 4.1, the summation of all normalized weighting factors indeed equals $+1$. This leads to the saturation states u^+/u^- of the hysteresis to equal $+1/-1$ when input $v(t)$ reaches the saturation points v^+/v^- respectively. This fact can be proven by calculating the outputs at saturation states by equations (4.50) and (4.59) as

$$u(t) = \sum_{i=1}^{l+1}\sum_{j=1}^{i} k_{p_{ij}} \rho_{ij} = \sum_{i=1}^{l+1}\sum_{j=1}^{i} (\pm 1)\rho_{ij} = (\pm 1)\sum_{i=1}^{l+1}\sum_{j=1}^{i} \rho_{ij} = \pm 1$$

Thus the normalization of the density distribution also normalizes the hysteresis loops.

The hysteresis system with the above density distribution (table 4.1) is first assumed to be subject to a sinusoidal input signal (see Fig.4.15) as

$$v(t) = -4\cos(\pi t / 20). \tag{4.60}$$

All responses of kernels defined by $l = 4$ and $v \in [v^-, v^+] = [-2, +2]$ to the input signal, $v(t) = -4\cos(\pi t / 20)$, are illustrated in Fig.4.16. These kernels do not have inner loops inside their major loops because all extremes of the input $v(t)$ exceed the saturated inputs (+2). Furthermore, the summation of weighted kernels forms a rough hysteresis with only a major loop (see Fig.4.17).

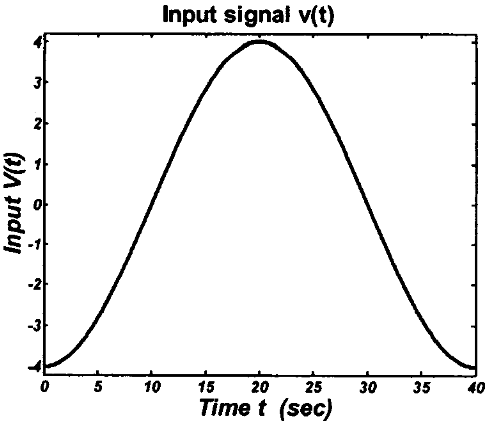


Fig.4.15 Input signal $v(t) = -4\cos(\pi t/20)$

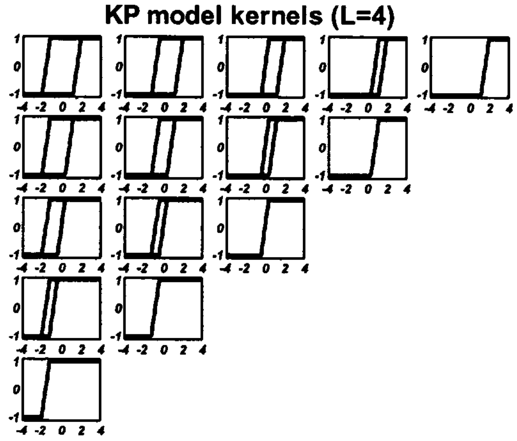


Fig.4.16 kernels in P plane: $v \in [v^-, v^+] = [-2,+2]$ $v(t) = -4\cos(\pi t/20)$

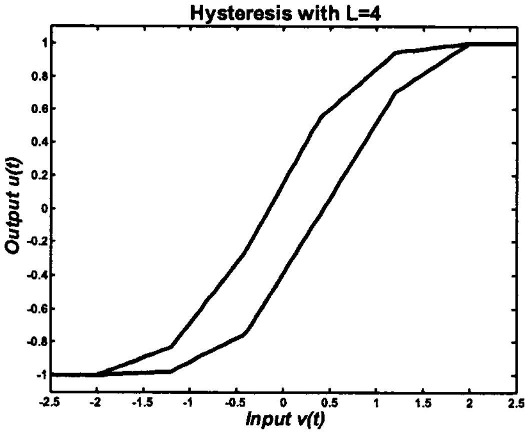


Fig.4.17 Hysteresis with $v(t) = -4\cos(\pi t/20)$, $l = 4, \sigma = 1.0$ and $v \in [v^-, v^+] = [-2,+2]$

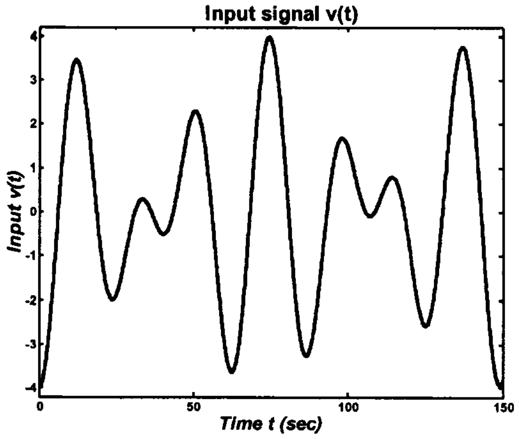


Fig.4.18 Input signal $v(t) = -2[(\cos(\pi t/15) + \cos(\sqrt{2}\pi t/15)]$

For the same hysteresis, if the inputs signal (see Fig.4.18) is selected as

$$v(t) = -2[(\cos(\pi t/15) + \cos(\sqrt{2}\pi t/15)], \tag{4.61}$$

simulation results are shown in Fig.4.19 and Fig.4.20. From Fig.4.19, one can see that every kernel have inner curves inside their major loops because the input signal has some local maximal and minimal which are inside the region of $[v^-, v^+]$. The sign switching of the input variation for every kernel causes its input-output trajectory to traverse inside its major loop. Therefore, the accumulation of all input-output trajectories of all kernels forms the hysteresis loops with both major loop and minor loops (see Fig.4.20).

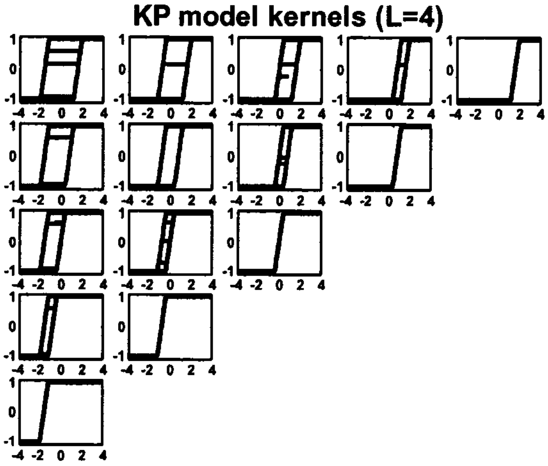


Fig.4.19 kernels in P plane: $v \in [v^-, v^+] = [-2,+2]$ $v(t) = -2$ $[(\cos(\pi t/15) + \cos(\sqrt{2}\pi t/15)]$

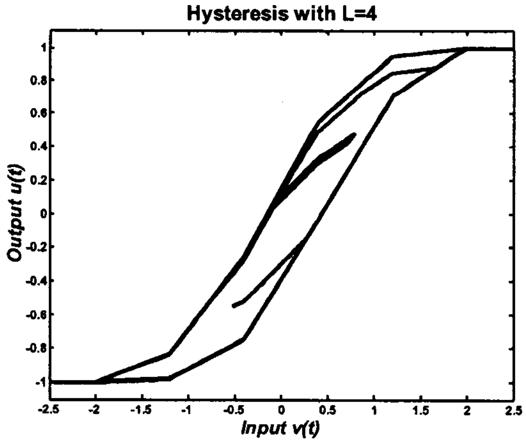


Fig.4.20 Hysteresis with $l = 4, \sigma = 1.0$ and $v \in [v^-, v^+] = [-2,+2]$ $v(t) = -2[(\cos(\pi t/15) + \cos(\sqrt{2}\pi t/15)]$

The above simulation results (Fig.4.17 and Fig.4.20) show that the KP model can conveniently describe the hysteresis phenomena including major loop and inner minor loops, but when the number of discretization lines l is chosen too small, the simulated result is very rough. As the number of l is chosen larger, the hysteresis plane P is discretized finer and the resulted hysteresis loops become smoother. For example, in the above two cases, when l is selected larger than 10, acceptable smooth modeling results can be achieved.

For a finer discretization of $l = 19$, Fig.4.21 and Fig4.22 illustrates two hysteresis nonlinearities with different density distributions in the same input region $v \in [v^-, v^+] = [-2,+2]$ when it is subjected to an input signal as shown in Fig.4.18. When a lumped density distribution is given by equation (4.56) with $\sigma = 0.6$, the hysteresis result is shown as in Fig.4.21; with parameter $\sigma = 1.0$ the modeled hysteresis loops are displayed as Fig.4.22, in which the loops are wider than the hysteresis loops in the Fig.4.21. These simulation results show that the density distribution of the KP model indeed influences the shape of the hysteresis loops.

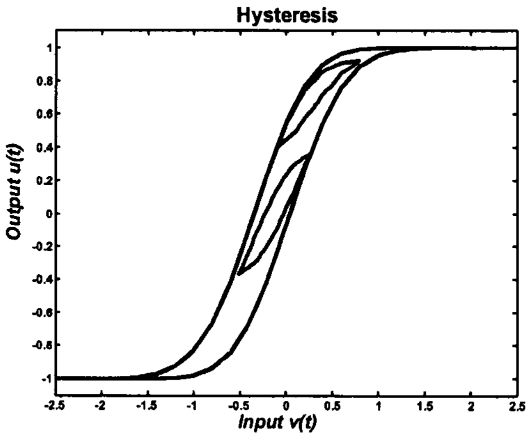


Fig.4.21 Hysteresis with $l = 19$, $\sigma = 0.6$ and $v(t) = -2[(\cos(\pi t/15) + \cos(\sqrt{2}\pi t/15)]$

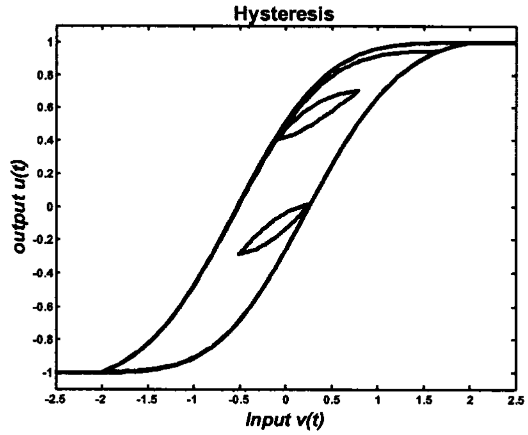


Fig.4.22 Hysteresis with $l = 19$, $\sigma = 1.0$ and $v(t) = -2[(\cos(\pi t/15) + \cos(\sqrt{2}\pi t/15)]$

4.6 Identification of Model Parameters by Least Squares Method

To model the input hysteresis of a system in the linearly parameterized KP representation, its parameters (density vector Γ) must be identified in advance by the least squares method by fitting the experimental data.

In the above equations (4.50) and (4.55), for every input $v(t)$, in terms of the definition of kernels one can get $K = [k_{p_1}, k_{p_2}, ..., k_{p_i}, ...k_{p_N}]^T$ with $\Gamma^T = [\rho_1, \rho_2, ..., \rho_i, ...\rho_N]$ which is a parameter vector to be found. For an input signal $v(t)$, the number of N outputs $u^*(t)$ of a real hysteresis actuator can be measured. It is assumed that vector $u^*(t)$ can be exactly generated by the linearly parameterized KP model using real parameters $\Gamma^* = [\ \rho_1^*, \rho_2^*, ..., \rho_i^*, ... \rho_N^*\ ]^T$ as

$$u^*(t) = \Gamma^{*\ T} K \tag{4.62}$$

However, since the exact parameters Γ^* is unknown, any approximation to the parameter vector Γ over the Preisach plane P will lead to different output of the hysteresis subjected to the same input signal $v(t)$. One can use the least squares method to minimize the square error of hysteresis output to find acceptable densities, given as:

$$\min \sum_{i=1}^{N} | u_i - u_i^* |^2 = \min \sum_{i=1}^{N} | (\Gamma_i^T - \Gamma_i^{*\ T}) K |^2 \tag{4.63}$$

This calculation can be implemented by calling an m. function "*lsqnonneg* $(K, u^*(t))$" where K *and* $u^*(t)$ are vectors consisting of N number of elements.

The off-line identified parameters can be used to predict the range of KP model parameters for adaptive control design in next section.

The following is an example of using the least squares method to identify the density distribution of the hysteresis. For a hysteresis with a hysteretic input region as $v \in [v^-, v^+] = [-2, +2]$, it is assumed that the real lumped density distribution $\Gamma = [\rho_1, \rho_2, ..., \rho_N]^T$ is governed by equation (4.56). If the Preisach plane P is uniformly divided by $l = 9$ horizontal lines and $l = 9$ vertical lines, thus $N = (l+2)(l+1)/2 = 55$ left-lower nodes associated with 55 small cells. The lumped density distribution Γ is a 55×1 vector, which can be rearranged as a 10×10 matrix over the Preisach plane P in a mechanism described by following program.

```
$N=(l+2)(l+1)/2;$
$\Gamma\_vector \ =[\rho_i]_{1\times N}$          % enter a density vector with demension of $1\times N$
$\Gamma\_matrix \ =[0]_{(l+1)\times(l+1)}$        % define a $(l+1)\times(l+1)$ matrix
for $i=1:l+1$
    for $j=1:i$
    $m=i*(i-1)/2+j$
    $\rho_{ij}=\rho_m$                            % evalue element of the $(l+1)\times(l+1)$ matrix
    end
end
$\Gamma\_matrix \ =[\rho_{ij}]_{(l+1)\times(l+1)}$
```

The rearranged 10×10 density matrix is shown in table 4.2. The identification result of densities is also rearranged into a 10×10 matrix $\hat{\Gamma}=[\hat{\rho}_{ij}]_{10\times10}$ in the same mechanism and the matrix is shown in table 4.3. The corresponding identification error matrix is calculated and shown in table 4.4.

Table 4.2 Given normalized densities with $l=9,\sigma=1.0$ and $v\in[v^-,v^+]=[-2,+2]$

ρ_{ij}	β j=1~10 ⟶									
α i=1~10 ⟶	0.0019	0.0039	0.0068	0.0101	0.0128	0.0139	0.0128	0.0101	0.0068	0.0039
	0.0033	0.0068	0.0119	0.0177	0.0225	0.0244	0.0225	0.0177	0.0119	0
	0.0049	0.0101	0.0177	0.0264	0.0335	0.0363	0.0335	0.0264	0	0
	0.0063	0.0128	0.0225	0.0335	0.0426	0.0462	0.0426	0	0	0
	0.0068	0.0139	0.0244	0.0363	0.0462	0.0500	0	0	0	0
	0.0063	0.0128	0.0225	0.0335	0.0426	0	0	0	0	0
	0.0049	0.0101	0.0177	0.0264	0	0	0	0	0	0
	0.0033	0.0068	0.0119	0	0	0	0	0	0	0
	0.0019	0.0039	0	0	0	0	0	0	0	0
	0.0009	0	0	0	0	0	0	0	0	0

Table 4.3 Calculated normalized densities with $l=9,\sigma=1.0$ and $v\in[v^-,v^+]=[-2,+2]$

$\hat{\rho}_{ij}$	β j=1~10 ⟶									
α i=1~10 ⟶	0.0019	0	0	0.0475	0	0	0.0128	0.0101	0.0068	0.0039
	0.0033	0	0.0419	0	0.0018	0	0.0355	0.0441	0.0119	0
	0.0049	0	0	0	0	0.1208	0.0632	0	0	0
	0.0063	0.0141	0	0.0766	0.1097	0	0	0	0	0
	0.0068	0	0.0746	0	0.0462	0.0500	0	0	0	0
	0.0063	0.0353	0	0.0335	0.0426	0	0	0	0	0
	0.0049	0.0278	0	0.0264	0	0	0	0	0	0
	0.0033	0	0.0186	0	0	0	0	0	0	0
	0.0019	0.0039	0	0	0	0	0	0	0	0
	0.0009	0	0	0	0	0	0	0	0	0

Table 4.4. Error of normalized densities with $l = 9, \sigma = 1.0$ and $v \in [v^-, v^+] = [-2, +2]$

$\tilde{\rho}_{ij}$	β j=1~10 →									
α i=1~10 →	0.0000	0.0039	0.0068	-0.0374	0.0128	0.0139	-0.0000	0.0000	-0.0000	-0.0000
	-0.0000	0.0068	-0.0301	0.0177	0.0207	0.0244	-0.0130	-0.0264	-0.0000	0
	0.0000	0.0101	0.0177	0.0264	0.0335	-0.0845	-0.0296	0.0264	0	0
	0.0000	-0.0013	0.0225	-0.0430	-0.0670	0.0462	0.0426	0	0	0
	-0.0000	0.0139	-0.0502	0.0363	0.0000	-0.0000	0	0	0	0
	0.0000	-0.0225	0.0225	-0.0000	-0.0000	0	0	0	0	0
	-0.0000	-0.0177	0.0177	0.0000	0	0	0	0	0	0
	0.0000	0.0068	-0.0068	0	0	0	0	0	0	0
	-0.0000	0.0000	0	0	0	0	0	0	0	0
	-0.0000	0	0	0	0	0	0	0	0	0

Using these calculated parameters (lumped densities), the outputs corresponding to the input signals (4.61) are obtained and the hysteresis loop is shown as the curve "B" in Fig.4.23 and Fig.4.24 when l is chosen as $l = 9$ and $l = 19$ respectively. Using the original densities listed in table4.2 to simulate the system, the result is also shown as the curve "A" in Fig.4.23 and Fig.4.24. By comparing the two figures, it can be seen that the error approaches zero when l is chosen as an acceptable large number.

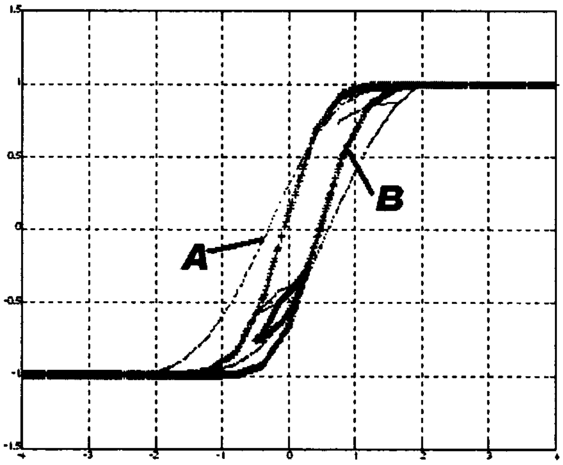


Fig.4.23 Hysteresis calculated by given density distribution vs identified densities with $l = 9$ and $v(t) = -2[(\cos(\pi t/15) + \cos(\sqrt{2}\pi t/15)]$

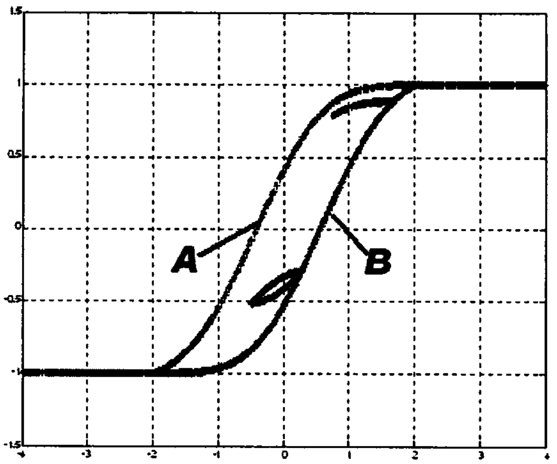


Fig.4.24 Hysteresis calculated by given density distribution vs identified densities with $l = 19$ and $v(t) = -2[(\cos(\pi t/15) + \cos(\sqrt{2}\pi t/15)]$

4.7 Inverse Hysteresis Operator and Open Loop Compensation

The KP model is to find the output $u(t)$ of the hysteresis nonlinearity with a given input $v(t)$ applied to the hysteretic actuator at every moment t. The goal of modeling the hysteresis with the KP model is to cancel or reduce the hysteresis nonlinearity. Compensation through an inverse operator of the KP model is one effective approach.

The objective of the KP inverse operator model is to find the input $v(t)$ for a given desired output $u_d(t)$. The goal is to predict the input $v(t)$ at every moment of time t with knowledge of previous inputs, outputs before the moment t, and the desired output $u_d(t)$ at the moment t. Thus, the operator is the application of the KP model, in which the input vector consists of previous input values and the estimated input value at the moment of concern. The input value at the moment of concern is estimated by adding approaching steps ($ndv, n = 1,2,...$) to the last known input value. The output $u(t)$ is calculated by using every estimated input value and $u(t)$ approaches the desired value $u_d(t)$ at the moment t. As the calculated output value $u(t)$ becomes greater than the desired output $u_d(t)$ at the moment t, the adding of the approaching steps stops. The last two calculated output values will be used to interpolate an optimal output value for the moment t, as given by equations (4.64) and (4.65). The numerical method is illustrated in Fig.4.25 and Fig.4.26 and the algorithm to approximate the KP inverse model is as follows:

1) Set input $v_2 = v_{d\,present}$ and output $u_2 = u_{d\,present}$

2) If $u_d > u_{d\,present}$ (u increase as v increase, see Fig.4.25)

2.1) Set $v_1 = v_2$ and $u_1 = u_2$

2.2) $v_2 = v_1 + dv$ and $u_2 = H(v_2)$
(Increase input v by step dv and calculate output u_d by hysteresis operator)

2.3) If $u_2 < u_d$, back to step 2.1
(Not yet arrive or far away from the output u_d, continue to increase the input v)

2.4) As $u_2 > u_d$ has just been satisfied, conduct the following interpolation to find v.

$$\hat{v} = v_1 + dv\frac{u_d - u_1}{u_2 - u_1} \tag{4.64}$$

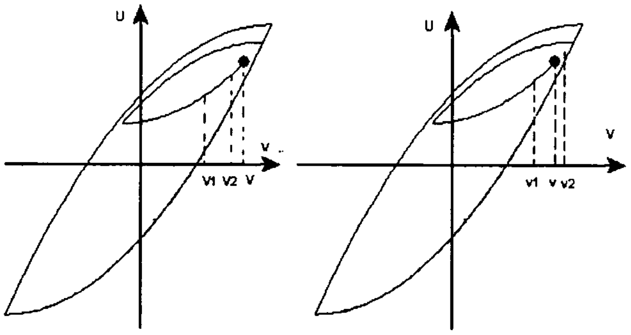


Fig.4.25 Desired u_d changes to increase

3) If $u_d < u_{d\,present}$ (the output u_d changes to decrease, see Fig.4.26)

3.1) Set $v_1 = v_2$ and $u_1 = u_2$

3.2) $v_2 = v_1 - dv$ and $u_2 = H(v_2)$

(Decrease the input v by step dv and calculate the output u_d by hysteresis operator)

3.3) If $u_2 > u$, back to step3.1

(The output u_2 has not arrived the value u_d , the input v must be decreased again)

3.4) As $u_2 < u_d$ has just been satisfied, conduct the following interpolation to find v.

$$\hat{v} = v_1 - dv\frac{u_d - u_1}{u_2 - u_1} \tag{4.65}$$

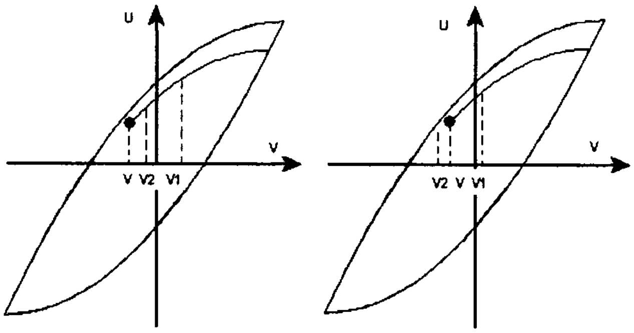


Fig.4.26 Desired u_d changes to decrease

It can be seen that if $v_1 \le \hat{v} \le v_2$, $v_1 \le v^* \le v_2$ and $dv = | v_2 - v_1 |$, then $| v^* - \hat{v} | \le dv$. In fact, the estimated $\hat{v}$ will generally be much closer to the exact inverse v^* than dv. In addition, as $dv \to 0$, one has $\hat{v} \to v^*$. However, dv and the number l of dividing lines of $H(v)$ increase the computational cost significantly in this inverse method while $\frac{dv}{V_{\max} - V_{\min}} \le 1\%$ and $l \ge 19$.

Based on the results presented above, it is assumed that the KP model captures the hysteresis characteristics accurately and comprehensively. Consequently, it is possible to pursue model-based compensation of hysteresis nonlinearity using an open-loop compensation strategy depicted in Fig.4.27. The desired output from the hysteresis system is the input to the inverse KP model. The inverse model predicts the input $v(t)$ which is applied to the hysteresis to achieve the desired output $u_d(t)$, and thus compensates the hysteresis. If the KP model cannot capture the hysteresis characteristics accurately and comprehensively, there will be a deviation between the desired output $u_d(t)$ and the generated output $u(t)$. This error can be used as a measure of success for the open-loop compensation strategy.

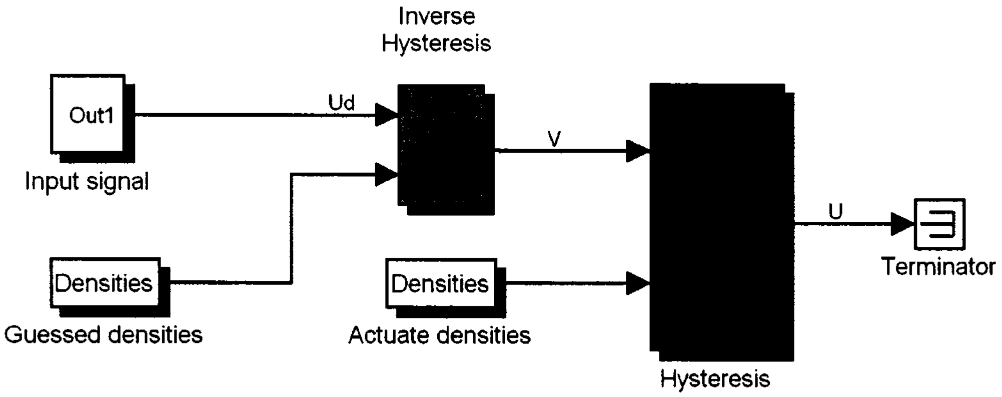


Fig.4.27 Open-loop compensation to hysteresis of actuator described in KP operators

According to the approximate inversion algorithm, a *Simulink* diagram (the inverse KP model) is designed as shown in Fig.4.28, where the "while" subsystem acts as a "while loop" which represents the approaching of the final value of $v(t)$ towards the desired $u_d(t)$.

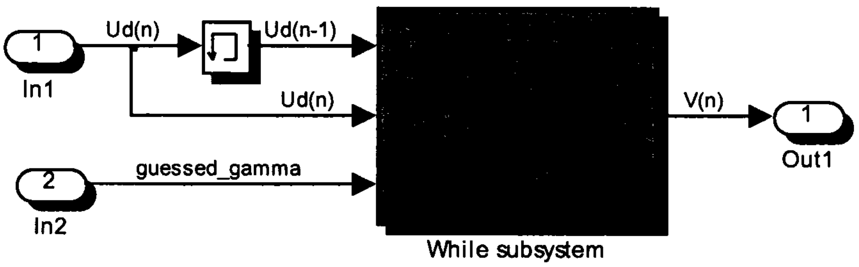


Fig.4.28 The inverse hysteresis operator based on the KP model

To judge how the input $u_d(t)$ varies, a "memory" block is used to store the previous sampled input value $u_d(n-1)$. Comparing the current sampled input value $u_d(n)$ to the previous $u_d(n-1)$ generates the judgment conditions as

$$\begin{cases} channel\_state = 1 & if\ u_d(n-1) < u_d(n) \\ channel\_state = 2 & if\ u_d(n-1) = u_d(n) \\ channel\_state = 3 & if\ u_d(n-1) > u_d(n) \end{cases} \tag{4.66}$$

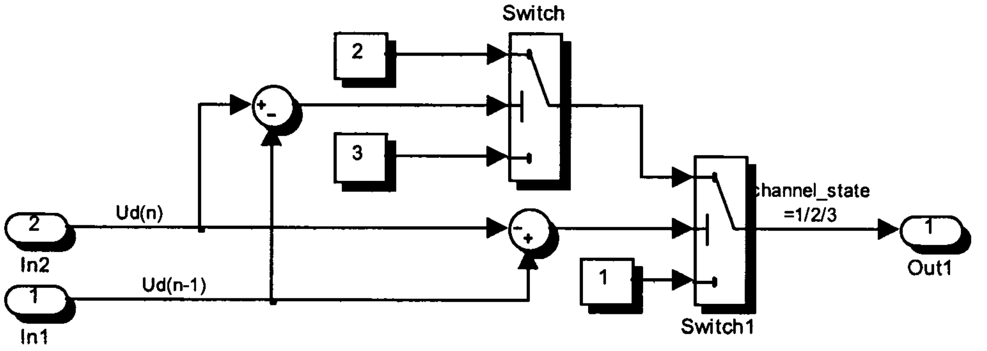


Fig.4.29 Judgment for variation of input

A *Simulink* diagram (see Fig.4.29) is constructed to work as a subsystem, *Judgment for variation of input* $u_d(t)$, in a *while subsystem* (see Fig.4.30) of the real time compensation configuration. The channel_state =1 represents an increasing desired input $u_d(t)$, i.e., the desired output of the actuator, passing to the inverse hysteresis; the channel_state =3 indicates a decreasing $u_d(t)$ entering to the inverse hysteresis; and the channel_state =2 implies that input $u_d(t)$ arrives at u^+/u^- which is the output of the positive/negative saturation state. The value of the variable "channel_state" is used to decide the open channel of the block "Switch1" in the "While" subsystem to pass the signal from the input gates to its output gate.

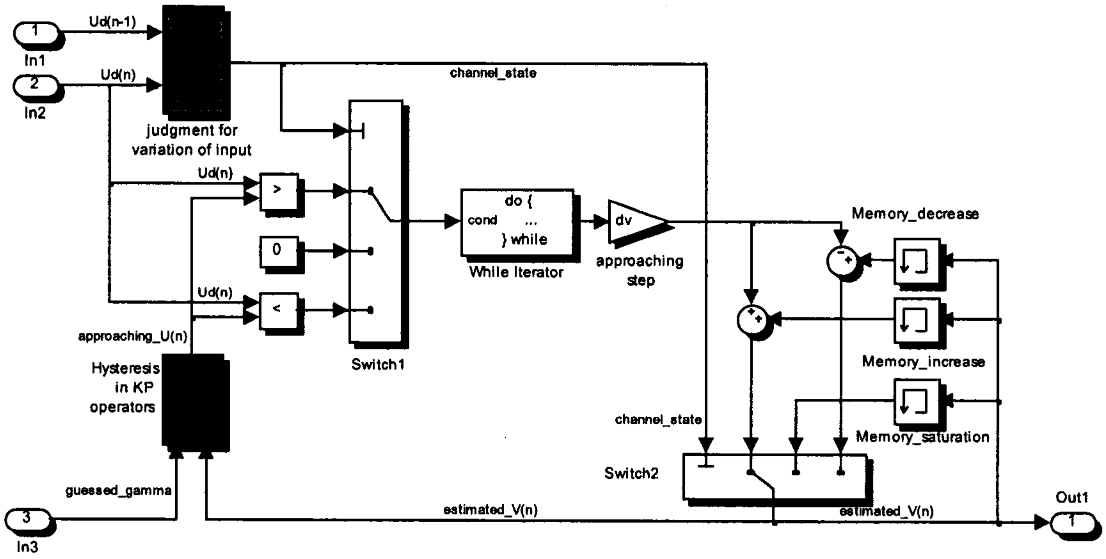


Fig.4.30 While subsystem

With reference to Fig.4.30, as the desired input to the inverse hysteresis $u_d(n)$ is increasing, channel_state =1, the branch $u_d(n) > u_d(n-1)$ is allowed to pass through switch1 as the input to the do while loop, *while iterator.* Meanwhile, the previously calculated candidate $v(n)$ passes from *the memory_increase* and increases by an approaching step dv to update as a new candidate of $v(n)$, *estimated_v(n),* which enters to the subsystem *"Hysteresis in* KP *operator"* to calculate a new *approaching_u(n).* The *while_iterator* loop continuously increases by 1 so that the candidate value of $v(n)$ increases by an approaching step dv while the logical operation " $u_d(n) > approaching\_u(n)$ " still has a logical value "1", which means the $approaching\_u(n)$ *is still less than the desired input* $u_d(n)$. Similarly, when the desired input to the inverse hysteresis $u_d(n)$ changes to decrease, both switch 1 and switch 2 opens and switch number 3 closes to find the closest $v(n)$. When the desired input to the inverse hysteresis $u_d(n)$ arrives positive or negative saturations, $u_d(n) = u_d(n-1)$, then the *while_iterator* loop is skipped, and the previous calculated candidate $v(n)$ passes from *the memory_saturation* as the closest $v(n)$.

If the desired output of the hysteretic actuator or the input of the open-loop compensation system is given by the equation

$$u_d(t) = -2[(\cos(\pi t/15) + \cos(\sqrt{2}\pi t/15)], \tag{4.67}$$

which is shown in Fig.4.31a, then the predicted input to the hysteresis system is calculated by using the inverse KP operator as shown in Fig.4.31b, and the real output of the hysteresis system is calculated by the KP model as shown in Fig.4.31c. The compensated result is shown as Fig.4.31f. Fig.4.31d and Fig.4.31e show respectively the inverse hysteresis and hysteresis loops.

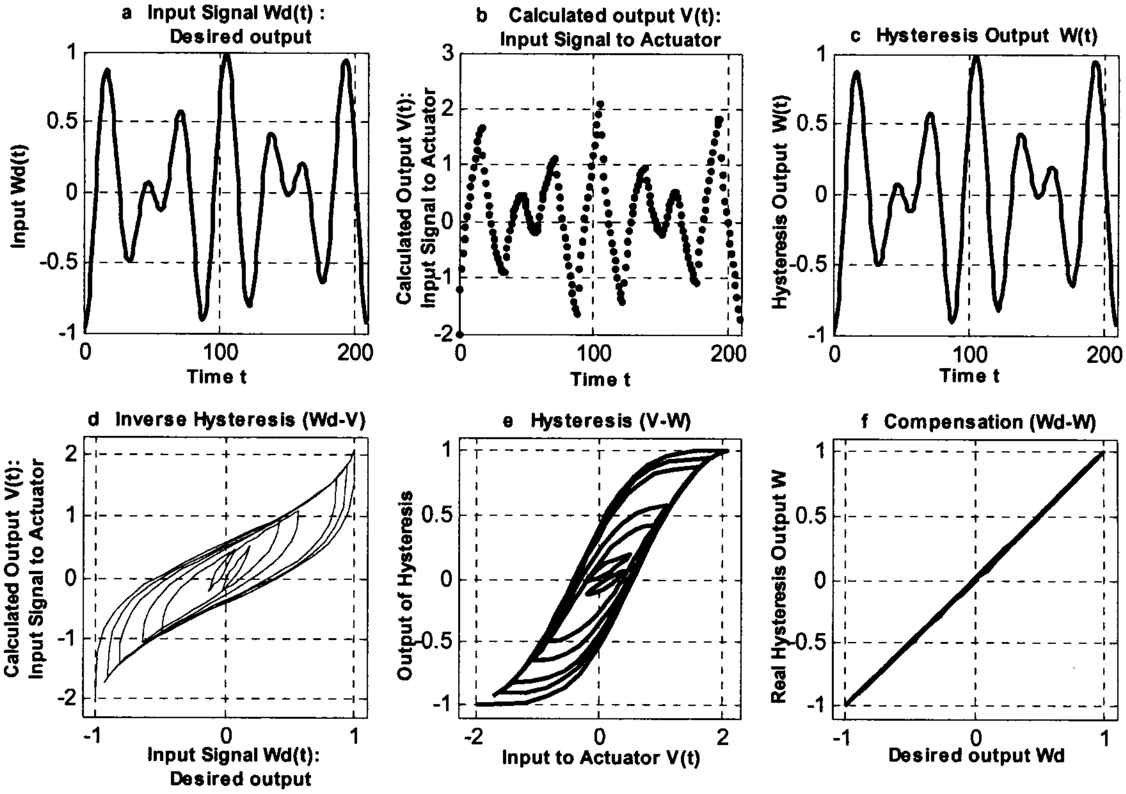


Fig.4.31 Open loop compensation by inverse KP operator

If the inverse hysteresis is a rough approximation, an error $e(t) = u(t) - u_d(t)$ between the desired output $u_d(t)$ and the real output $u(t)$ will still remain in the system after the compensation. For example, assume that an actuator hysteresis system is exactly represented by the parameters as $l = 19$, $\sigma = 0.6$, $v^- = -2$, $v^+ = 2$, $u^- = -1$ and $u^+ = 1$. If the same system is approximately described by the KP model with the same parameters except for $\sigma = 1.0$, the compensation (shown as Fig.4.32f) will introduce some error between the outputs of the real system and the modeled hysteresis. In the simulation results of Fig.4.31, all the conditions are same as the open-loop compensation results of (Fig.4.31), except for $\sigma = 1.0$ instead of $\sigma = 0.6$.

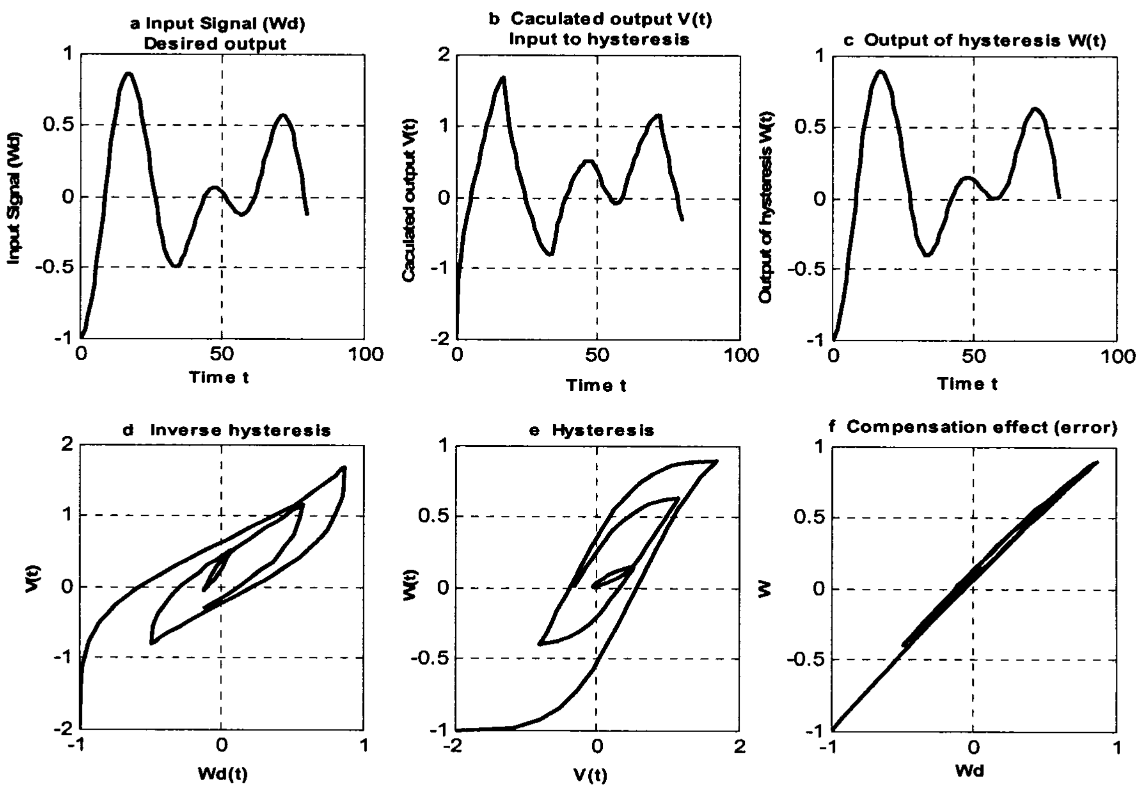


Fig.4.32 Open loop compensation results with inverse KP operator

In general, the hysteresis in actuators cannot be precisely modeled because its parameters can only be represented to a finite limit, and thus the hysteresis effect cannot be completely eliminated. If the hysteresis is represented by the linearly parameterized KP model, the precision of approximating its density distribution and the number l of dividing lines are the determining factors to the accuracy of the model. The accuracy of the inverse KP model is also dependent on these parameters. Consequently, the compensation effect is influenced by the selection of these parameters. To improve the compensation effects as influenced by the parameter values, a methodology of parameter updating based on an adaptive closed-loop control method will be described in later chapters.

CHAPTER 5

MODEL REFERENCE CONTROL OF KNOWN LINEAR SYSTEMS THROUGH ADAPTIVE COMPENSATION OF INPUT HYSTERESIS

In Chapter 4, the modeling, identification and inverse compensation of hysteresis through the linearly parameterized KP model have been presented. In this chapter, the model will be used to describe the actuator hysteresis and the corresponding inverse KP model will be utilized to partially or completely compensate the hysteresis of the actuator. The tracking problem for known linear time-invariant systems with input hysteresis will be solved with the model reference control (MRC) method. An inner loop gradient adaptive compensation of input hysteresis will be used to asymptotically reduce the disturbance injected into the linear systems by the input hysteresis. Choices of MRC controller law and gradient adaptive law for compensation will be derived. Convergence of the parameters for the linearly parameterized KP model will be proven.

5.1 System Description and Control Objective

5.1.1 System Description

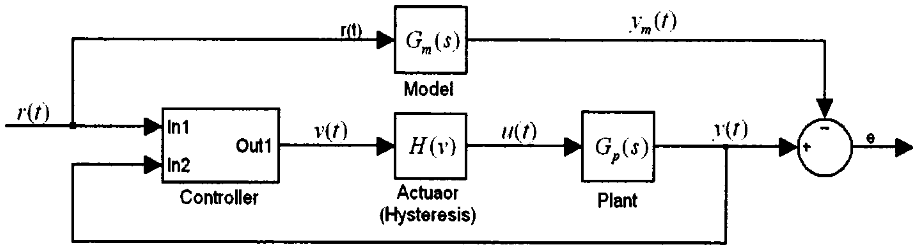


Fig.5.1 Tracking control scheme for linear systems with input hysteresis

Consider a known linear time-invariant plant with input hysteresis from an actuator (see Fig.5.1). The linear plant to be controlled is described by

$$y = G_p(s)u = g_p \frac{N_p(s)}{D_p(s)} u \tag{5.1}$$

where $u(t) \in R$ and $y(t) \in R$ are the plant input and output, respectively; $N_p(s)$ and $D_p(s)$ are monic polynomials of degrees m and n, respectively; and g_p is a constant scalar gain. The hysteresis nonlinearity of the actuator can be described in an operator form as

$$u(t) = H[v](t), \tag{5.2}$$

where $v(t) \in R$ is the input of the actuator and $u(t) \in R$ denotes its output. The operator $H(\cdot)$ is the KP model to describe the hysteresis characteristic of the actuator, which has been presented in detail in Chapter 4. The operator $H(\cdot)$ can be expressed in the linearly parameterized form as

$$u(t) = H(v(t)) = \Gamma^T K, \tag{5.3}$$

where the $N \times 1$ vector Γ consists of parameters of N KP kernels, i.e., $\Gamma = [\mu_1, \mu_2, ..., \mu_N]^T$, and the $N \times 1$ vector K includes output values of N KP kernels as they are subjected to the input $v(t)$.

Some assumptions about the known linear time-invariant plant and the hysteretic actuator are made as follows:

(1) $G_p(s)$ is minimum phase;

(2) The relative degree $n^* = n - m$ of $G_p(s)$ is known, and the transfer function is proper ($n \geq m$);

(3) The degree of pole $D_p(s)$ is known as n;

(4) The sign of g_p is known.

(5) The parameters Γ of the KP model used to describe the actuator hysteresis are not exactly known, but their lower and upper boundaries $(\Gamma^{lower}, \Gamma^{upper})$ and initial estimations $\hat{\Gamma}(t=0)$ are known.

5.1.2 Control Objective

The control objective is to design a feedback control $v(t)$ so that all closed loop signals are bounded and the output $y(t)$ of the plant tracks the output $y_m(t)$ of a reference model defined by

$$y_m = G_m(s)r = g_m \frac{N_m(s)}{D_m(s)} r, \tag{5.4}$$

where $G_m(s)$ is a stable rational proper transfer function of relative degree $n^* = n - m$, the gain g_m is assumed positive without loss of generality, and the command reference $r(t)$ is a bounded piecewise continuous signal.

As the linear time-invariant plant is known, one can use the model reference control scheme to solve the tracking problem while the hysteresis nonlinearity of the actuator is absent. i.e., $u(t) = v(t)$. In the case where the input hysteresis is present, a linear controller alone cannot achieve the control objective even for a known linear plant. If the hysteresis output is measurable, one can use the hysteresis inverse to construct an adaptive compensation configuration to cancel or reduce the effects of the actuator hysteresis (see Fig.5.2 and Fig.5.3). In Fig.5.3 the actuator is represented by an exactly modeled hysteresis.

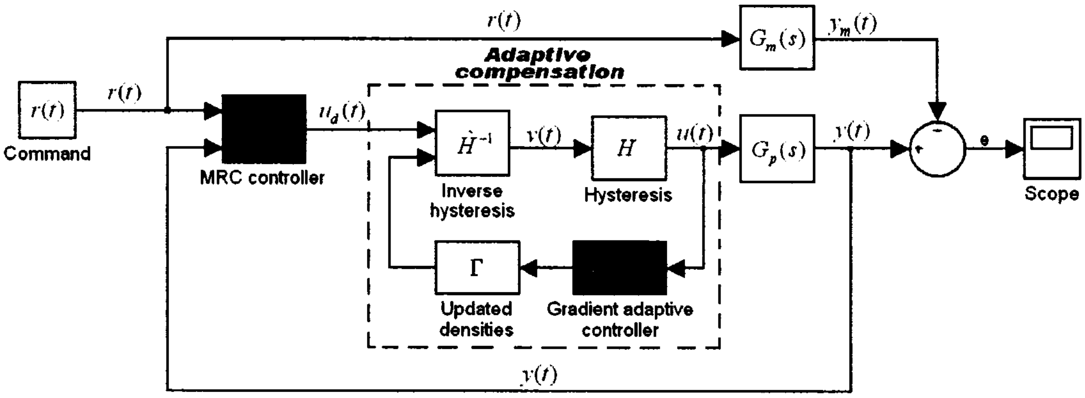


Fig.5.2 MRC configuration of a known linear time-invariant system with adaptively compensated input hysteresis

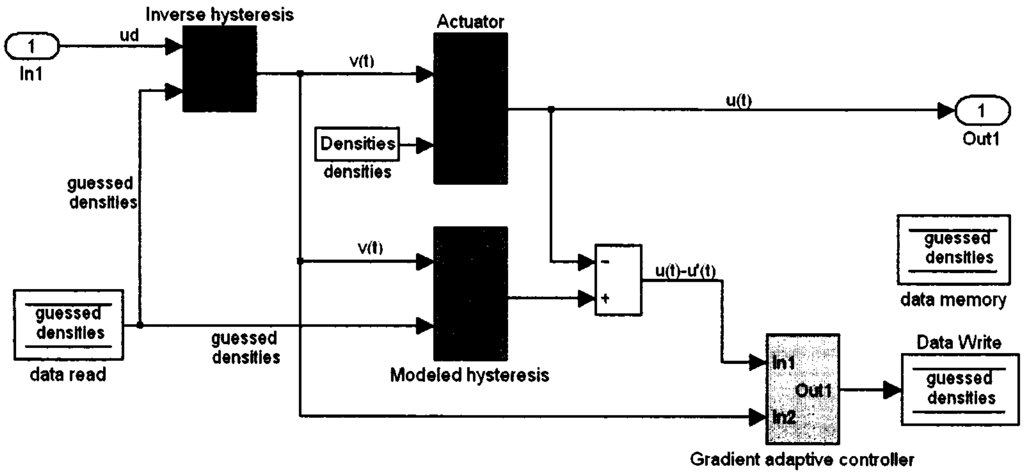


Fig.5.3 Subsystem of gradient adaptive controller in Fig.5.2

5.2 MRC for Known Linear Plants with Measurable Input Hysteresis through Adaptive Compensation for Hysteresis

5.2.1 Gradient Adaptive Compensate for Measurable Input Hysteresis

Choice of Adaptive Law

For a hysteretic actuator, the nonlinear operator mapping the input $v(t)$ into the actuator output $u(t)$ can be represented as $u = H(v)$. According to equation (4.50) from Chapter 4, the output from the KP hysteresis model can be expressed in vector form as $H(v) = \Gamma^T K$. Thus, the hysteresis model is transformed into a suitable form for the gradient adaptive control. Due to inaccurate prior knowledge concerning the exact parameters Γ from the KP hysteresis model, the exact hysteresis parameters Γ can be estimated off-line by the estimated parameters $\hat{\Gamma}$ in advance. Thus, the exact KP model and the estimated KP model can be expressed, respectively, by

$$u = H(v) = \Gamma^T K \tag{5.3}$$

and

$$\hat{u} = \hat{H}(v) = \hat{\Gamma}^T K. \tag{5.5}$$

Furthermore, the error between the outputs of the two models is expressed by

$$e = u - \hat{u} = (\Gamma - \hat{\Gamma})^T K \overset{\Delta}{=} \tilde{\Gamma}^T K \tag{5.6}$$

where $\tilde{\Gamma} \overset{\Delta}{=} \Gamma - \hat{\Gamma}$ is called the parameter error of the input hysteresis. As the exact output u of the hysteresis is measurable, the error $e = u - \hat{u}$ can be calculated, and can be used to carry out the gradient adaptation for $\hat{\Gamma}$ to converge to the exact parameters Γ.

To ensure that the signals are bounded, a normalized estimation error [27] is defined as

$$\varepsilon \overset{\Delta}{=} \frac{e}{m^2} = \frac{u - \hat{u}}{m^2} = \frac{u - \hat{\Gamma}^T K}{m^2} = \frac{\tilde{\Gamma}^T K}{m^2}, \tag{5.7}$$

where $m^2 = 1 + n_s$, and n_s is the normalizing signal designed so that

$$\frac{K}{m} \in L_\infty.$$

As in section 4.3.4 of [27], typical choices for n_s are $n_s^2 = K^T K$, and $n_s^2 = K^T P K$ for $P = P^T > 0$, etc. Thus, by choosing $P = P^T = I$, one has

$$m^2 = 1 + n_s = 1 + K^T P K > 0 \quad for \ \ \forall \ \ P = P^T > 0 \tag{5.8}$$

also,

$$\frac{K}{m^2} = \frac{K}{1 + n_s^2} = \frac{K}{1 + K^T P K} \in L_\infty \quad for \ \ \forall \ \ P = P^T > 0, \tag{5.9}$$

and $$\frac{K}{m} \in L_\infty \quad for \quad P = P^T > 0$$

For example, let $P = P^T = I > 0$, equation (5.8) can be rewritten as

$$m^2 = 1 + n_s = 1 + K^T K > 0 \tag{5.10}$$

one has $$\lim_{t\to\infty} |\frac{K}{m}| = \lim_{t\to\infty} \frac{|K|}{\sqrt{1+K^T PK}} = \lim_{t\to\infty} \frac{|K|}{\sqrt{1+K^T IK}} = [1, \quad 1, \quad ..., \quad 1]^T{}_{N\times 1}$$

Thus, $$\frac{K}{m} \in L_\infty \quad for \quad P = P^T = I > 0$$

From equation (5.7) it is clear that signal $\varepsilon m = \tilde{\Gamma}^T \frac{K}{m} \in L_\infty$ is a bounded signal and a reasonable measure of the parameter error $\tilde{\Gamma}$ because for any piecewise continuous signal vector K, large $|\varepsilon m|$ implies large $|\tilde{\Gamma}|$. Thus adaptive laws for $\hat{\Gamma}$ can be generated by using the gradient method to minimize a wide class of cost functions of εm with respect to the parameters $\hat{\Gamma}$.

Consider a quadratic cost function given by

$$J(\hat{\Gamma}) = \frac{\varepsilon^2 m^2}{2} = \frac{(u - \hat{\Gamma}^T K)^2}{2m^2} \tag{5.11}$$

To minimize the trajectory for the cost function (5.11) with respect to the estimated parameters $\hat{\Gamma}$, it is required to determine the convexity of $J(\hat{\Gamma})$ over the space of $\hat{\Gamma}$ at every moment of time t. This means that the minimum of the cost function $J(\hat{\Gamma})$ requires the variables $\hat{\Gamma}$ to be strictly decreasing to approach the convex point. Applying the gradient method, the minimizing trajectory is generated by the differential equation

$$\dot{\hat{\Gamma}} = -Z\nabla J(\hat{\Gamma}) = Z\frac{(u - \hat{\Gamma}^T K)K}{m^2} = \frac{(\Gamma^T - \hat{\Gamma}^T)K}{m^2} K = Z\frac{\tilde{\Gamma}^T K}{m^2} K \tag{5.12}$$

Finally, substituting equation (5.7) into (5.12) results in the adaptive update law as

$$\dot{\hat{\Gamma}} = \mathrm{Z}\varepsilon K_p \tag{5.13}$$

where, $\mathrm{Z} = \mathrm{Z}^T > 0$ is a scaling matrix of adaptive gains. The larger Z is chosen, the faster $\hat{\Gamma}$ converges to the convex point of the cost function $J(\hat{\Gamma})$, namely to the true value Γ.

To ensure that the estimated parameters $\hat{\Gamma}$ remain bounded in a certain set Q, derived from a prior knowledge of the properties of Γ, a parameter projection term (see Appendix 7) can be used. By defining Q as

$$Q = \{\hat{\Gamma} \in R^{N\times 1} : g(\hat{\Gamma}) \leq 0\} \tag{5.14}$$

where $g(\hat{\Gamma})$ is a vector of constraint equations on estimated parameters $\hat{\Gamma}=[\hat{\mu}_1, \hat{\mu}_2, \ldots, \hat{\mu}_N]^T$, the adaptive update law (5.13) becomes

$$\dot{\hat{\Gamma}} = \Pr(\mathrm{Z}\varepsilon K) = \begin{cases} \mathrm{Z}\varepsilon K & \begin{array}{l} \textit{if} \quad \hat{\Gamma} \in Q_o \\ \textit{or} \quad \hat{\Gamma} \in \delta(Q) \ \ \textit{and} \ \ (\mathrm{Z}\varepsilon K)^T \nabla g \leq 0 \end{array} \\ \\ \mathrm{Z}\varepsilon K - \mathrm{Z}\dfrac{\nabla g \nabla g^T}{\nabla g^T \Pi \nabla g} \mathrm{Z}\varepsilon K & \textit{otherwise} \end{cases} \tag{5.15}$$

where Q_o is the interior of Q, and $\delta(Q)$ is the boundary of Q and $\hat{\Gamma}(t=0)$ is chosen to be in Q, i.e., $\hat{\Gamma}(t=0) \in Q$.

For a considered hysteresis, in case of the existing of positive and negative saturation states, a non-restrictive choice for Q is simple. If one has an estimate from a prior knowledge on the maximum achievable output of the hysteretic actuator, u^+, then any individual $\hat{\mu}_i$ of the vector $\hat{\Gamma}$ would never be greater than u^+. This result comes from following reasoning:

For $v(t)\ge v^+$, $\hat{u}=\hat{H}(v)=\hat{\Gamma}^T K=[\hat{\mu}_1,\hat{\mu}_1,\ldots,\hat{\mu}_N][1,1,\ldots,1]^T{}_{N\times 1}=u^+$. Since $\hat{\mu}_{i=1\sim N}\ge 0$, one has $\hat{\mu}_i\le u^+\ \ \forall i=1\sim N$.

Therefore, referring to (5.14), the set Q can be defined by

$$Q=\{\hat{\Gamma}\in R^{N\times 1}:\hat{\mu}_i-u^+\le 0,\forall i=1\sim N\} \tag{5.16}$$

where $\hat{\mu}_i=u^+\ \ \forall i=1\sim N$ is the upper bound of $\hat{\mu}_i$. Since $g_i(\hat{\Gamma})=g(\hat{\mu}_i)=\hat{\mu}_i-u^+$ $\forall i=1\sim N$, then $\nabla g=\nabla[g_1(\hat{\Gamma}),g_2(\hat{\Gamma}),\ldots,g_N(\hat{\Gamma})]^T=[1,\ \ 1,\ \ \ldots,\ \ 1]^T{}_{N\times 1}$.

If $\hat{\Gamma}\in Q_o$ (i.e., $\hat{\mu}_i-u^+<0$ for $i=1\sim N)$, the values of estimated parameters $\hat{\Gamma}$ is updated by the adaptive law $\dot{\hat{\Gamma}}=Z\varepsilon K$.

If $\hat{\Gamma}\in\delta(Q)$, i.e., $\hat{\mu}_i-u^+=0\forall i=1\sim N$ and $Z\varepsilon K\nabla g=Z\varepsilon K[1,\ \ 1,\ \ \ldots,\ \ 1]^T{}_{N\times 1}=Z\varepsilon K\le 0$, one has $\dot{\hat{\Gamma}}=Z\varepsilon K\le 0$ which ensures the trajectory remains in the set of Q.

If the $\hat{\Gamma}\in\delta(Q)$ and $(Z\varepsilon K)^T\nabla g=Z\varepsilon K>0$, $\dot{\hat{\Gamma}}=Z\varepsilon K>0$ can not ensure the trajectory remains on the boundary of the set of Q. In order to keep the trajectory remaining on the boundary of the set of Q, one can set

$$\dot{\Theta}=Z\varepsilon K-Z\frac{\nabla g\nabla g^T}{\nabla g^T\Pi\nabla g}Z\varepsilon K=0$$

Thus, the adaptive update law with projection (5.15) can be rewritten as

$$\dot{\hat{\Gamma}}=proj\ (Z\varepsilon K)=\begin{cases}Z\varepsilon K & \begin{matrix} if\ \ \hat{\mu}_i<u^+ \\ or\ \ \hat{\mu}_i=u^+\ \ and\ \ Z\varepsilon K\le 0\end{matrix}\ for\ i=1\sim N \\ \\ 0 & otherwise\end{cases} \tag{5.17}$$

Parameter Convergence

The gradient adaptive law (5.17) guarantees that [27]

(i) $\varepsilon, \varepsilon n_s, \hat{\Gamma}, \dot{\hat{\Gamma}} \in L_\infty$;

(ii) $\varepsilon, \varepsilon n_s, \dot{\hat{\Gamma}} \in L_2$ are independent of the boundedness of the signal vector K;

(iii) if $n_s, \hat{\Gamma} \in L_\infty$ and K is *persistence of excitation (*PE), then $\hat{\Gamma}$ converges exponentially to Γ and

(iv) $\hat{\Gamma}(t) \in Q \;\; \forall t \geq 0$ provided $\hat{\Gamma}(0) = \hat{\Gamma}_0 \in Q$ and $\Gamma \in Q$.

Proof: Because Γ is a constant vector, from the definition $\tilde{\Gamma} \overset{\Delta}{=} \Gamma - \hat{\Gamma}$, one has $\dot{\tilde{\Gamma}} = -\dot{\hat{\Gamma}}$. Substituting $\dot{\tilde{\Gamma}} = -\dot{\hat{\Gamma}}$ into the gradient adaptive law (5.17) one has

$$\dot{\tilde{\Gamma}} = \begin{cases} -Z\varepsilon K \\ 0 \end{cases} \tag{5.18}$$

If a Lyapunov-like function is chosen as

$$V(\tilde{\Gamma}) = \frac{\tilde{\Gamma}^T Z^{-1} \tilde{\Gamma}}{2} \geq 0 \tag{5.19}$$

then along the solution of (5.18), one has

$$\dot{V}(\tilde{\Gamma}) = \begin{cases} -\tilde{\Gamma}^T K \varepsilon \\ 0 \end{cases}$$

From(5.7), one has $\tilde{\Gamma}^T K = \varepsilon m^2$. Substituting it into the above equation yields

$$\dot{V}(\tilde{\Gamma}) = \begin{cases} -\tilde{\Gamma}^T K \varepsilon = -\varepsilon^2 m^2 \leq 0 \\ 0 \end{cases} \tag{5.20}$$

Hence, $V(\tilde{\Gamma})$, $\tilde{\Gamma} \in L_\infty$, which, together with equation (5.7), implies that $\varepsilon, \varepsilon m^2 \in L_\infty$. In addition, from the properties of $V(\tilde{\Gamma})$ and $\dot{V}(\tilde{\Gamma})$ one has $\varepsilon m \in L_2$, which implies that $\varepsilon, \varepsilon n_s \in L_2$. Now from (5.18) one has

$$|\dot{\tilde{\Gamma}}| = |\dot{\hat{\Gamma}}| \leq \| Z \| | \varepsilon m | \frac{|K|}{m} \tag{5.21}$$

which together with $\frac{|K|}{m} \in L_\infty$ and $\varepsilon m \in L_2 \cap L_\infty$ implies that $\dot{\hat{\Gamma}} \in L_2 \cap L_\infty$ and the proof of (i) and (ii) is complete.

The proof for (iii) is long and complicated and the reader may refer to Section 4.8 of reference [27].

The proof for (iv) is comes from equation (5.17) that whenever $\hat{\Gamma} \in \delta(Q)$, i.e., $\hat{\mu}_i = u^+, \forall i = 1 \sim N$, one has $\dot{\hat{\Gamma}}^T \nabla g \leq 0$, which implies that the vector $\dot{\hat{\Gamma}}$ points either inside Q or along the tangent plane of boundary of Q at point $\hat{\Gamma}$. Because $\hat{\Gamma}(0) = \hat{\Gamma}_0 \in Q$, it follows that $\hat{\Gamma}(t)$ will never leave Q, i.e., $\hat{\Gamma}(t) \in Q \;\; \forall t \geq 0$.

Remarks: The property $V(\tilde{\Gamma}) \geq 0$ and $\dot{V}(\tilde{\Gamma}) \leq 0$ of the Lyapunov-like function implies that $\lim_{t \to \infty} V(\tilde{\Gamma}(t)) = V_\infty$. This does not imply $\lim_{t \to \infty} \dot{V}(\tilde{\Gamma}(t)) = 0$. Consequently, one cannot conclude that ε or εm go to zero as $t \to \infty$. i.e., that the steepest descent reaches the global minimum that corresponds to $\nabla J(\hat{\Gamma}) = -\varepsilon K = 0$. If however, $\frac{\dot{K}}{m}, \frac{\dot{m}}{m} \in L_\infty$, one can establish that $\frac{d}{dt}(\varepsilon m) \in L_\infty$, which, together with $\varepsilon m \in L_2$, implies that $\varepsilon(t) m(t) \to 0$ as $t \to \infty$ [27]. Because $m^2 = 1 + n_s^2$, one has $\varepsilon(t) \to 0$ as $t \to \infty$ and from (5.21) that

$\dot{\hat{\Gamma}}(t) \to 0$ as $t \to \infty$. Now $|\nabla J(\hat{\Gamma})| \le |\varepsilon K| \le |\varepsilon m| \frac{|K|}{m}$, which implies that $|\nabla J(\hat{\Gamma})| \to 0$ as $t \to \infty$, i.e., $\hat{\Gamma}(t)$ converges to a trajectory that corresponds to a global minimum of $J(\hat{\Gamma})$ asymptotically with time provided $\frac{\dot{K}}{m}$, $\frac{\dot{m}}{m} \in L_\infty$ [27].

5.2.2 MRC Design for Output Tracking of Known Linear Systems

In this section, a model reference controller will be designed for the linear time-invariant plant with the known transfer function (5.1) to track the linear reference model (5.4).

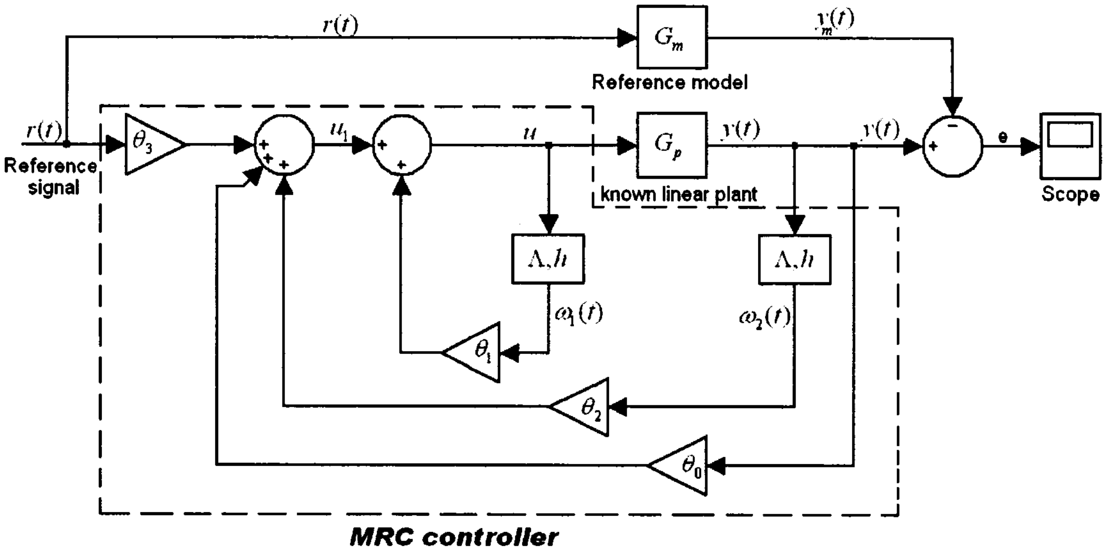


Fig.5.4 An MRC control system with perfect tracking

In the above section, the configuration (Fig.5.3) about the gradient adaptive compensation of actuator hysteresis through inverse hysteresis has been introduced. This scheme acts as the subsystem "***adaptation compensation***" of Fig.5.2. The subsystem passes the output $u_d(t)$ of the MRC controller together with an error to the input of the linear plant. The injected error asymptotically converges to zero, i.e., $u(t) = u_d(t) + e(t)$, where $e(t) \to 0$ as $t \to \infty$. To design the MRC controller for the known linear time-invariant plant, the parameters of the MRC controller must be found in advance. Thus, disregarding the error injected into the linear plant, one can directly construct an MRC configuration as shown in Fig.5.4.

In order to modulate the transfer function of the known linear plant with relative degree $n^* = n - m$ to be the same as the model transfer function with the same relative degree, some gains θ_0, θ_1, θ_2, θ_3 and filter signals $\omega_1(t)$, $\omega_2(t)$ in the controller structure, as shown in the Fig.5.4, must be introduced. The structure of this control system can be explained as follows.

Parameters θ_0, θ_1, θ_2 and θ_3 represent control gains which lead to perfect tracking when the plant parameters are known. The block for generating the filter signal $\omega_1(t)$ represents an $(n-1)^{th}$ order dynamics, which can be described by

$$\omega_1 = \frac{h(s)}{\Lambda(s)} u \tag{5.22}$$

where $\Lambda(s)$ is an $(n-1)$ degree monic polynomial and $h(s)$ is a $(n-2)$ degree monic polynomial such that $h(s)/\Lambda(s)$ is controllable. The monic polynomial $\Lambda(s)$ is chosen so that

$$\Lambda(s) = \tilde{\Lambda}(s) N_m(s) \tag{5.23}$$

For the plant and the reference model with relative degree one $(n^* = n - m = 1)$, $\tilde{\Lambda}(s) = 1$, $\Lambda(s) = N_m(s)$. The block for generating the signal $\omega_2(t)$ has the same dynamics but with $y(t)$ as its input,

$$\omega_2 = \frac{h(s)}{\Lambda(s)} y \ . \tag{5.24}$$

Since the scalar gain θ_3 in Fig.5.4 is intended to modulate the gain g_p of the plant to have the same gain g_m as the reference model, it is directly defined as

$$\theta_3 = \frac{g_m}{g_p}. \tag{5.25}$$

The vector θ_1 contains $(n-1)$ parameters which are intended to cancel the zero of the plant. It can be understood that the assumption for the plant to be minimum-phase allows the plant zeros to be cancelled by the controller poles. The vector θ_2 contains $(n-1)$ parameters which together with the scalar gain θ_0 can move the poles of the closed loop control system to the locations of the reference model poles.

The control input in this system is a linear combination of the reference signal $r(t)$, the vector signal $\omega_1(t)$ obtained by filtering the control input $u(t)$, the signal $\omega_2(t)$ obtained by filtering the plant output $y(t)$, and the output itself. The control input $u(t)$ can thus be written, in terms of the gains and the various signals as

$$u(t) = \theta_3 r(t) + \theta_1 \omega_1(t) + \theta_2 \omega_2(t) + \theta_0 y(t), \tag{5.26}$$

which is denoted as ***the model reference controller*** for model matching.

Since these parameters in the control law (5.26) result in perfect tracking, for any reference input $r(t)$, the output of the plant must equal the output of the reference model, i.e.,

$$y = G_p(s)u = y_m = G_m(s)r \tag{5.27}$$

To calculate the perfect tracking gains $\theta_0(t)$, $\theta_1(t)$, $\theta_2(t)$, $\theta_3(t)$ and choose the filter $h(s)/\Lambda(s)$, the following two cases will be considered to derivate the model matching formulas.

Case 1: $n=1$, first-order plant $G_p(s)$ and first-order reference model $G_m(s)$

Since $n=1$ it is impossible to select $h(s)$ as a $(n-2)$ degree monic polynomial. Thus, one can set $h(s)=0$. This means that the filtered signals $\omega_1(t)$ and $\omega_2(t)$ from the input $u(t)$ and the output $y(t)$ of the linear plant are not required, instead, the reference signal $r(t)$ and the output $y(t)$ of the plant can be used to construct **MRC *controller*** as

$$u(t)=\theta_3 r(t)+\theta_0 y(t). \tag{5.28}$$

Substituting (5.28) into the plant transfer function (5.1) obtains

$$y=G_p(s)u=g_p\theta_3\frac{N_p}{D_p}r+g_p\theta_0\frac{N_p}{D_p}y. \tag{5.29}$$

Solving y from (5.29) yields $$y=g_p\theta_3\frac{N_p}{D_p-g_p\theta_o N_p}r. \tag{5.30}$$

Substituting (5.30) and (5.4) into the model matching condition (5.27), one has

$$y=g_p\theta_3\frac{N_p}{D_p-g_p\theta_o N_p}r=y_m=g_m\frac{N_m}{D_m}r \tag{5.31}$$

Considering $\theta_3=g_m/g_p$ as (5.25) results in

$$\theta_o=\frac{N_m D_p-N_p D_m}{g_p N_p N_m}r. \tag{5.32}$$

If the plant is expressed as $$\dot{y}=-a_p y+b_p u \tag{5.33}$$

and the reference model is described as

$$\dot{y}_m=-a_m y_m+b_m r, \tag{5.34}$$

the ***model matching formulas*** (5.25) and (5.32) to calculate the exact parameters for perfect tracking are rewritten as

$$\theta_3=\frac{b_m}{b_p} \quad \text{and} \quad \theta_o=\frac{a_p-a_m}{b_p}.$$

Case 2: $n \geq 2$ high-order linear plant $G_p(s)$ and reference model $G_m(s)$

For $n \geq 2$, $h(s) = [s^{n-2}, s^{n-3}, \ldots, s, 1]$ and $\Lambda(s) = \tilde{\Lambda}(s) N_m$ with degree of $n-1$, substituting equations (5.22), (5.24) and (5.25) into the MRC controller (5.26) and solving the output y of plant yields

$$y = \frac{g_m N_p \Lambda}{D_p(\Lambda - \theta_1 h) - g_p N_p(\theta_o \Lambda + \theta_2 h)} r \tag{5.35}$$

Substituting equations (5.35) and (5.4) into the model matching condition (5.27), one has

$$\begin{aligned} y &= \frac{g_m N_p \Lambda}{D_p(\Lambda - \theta_1 h) - g_p N_p(\theta_o \Lambda + \theta_2 h)} r \\ &= g_m \frac{N_m}{D_m} r = y_m \end{aligned} \tag{5.36}$$

With $\Lambda(s) = \tilde{\Lambda}(s) N_m$, (5.36) can be rewritten as

$$g_p N_p(\theta_o \Lambda + \theta_2 h) + \theta_1 h D_p = D_p \Lambda - D_m N_p \tilde{\Lambda} \tag{5.37}$$

Equations (5.37) and (5.25) are denoted as the ***model matching formulas.***

5.3 MRC for Known Linear Systems with Non-measurable Input Hysteresis through Adaptive Hysteresis Compensation

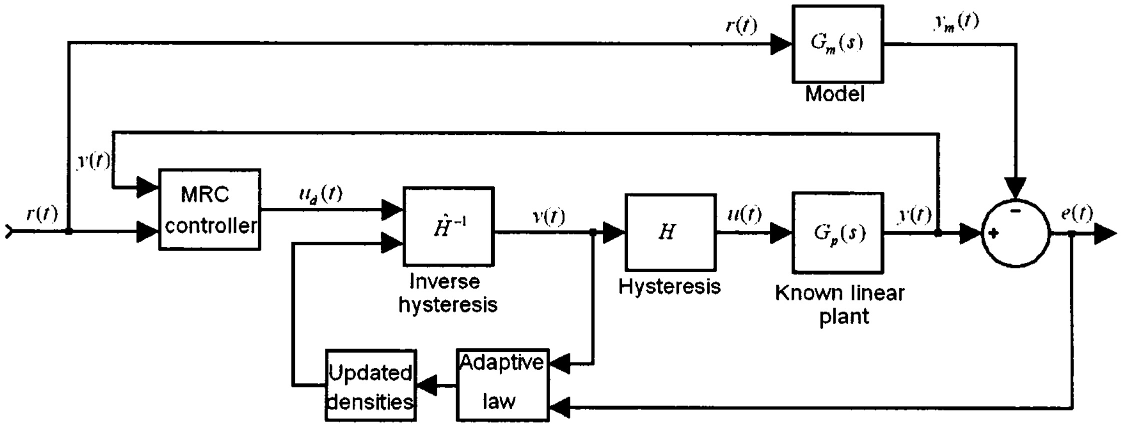


Fig.5.5 Diagram for adaptive hysteresis compensation with immeasurable hysteresis output

In this case, an adaptive control scheme will be designed for the known linear plant (5.1) with a hysteresis input to track the reference model (5.4) as the hysteresis input is not measurable.

To design this control scheme, it is assumed that the plant with a completely compensated hysteresis input $(\hat{H} = H)$ is controlled by an MRC scheme (Fig.5.5). According to the derivation in the above section, the plant input can be generated by hysteresis as

$$u(t) = H(H^{-1}(u_d(t))) = u_d(t) \ , \tag{5.38}$$

and the control signal $u_d(t)$ is provided by an MRC controller as

$$u_d(t) = \theta_3 r(t) + \theta_1 \omega_1(t) + \theta_2 \omega_2(t) + \theta_0 y(t) \tag{5.39}$$

with filtered signals defined as $\omega_1 = \dfrac{h(s)}{\Lambda(s)} u_d$ and $\omega_2 = \dfrac{h(s)}{\Lambda(s)} y$, (5.40)

gain θ_3 selected as $\theta_3 = \dfrac{g_m}{g_p}$, (5.41)

and other gains $\theta_1, \theta_2 \in R^{n-1}$, $\theta_0 \in R$ satisfying the following Diophantine equation

$$g_p N_p (\theta_o \Lambda + \theta_2 h) + \theta_1 h D_p = D_p \Lambda - D_m N_p \widetilde{\Lambda} \tag{5.42}$$

where $\Lambda(s) = \widetilde{\Lambda}(s) N_m$ (5.43)

with $\widetilde{\Lambda}(s) = 1$ if $n^* = n - m = 1$, or $\widetilde{\Lambda}(s)$ is a monic Hurwitz polynomial of degree $n-1$ if $n^* = n - m > 1$.

However, it is impossible to exactly model the hysteresis of the actuator. There exists a parameter error $\widetilde{\Gamma}(t) = \Gamma - \hat{\Gamma}(t)$ between the modeled hysteresis and its actual hysteresis. And the modeled linearly parameterized hysteresis is solely determined by its parameter vector $\Gamma(t)$ for a predetermined modeling precision, which is decided by the number of lines l for uniformly dividing the Preisach plane defined by a specific hysteresis input region$[v^-,v^+]$ horizontally and vertically. Consequently, the inverse hysteresis, which is constructed based on the modeled hysteresis, cannot completely cancel the hysteresis of actuator, and the cancellation effect of the hysteresis requires to be continuously improved by updating the parameter vector of the modeled hysteresis. Since the output of the actuator is not measurable in this case, an adaptive controller should be developed to update the inverse hysteresis according to the output error $e(t)$ between the plant $G_p(s)$ and the reference model $G_m(s)$. Thus, to implement this adaptive control, one should find out the relationship between the parameter error $\tilde{\Gamma}(t) = \Gamma - \hat{\Gamma}(t)$ and the output error $e(t)$.

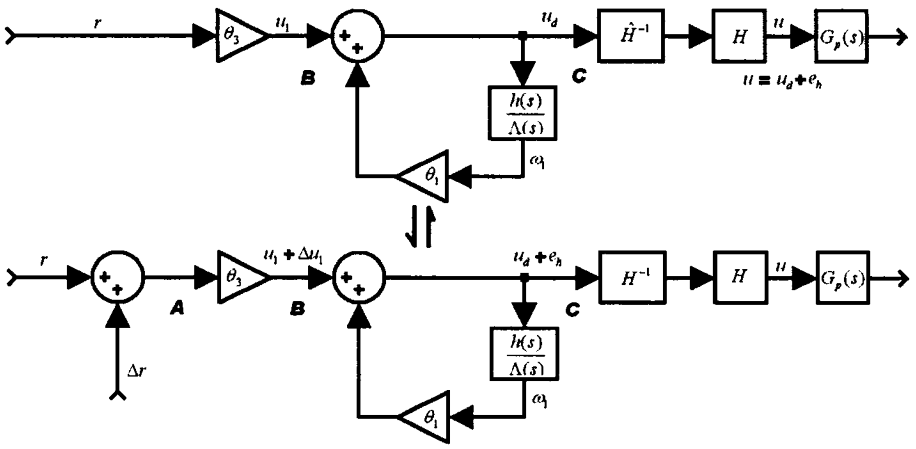


Fig.5.6 Equivalent diagrams: (the upper for an input error $e_h(t)$ caused by inaccurate compensation, the lower for an extra input $e_h(t)$ of plant with exact hysteresis compensation caused by extra reference signal)

An input error of the plant caused by the parameter error $\tilde{\Gamma}(t)$ can be expressed as

$$e_h(t) = u(t) - u_d(t) = \Gamma^T K(v) - \hat{\Gamma}^T K(v) = \tilde{\Gamma}^T K(v). \qquad (5.44)$$

Since the error $e_h(t)$ occurs between the MRC controller and the plant, it can be equivalently considered as another reference signal $\Delta r(t)$ together with the original reference command $r(t)$ applied to an MRC control system with perfect hysteresis compensation (Fig.5.4) to cause the plant input to vary.

To evaluate the extra reference signal $\Delta r(t)$, examine two equivalent diagrams in the Fig.5.6, where the upper diagram represents that an input error $e_h(t)$ to the plant is caused by inaccurate compensation; and the lower one represents that an extra input $e_h(t)$ to the plant with exact hysteresis compensation is caused by an extra reference signal $\Delta r(t)$. In the upper diagram, the signal relationship between point B and point C is

$$u_1 + \theta_1 \frac{h(s)}{\Lambda(s)} u_d = u_d \tag{5.45}$$

and in the lower diagram, the signal relationship between point B and point C is

$$u_1 + \Delta u_1 + \theta_1 \frac{h(s)}{\Lambda(s)} (u_d + e_h) = u_d + e_h \tag{5.46}$$

Subtracting (5.45) from (5.46) and solving for Δu_1 yields

$$\Delta u_1 = (1 - \theta_1 \frac{h(s)}{\Lambda(s)}) e_h \tag{5.47}$$

Further considering the signal relationship between the point A and point B in the lower diagram obtains

$$\Delta r = \frac{1}{\theta_3} (1 - \theta_1 \frac{h(s)}{\Lambda(s)}) e_h(t) \tag{5.48}$$

This extra reference signal Δr could result in an extra output Δy_m if it were applied to the reference model $G_m(s)$. Thus, the output error of the plant caused by the inaccurate compensation equals to

$$e(t) = y - y_m = \Delta y_m = \Delta r G_m(s)$$

$$= \frac{1}{\theta_3} G_m(s)[1 - \theta_1 \frac{h(s)}{\Lambda(s)}] e_h(t) \tag{5.49}$$

Substituting equation (5.44) into (5.49) gives

$$e(t) = \frac{1}{\theta_3} G_m(s)[1 - \theta_1 \frac{h(s)}{\Lambda(s)}] \tilde{\Gamma}^T K(v(t)) \tag{5.50}$$

With a new transfer function defined by

$$W(s) = \frac{1}{\theta_3} G_m(s)[1 - \theta_1 \frac{h(s)}{\Lambda(s)}], \tag{5.51}$$

which is a stable, strictly proper, and known transfer function, equation (5.50) can be simply expressed as

$$e(t) = W(s) \tilde{\Gamma}^T K(v(t)) \tag{5.52}$$

Equation (5.52) represents a common form for system identification, in which the parameters can be updated by a number of standard techniques such as the gradient method or the least-squares method. Using the gradient method with normalization, the parameters can be updated by an adaptive law combining with a projection operator that uses a prior knowledge of the boundaries of parameters of hysteresis model. The projected adaptive law [25, 27] is expressed as

$$Proj(\dot{\hat{\Gamma}}(t)) = \begin{cases} -\frac{Z\varsigma_h(t)\varepsilon_h(t)}{m_h^2(t)} & \textit{if} \quad \hat{\Gamma} \in (\Gamma^{lower}, \Gamma^{lower}) \\ & \textit{or} \quad \hat{\Gamma} = \Gamma^{lower} \; \textit{and} \; \frac{Z\varsigma_h(t)\varepsilon_h(t)}{m_h^2(t)} \geq 0 \\ & \textit{or} \quad \hat{\Gamma} = \Gamma^{upper} \; \textit{and} \; \frac{Z\varsigma_h(t)\varepsilon_h(t)}{m_h^2(t)} \leq 0 \\ 0 & \textit{otherwise} \end{cases} \tag{5.53}$$

where $\varepsilon_h(t)$ is an augmented error defined by

$$\varepsilon_h(t) = e(t) + \xi_h(t) \tag{5.54}$$

using an auxiliary error as

$$\xi_h(t) = \Gamma^T(t)\varsigma_h(t) - W(s)[\Gamma^T F(v)](t) \tag{5.55}$$

where

$$\varsigma_h(t) = W(s)[F(v)](t), \tag{5.56}$$

and the normalization signal $m_h(t)$ is defined by

$$m_h(t) = \sqrt{1 + \varsigma_h^T(t)\varsigma_h(t) + \xi_h^2(t)} \tag{5.57}$$

The theorem and the subsequent proof that adaptive law (5.53) ensures signal boundedness can be founded in reference [25, 28].

5.4 Simulation Studies

5.4.1 MRC of Known Linear Plant with Measurable Input Hysteresis

In this section, the methodology introduced in section 5.2 will be illustrated by application on a simple linear plant with a known transfer function as

$$y = G_p(s)u = \frac{s+1}{s^2+3s-1}u \tag{5.58}$$

The plant is unstable since it has a positive pole at $s = (3+\sqrt{13})/2 > 0$. The plant is preceded by a hysteretic actuator which in turn injects nonlinearity into the plant. The output of the hysteretic actuator is assumed measurable.

The control objective is to design a controller to force the output of the plant (5.58) to track the output of a reference model described by

$$y_m = G_m(s)r = \frac{s+4}{s^2+6s+9}r. \tag{5.59}$$

If the hysteresis of the actuator can be perfectly modeled, a perfect inverse model could be derived to cancel the nonlinearity from the actuator. In fact, the parameters of the hysteresis KP model are impossible to be identified exactly, thus, an inaccurate inverse compensation is usually carried out to reduce the hysteresis nonlinearity of the actuator. If the output of the actuator is measurable, a gradient adaptive controller can be designed to guarantee that the injected error into the plant converges to zero. In this situation, for the known linear plant, the tracking problem can be accomplished by the method of model reference control, which matches the plant by introducing a controller to be identical to the reference model. The parameters of the MRC scheme are determined by omitting the actuator and assume that the signal from the controller is directly applied to the plant.

In the transfer function of the known linear plant (5.58), $N_p = s+1$, $D_p = s^2+3s-1$, and $g_p = 1$; in the model transfer function (5.59), $N_m = s+4$, $D_m = s^2+6s+9$ and $g_m = 1$. Since the plant and the reference model are second order systems with relative degree one ($n^* = n-m = 2-1 = 1$), one has $h(s) = 1$, and $\Lambda(s) = \tilde{\Lambda}(s) = s+4$. Also $\theta_3 = g_m / g_p = 1$. Substituting all the above variables into the model matching condition (5.37) yields the parameters as $[\theta_3 \quad \theta_1 \quad \theta_2 \quad \theta_0] = [1 \quad 3 \quad 2 \quad -3]$,and the model reference controller (5.26) is can be written as

$$u_d(t) = r(t) + 3\omega_1(t) + 2\omega_2(t) - 3y(t) \tag{5.60}$$

with the filtered signal $\omega_1 = \frac{1}{s+4} u_d$ and $\omega_2 = \frac{1}{s+4} y$; and as an example, the reference command to the closed loop system is chosen as

$$r = -0.16[\cos(\pi t/3) + \cos(\sqrt{2}\pi t/3)]. \tag{5.61}$$

Case 1: **MRC for known linear plant without input hysteresis**

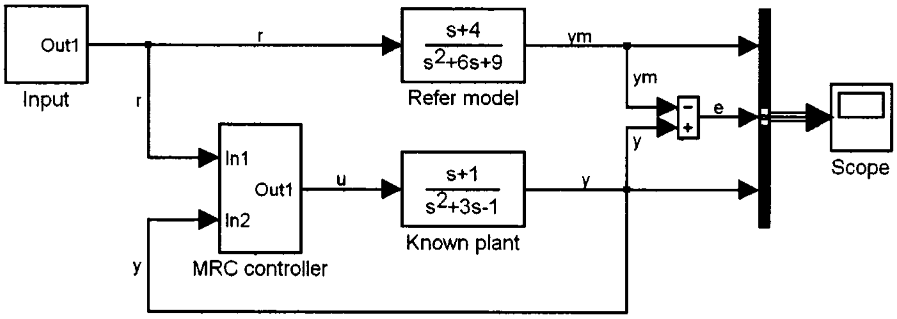


Fig.5.7 Simulation model for MRC of a known unstable linear plant

The simulation model for model reference control to the plant (5.58) is established in the Simulink environment as in Fig.5.7, and the simulation result is shown in Fig.5.8.

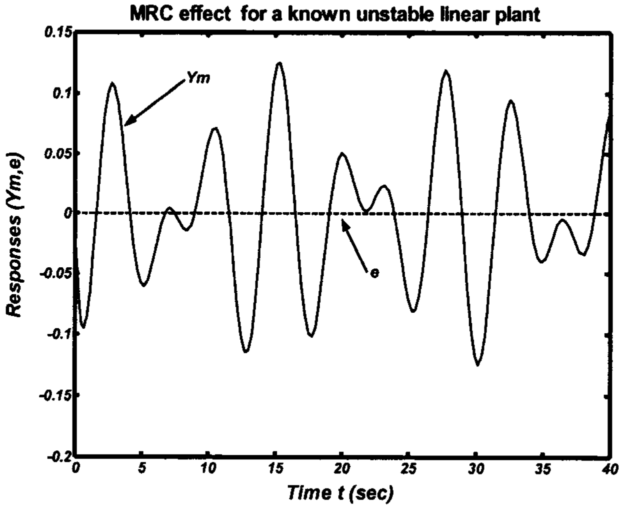


Fig.5.8 MRC tacking effect for a known unstable linear plant

With reference to Fig.5.8, one can see that the tracking error is zero for all simulation time. This is because the linear plant is a time-invariant system, and the parameters of the MRC controller are calculated based on the plant transfer function without any adaptation to the hysteresis.

Case 2: **MRC for known linear plant with input hysteresis**

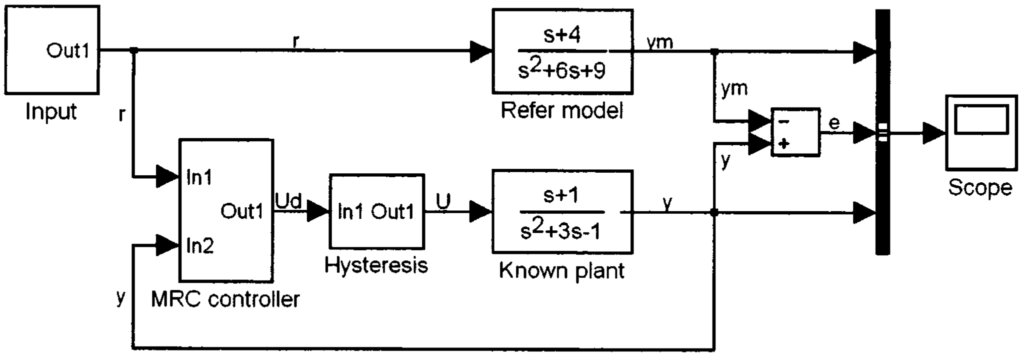


Fig.5.9 Simulation model for MRC of a known unstable linear plant with input hysteresis

However, if there is hysteresis existing in the actuator, it will introduce hard nonlinearly error into the plant. To examine the influence of the hysteresis on the tracking effect of the MRC of the linear plant, a simulation model for the linear plant with uncompensated hysteresis is established as in Fig.5.9, where the parameters for exact hysteresis is listed in table 5.1, and the simulation result is shown as Fig.5.10.

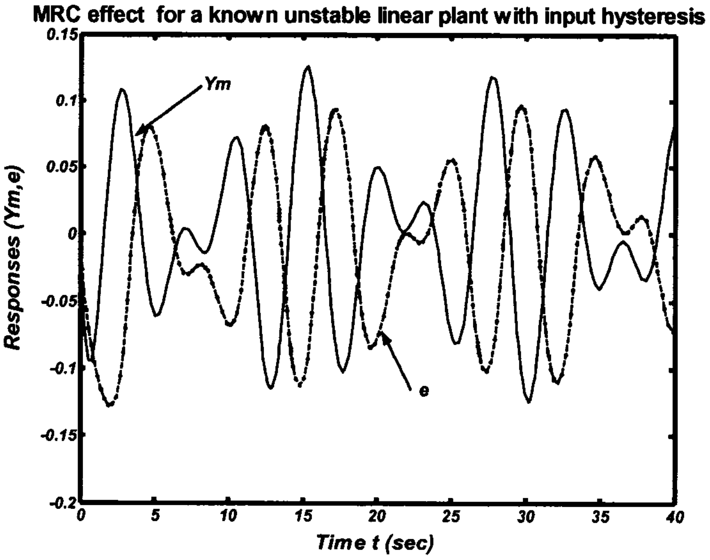


Fig.5.10 MRC tracking effect for a known unstable linear plant with input hysteresis

Table 5.1 Parameters for hysteresis and estimated hysteresis model

Characteristics	Hysteresis	Estimated hysteresis (initial guess)
Discretization number	$l = 19, i \ge j = 1,2,\ldots,l+1$	$l = 19, i \ge j = 1,2,\ldots,l+1$
Number of nodes	$N = (l+2)(l+3)/2 = 231$	$N = (l+2)(l+3)/2 = 231$
Hysteresis input region	$v(t) \in [v^-, v^+] = [-4, +4]$	$v(t) \in [v^-, v^+] = [-4, +4]$
Parameter distribution	$\mu(\alpha_i, \beta_j) = \frac{1}{2\pi\sigma^2} e^{\frac{-1}{2\sigma^2}((\alpha_i+\alpha_0)^2+(\beta_j+\beta_0)^2)}$	$\hat{\mu}(\alpha_i, \beta_j) = \frac{1}{2\pi\sigma^2} e^{\frac{-1}{2\sigma^2}((\alpha_i+\alpha_0)^2+(\beta_j+\beta_0)^2)}$
	$\alpha_0 = -0.1, \beta_0 = 0.6, \text{and}\, \sigma = 1.8$	$\alpha_0 = -0.1, \beta_0 = 0.6, \text{and}\, \sigma = 1.0$
	$\theta_{ij} = \mu(\alpha_i, \beta_j) / \sum_{i=1}^{l+1} \sum_{j=1}^{i} \mu(\alpha_i, \beta_j)$	$\hat{\theta}_{ij} = \hat{\mu}(\alpha_i, \beta_j) / \sum_{i=1}^{l+1} \sum_{j=1}^{i} \hat{\mu}(\alpha_i, \beta_j)$

From Fig.5.10, one can see that the tracking error is very large. The MRC controller can only replace the poles of the plant into the stable region to make the unstable plant become stable, but can not ensure the output of the plant to track the output of the model. This means the error injected from the hysteretic actuator is severely disturbing the linear plant and the hysteresis is required to be cancelled or reduced.

***Case 3:* MRC for known linear plant with fix compensated input hysteresis**

Due to the MRC scheme being not able to reject the disturbance from the input, an inverse hysteresis based on modeling of the hysteresis can be constructed to compensate the hysteresis. However, due to the inaccurate modeling of the hysteresis, the exact inverse hysteresis cannot be found. Therefore, because of inaccurate modeling as well as computation errors, disturbances will still be injected into the linear plant. This raises some tracking error in the MRC scheme. The simulation model for model reference control to the plant (5.58) with inaccurate compensation of its input hysteresis is setup as shown in Fig.5.11, where the parameters for exact hysteresis and estimated hysteresis are listed in table 5.1, and the simulation result is shown in Fig.5.12.

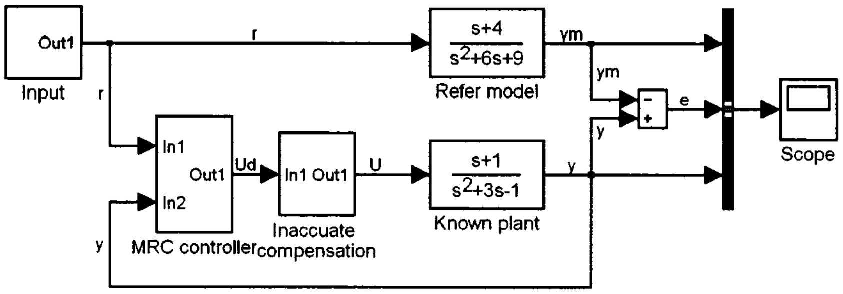


Fig.5.11 Simulation model for MRC of a known unstable linear plant with inaccurate compensated input hysteresis

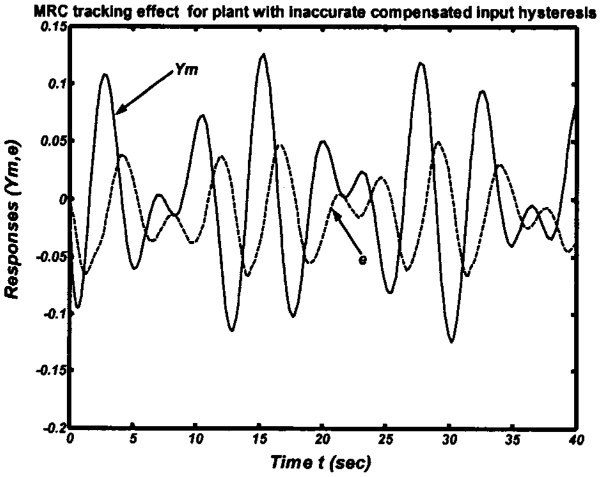


Fig.5.12 MRC tracking effect for a known unstable linear plant with inaccurate compensated input hysteresis

Comparing the tracking error in the Fig.5.12 with the tracking error in the Fig.5.10, one can observe that the tracking error is reduced by about 50%. But, there is still relative large tracking error in this inaccurate compensation method. This is caused by the inaccurate estimation parameter $\sigma = 1.0$ to the exact $\sigma = 1.8$.

***Case 4:* MRC for known linear plant with adaptive compensated input hysteresis**

In fact, the exact parameters are impossible to be investigated. If the output of the actuator is measurable, a gradient adaptive method can be utilized to ensure that the disturbance injected into the plant asymptotically converging to zero. The gradient adaptive method and its convergence have been presented in section 5.2. The simulation model for model reference control to the plant (5.58) with gradient adaptive compensation of the hysteresis of the actuator is established as in Fig.5.13, where the simulation parameters for representing the exact hysteresis and initial estimated hysteresis are listed in table 5.1. The simulation result with the adaptive gain η chosen as $\eta = 2$ is shown in Fig.5.14. The tracking error has been already extremely reduced (see Fig.5.14), and the convergence of the parameters in the simulation period ($t = 40\text{sec}$) is relative smooth (see Fig.5.15). If the adaptative gain is chosen as large as $\eta = 40$ the tracking error is reduced almost to zero (see Fig.5.16), but the convergence of the parameters in the simulation period ($t = 40\text{sec}$) is relatively oscillatory with much dynamics involved (see Fig.5.17); this maybe stimulating the unmodeled dynamics in the plant which is causing the MRC scheme to become unstable.

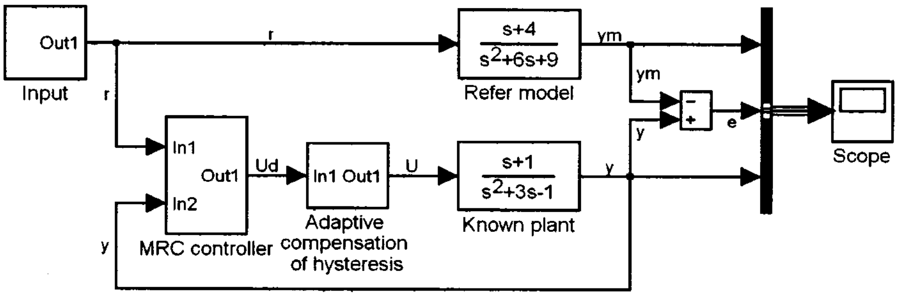


Fig.5.13 Simulation model for MRC of a known unstable linear plant with adaptive compensated input hysteresis

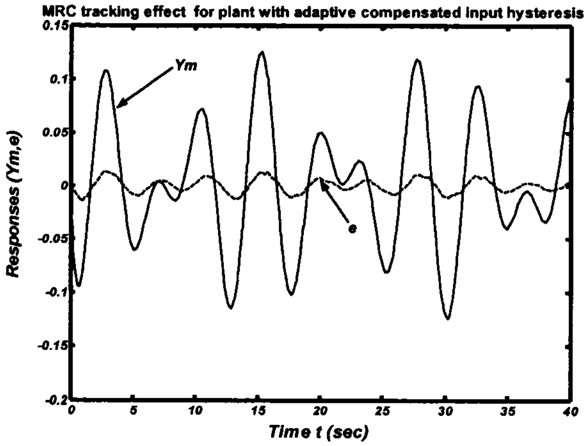


Fig.5.14 MRC tracking effect for a known unstable linear plant with adaptive compensated input hysteresis with $\eta = 2$

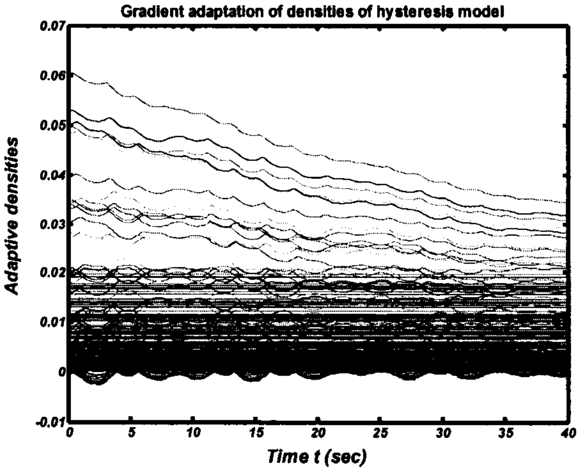


Fig.5.15 Gradient adaptation of parameters of KP hysteresis model with $\eta = 2$

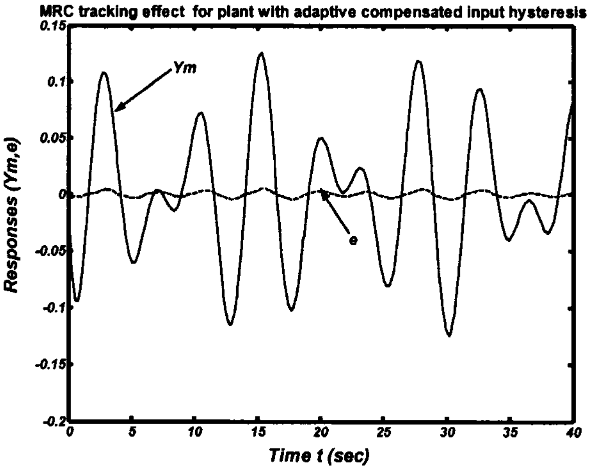


Fig.5.16 MRC tracking effect for a known linear plant
with adaptive compensated input hysteresis with $\eta = 40$

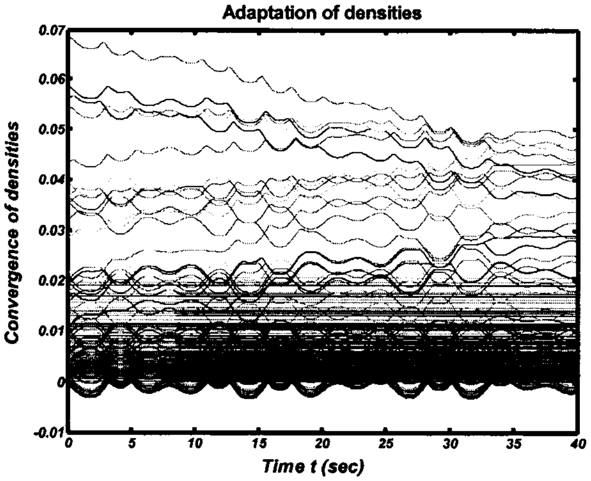


Fig.5.17 Gradient adaptation of parameters of KP hysteresis model with $\eta = 40$

5.4.2 MRC of Known Linear Plant with Non-measurable Input Hysteresis

In this section, the methodology introduced in section 5.3 will be illustrated by application on a linear plant with a known transfer function as

$$y = G_p(s)u = \frac{1}{s^2 + 2s - 8}u \tag{5.62}$$

The plant has a relative degree of 2 and is unstable since it has a positive pole at $s = 2 > 0$. The plant is preceded by a hysteretic actuator which in turn injects nonlinear dynamics into the plant. For this plant, the output of the hysteretic actuator is not measurable.

The control objective is to design a controller to guarantee that the output of the plant (5.62) will track the output of a reference model described by

$$y_m = G_m(s)r = \frac{3}{s^2 + 6s + 9}r. \tag{5.63}$$

If the hysteresis of the actuator can be perfectly modeled, a perfect inverse model could be derived to cancel the hysteresis nonlinearity. Realistically, the parameters of the hysteresis KP model are impossible to be identified exactly, thus, an inaccurate inverse compensation is usually carried out to reduce the hysteresis nonlinearity of the actuator. However, the inaccurate inverse compensation generates disturbances to be injected into the plant, and the disturbance furthermore affects the tracking performance of the MRC for the plant. Due to the unavailable output of the hysteretic actuator, a gradient adaptive inverse hysteresis is modeled based on the measurement of the output error between the linear plant and its reference model. The gradient adaptive inverse hysteresis guarantees that the input error (disturbance) injected into the linear plant asymptotically converges to zero, and furthermore ensures the MRC scheme to still have perfect tracking performance.

In the transfer function of the known linear plant (5.62), $N_p = 1$, $D_p = s^2 + 2s - 8$, and $g_p = 1$. In the model transfer function (5.58), $N_m = 1, D_m = s^2 + 6s + 9$ and $g_m = 3$. Since the plant and the reference model are second order systems with relative degree 2 ($n^* = n - m = 2$), thus, $h(s) = 1$ and $\Lambda(s) = \tilde{\Lambda}(s) = s + 1$ are selected. Also one has $\theta_3 = g_m / g_p = 3$. Substituting all the above variables and values into the model matching condition (5.37) obtains the parameters as $[\theta_3\ \theta_1\ \theta_2\ \theta_0] = [3\ -4\ -36\ -13]$. Substituting these control gains into the model reference controller (5.26) gives

$$u_d(t) = 3r(t) - 4\omega_1(t) - 36\omega_2(t) - 13y(t) \tag{5.64}$$

with the filtered signal $\omega_1 = \frac{h(s)}{\Lambda(s)} u_d = \frac{1}{s+1} u_d$ and $\omega_2 = \frac{h(s)}{\Lambda(s)} y = \frac{1}{s+1} y$. For the following simulations, the signal (5.61) is still selected as the reference command to the MRC schemes.

Case 1: **MRC for known linear plant with exactly compensated input hysteresis**

The simulation model for the MRC to the plant (5.62) without input hysteresis is established as in Fig.5.18, and the simulation result is shown in Fig.5.19.

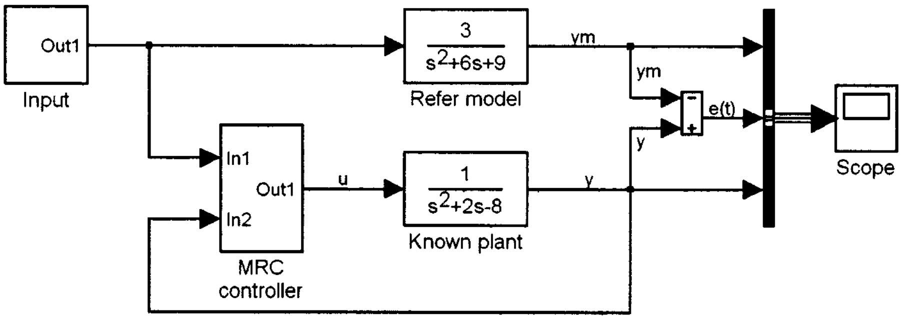


Fig.5.18 Simulation model for MRC to an unstable linear plant with relative degree of 2

From Fig.5.19, one can observe that the tracking error is zero for all simulation time. This is because the control gains in the MRC controller are calculated based on the plant transfer function without any hysteresis, and the MRC controller places the poles of the linear time-invariant plant at the same location as that of the reference model, and adjusts the plant transfer function to the same gain g_m to that of the model.

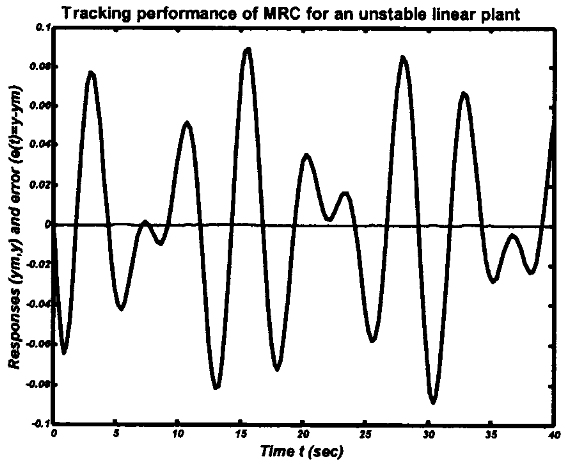


Fig.5.19 Simulation result for MRC to an unstable linear plant with relative degree of 2

Case 2: **MRC for known linear plant with uncompensated input hysteresis**

However, if there is hysteresis existing in the actuator of the plant (5.62), it will introduce hard nonlinearly input error into the plant. To examine the influence of the hysteresis on the tracking effect of the MRC of the linear plant, a simulation model for the plant with an uncompensated hysteresis is established as in Fig.5.20, where the simulation parameters for representing the exact hysteresis is listed in table 5.1, and the simulation result is shown in Fig.5.21.

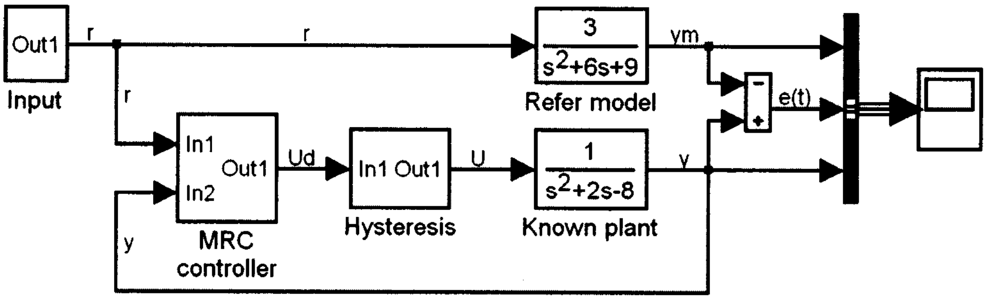


Fig.5.20 Simulation model for MRC to an unstable linear plant with input hysteresis

From Fig.5.21, one can see that the closed loop system with input hysteresis is unstable even though the MRC scheme can guarantee that the unstable linear plant without input hysteresis is stable. The response $y(t)$ of the plant diverges very fast at the beginning of the simulation; and the tracking error goes to infinite. This is because the linear plant with high relative degree $(n^* = n - m \geq 2)$ characterizes more internal dynamics and is more sensitive to any disturbance. Thus, a relative large disturbance can stimulate its dynamics and cause the system to become unstable.

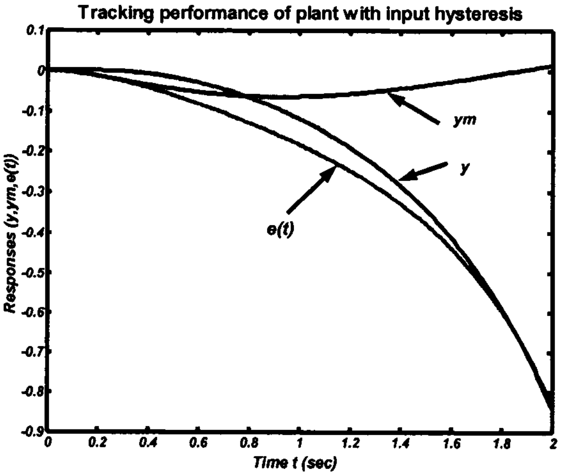


Fig.5.21 Simulation result for MRC to an unstable linear plant with input hysteresis

***Case 3:* MRC for known linear plant with fix compensated input hysteresis**

From the simulation study of case 2, it is known that the MRC scheme can not reject the disturbance from the input, and the disturbance injected by input hysteresis causes the plant with high relative degree to be unstable. These facts motivate the construction of an inverse hysteresis based on modeling of the hysteresis to compensate the hysteresis. However, due to the inaccurate modeling of the hysteresis, the exact inverse hysteresis cannot be found. Therefore, the error caused by inaccurate modeling together with the computation round-off will still inject disturbance into the linear plant. This input error raises some tracking error in the MRC scheme. The simulation model for the MRC of the plant (5.62) with fix compensated input hysteresis is setup as in Fig.5.22, where the simulation parameters for representing the exact hysteresis and the estimated hysteresis are listed in table 5.1. Simulation results are shown in Fig.5.23 and Fig.5.24.

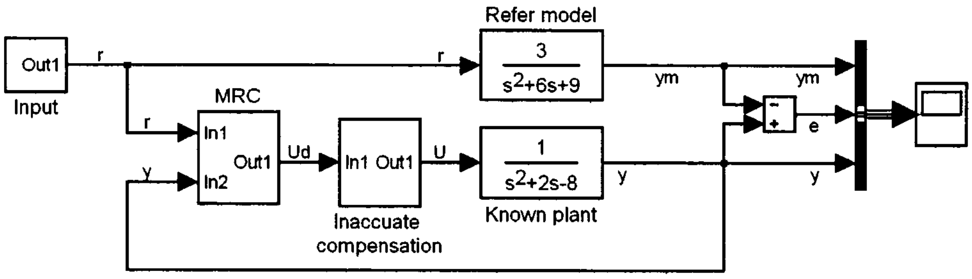


Fig.5.22 Simulation model for MRC to plant with fix compensation for input hysteresis

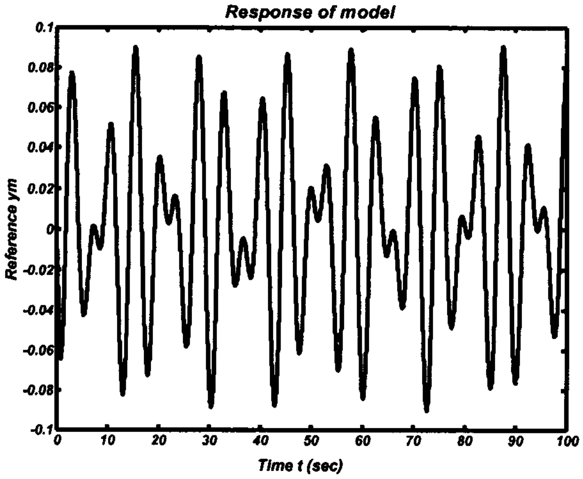


Fig.5.23 Reference response $y_m(t)$

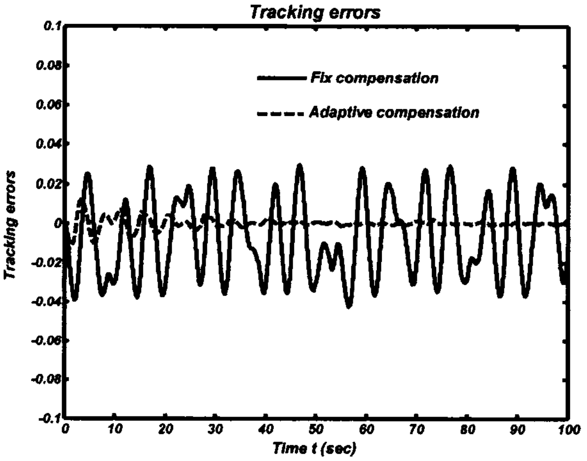


Fig.5.24 Tracking performances of MRC to plant with fix/adaptive compensation for input hysteresis

Fig.5.23 shows the time response of the reference model. In Fig.5.24, the controlled system becomes stable and the tracking error of the MRC with fix compensation of hysteresis still remains large in the entire simulation period. This is caused by the inaccurate estimation of the parameter $\sigma = 1.0$ to the exact $\sigma = 1.8$.

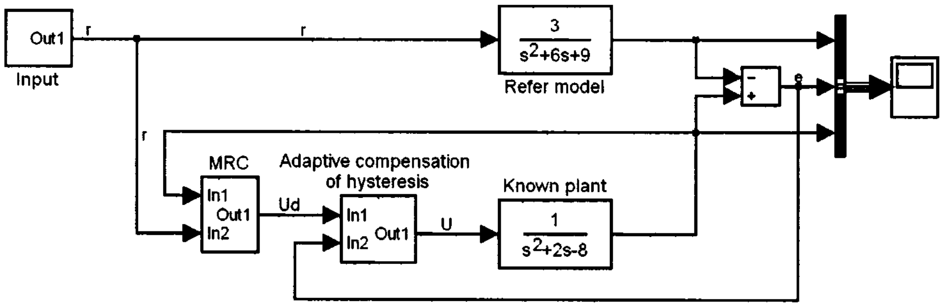


Fig.5.25 Simulation model for MRC to plant with adaptive compensation for input hysteresis

***Case 4:* MRC for known linear plant with adaptive compensated input hysteresis**

To reduce the tracking error and make it asymptotically converge to zero, an adaptive law can be designed to update the parameters of the inverse model according to the output error of the plant. The gradient adaptive method and its convergence have been presented in section 5.3. The simulation model for MRC for the plant (5.62) with adaptive compensation of the hysteresis is established as in Fig.5.25, where the simulation parameters for representing the exact hysteresis and initial estimated hysteresis are listed in table 5.1. The tracking error simulation result with the adaptive gain Z chosen as $Z = 2$ is shown as the dash-line in Fig.5.24; one should observe that the tracking error has been significantly reduced and has the tendency to converge to zero.

5.5 Conclusions

In this chapter, the linearly parameterized KP model has been used to describe the actuator hysteresis, and the corresponding inverse KP model has been used to partly or completely compensate the hysteresis of the actuator. The tracking problem for known linear time-invariant systems with input hysteresis has been solved using the model reference control method. An inner loop gradient adaptive compensation of input hysteresis was used to asymptotically reduce the disturbance injected into the linear systems by the input hysteresis. Choices of MRC controller law and gradient adaptive law for compensation have been derived. Convergence of parameters of the linearly parameterized KP model has been proved.

In the MRC configuration, the MRC controller can only replace the poles of the plant into the stable region to make the unstable plant to become stable, but cannot force the output of the plant to track the output of the model if the plant has any input disturbance due to hysteresis. This means the error injected from the hysteretic actuator is severely disturbing the linear plant.

The inaccurate compensation of hysteresis between the MRC controller and the plant has an effect to reduce the tracking error, but it still cannot satisfy the design requirement because it cannot completely eliminate the hysteresis effects.

If the gradient adaptive hysteresis compensation is constructed as an inner loop of the MRC and is utilized to reduce the hysteresis, relative larger adaptation gains for the parameters of the hysteresis model can greatly reduce the tracking error, and the convergence of the parameters is relatively smooth. If too large an adaptation gain is chosen, the tracking error is reduced almost to zero, but the convergence of the parameters have large oscillations, and this may excite the non-modeled dynamics in the plant and cause the MRC scheme to become unstable.

CHAPTER 6

MODEL REFERENCE ADAPTIVE CONTROL OF UNKNOWN LINEAR SYSTEMS WITH COMPENSATED INPUT HYSTERESIS

In Chapter 5, the model reference control (MRC) method has been presented to achieve the tracking goal for known linear time-invariant systems. In this chapter, a problem of model reference adaptive (MRAC) control for unknown time-invariant linear systems with input hysteresis will be discussed. Due to the output from the hysteretic actuator of the systems which are assumed to be not measurable, the inaccurate compensation of hysteresis through the inverse KP hysteresis model will be used to reduce the disturbance injected into the linear plants. Choices of controller law and adaptive law will be derived for plants with relative degree of one or more. Convergence of tracking and adaptation of parameters will be proved.

6.1 System Description and Control Objective

6.1.1 System Description

Consider an unknown linear time-invariant plant with input hysteresis from the actuator (see Fig.6.1). The linear plant to be controlled is described by

$$y = G_p(s)u = g_p \frac{N_p(s)}{D_p(s)} u \tag{6.1}$$

where $u(t) \in R$ and $y(t) \in R$ are the plant input and output, respectively; $N_p(s)$, $D_p(s)$ are monic polynomials of degrees n and m, respectively; g_p is a constant scalar gain.

The hysteresis nonlinearity of the actuator can be denoted in operator form as

$$u(t) = H[v](t), \tag{6.2}$$

where $v(t) \in R$ is the input of the actuator and $u(t) \in R$ denotes its output. The operator $H(\cdot)$ is the KP model to describe the hysteresis characteristics of the actuator, which has been discussed in detail in Chapter 4. The operator $H(\cdot)$ can be expressed in linearly parameterized form as

$$u(t) = H(v(t)) = \Gamma^T K, \tag{6.3}$$

where the $N \times 1$ vector Γ consists of densities of N KP operators, i.e., $\Gamma = [\mu_1, \mu_2, \ldots, \mu_N]^T$, and the $N \times 1$ vector K includes output values of N KP operators as they are subjected to an input $v(t)$.

The assumptions for the unknown linear time-invariant plant and the hysteretic actuator are:

(1) $G_p(s)$ is minimum phase;

(2) the relative degree $n^* = m - n$ of $G_p(s)$ is known;

(3) the degree of pole $D_p(s)$ is known as m;

(4) the sign of g_p is known.

(5) the output $u(t)$ of the hysteretic actuator is unavailable for measurement;

(6) the densities Γ of the KP model used to describe actuator hysteresis are not exactly known, but their upper and lower boundaries and initial estimations are known.

6.1.2 Control Objective

The control objective is to design a feedback control $v(t)$ so that all closed-loop signals are bounded and the plant output $y(t)$ tracks output $y_m(t)$ of a reference model:

$$y_m = G_m(s)u = g_m \frac{N_m(s)}{D_m(s)} r \tag{6.4}$$

where $G_m(s)$ is a stable rational transfer function of relative degree $n^* = m - n$, the gain g_m is assumed positive without loss of generality, and the r is a bounded piecewise continuous signal.

As the linear time-invariant plant is unknown, the model reference adaptive control method will be applied to achieve the tracking goal while the hysteresis nonlinearity of the actuator is absent. With the presence of actuator hysteresis, an inaccurate compensation is necessary in the MRAC configuration to reduce the disturbance injected into the linear plant (see Fig.6.2).

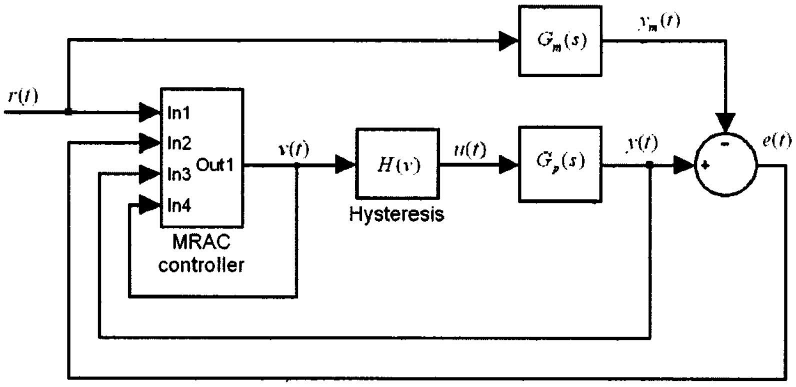


Fig.6.1 Tracking control scheme for unknown linear systems with input hysteresis

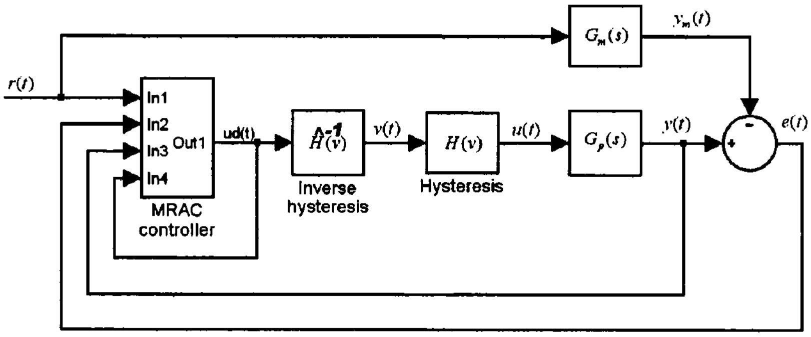


Fig.6.2 MRAC configuration of an unknown linear time-invariant system with inaccurate inverse compensation of input hysteresis

6.2 Model Reference Adaptive Controller Design for Unknown Linear Systems with Relative Degree One

6.2.1 Model reference adaptive control law

In this section, it is assumed that the linear time-invariant plant that was introduced in section 5.1 is unknown. The objective is similar, that is, to design a controller to ensure that the output of an unknown plant will track the output of a reference model.

In this adaptive control problem, since the plant parameters are unknown, thus the ideal control parameters are unknown. Instead of a control law with ideal control parameters as in (5.26) or (5.28), the ***adaptive control law*** is chosen to be

$$u(t) = \hat{\theta}_3 r(t) + \hat{\theta}_1 \omega_1(t) + \hat{\theta}_2 \omega_2(t) + \hat{\theta}_0 y(t) \tag{6.5}$$

where $\hat{\theta}_3(t)$, $\hat{\theta}_1(t)$, $\hat{\theta}_2(t)$, $\hat{\theta}_0(t)$ are adaptive control parameters to be provided by the adaptation law. The initial values of these parameters are estimated through experiment or simulation. This adaptive control law (6.5) results in an output of the unknown plant as

$$y(t) = G_p(s)u \tag{6.6}$$

The MRAC configuration is shown as Fig.6.3.

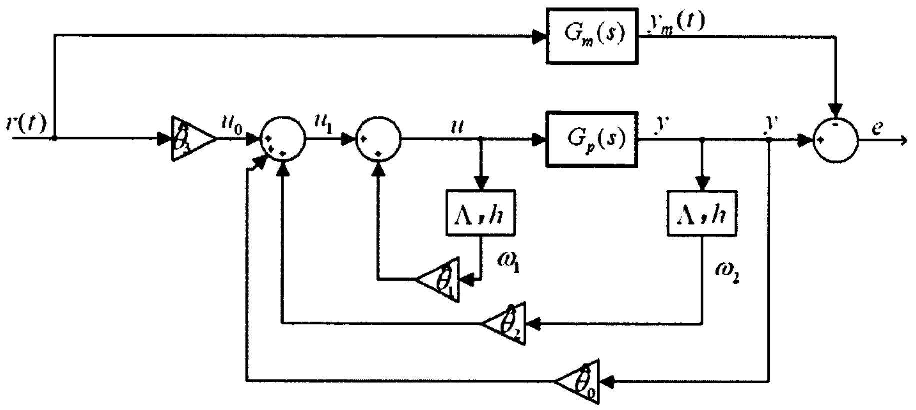


Fig.6.3 Model reference adaptive control of a unknown linear plant

6.2.2 Choice of Adaptation Law

To simplify the notation, let $\hat{\theta}(t)$ be the $2n \times 1$ vector containing all the controller parameters, and $\omega(t)$ be the $2n \times 1$ vector containing the corresponding signals, i.e.,

$$\hat{\theta}(t) = [\hat{\theta}_3(t) \quad \hat{\theta}_1(t) \quad \hat{\theta}_2(t) \quad \hat{\theta}_0(t)]^T \tag{6.7}$$

$$\omega(t) = [r(t) \quad \omega_1(t) \quad \omega_2(t) \quad y(t)]^T \tag{6.8}$$

Then the adaptive control law (6.5) can be written in compact form as

$$u(t) = \hat{\theta}^T(t)\omega(t) \tag{6.9}$$

Denoting the ideal value of $\hat{\theta}(t)$ by $\theta(t)$ and the error between $\hat{\theta}(t)$ and $\theta(t)$ by $\phi(t) = \hat{\theta}(t) - \theta(t)$, the estimated parameters can be represented as

$$\hat{\theta}(t) = \theta(t) + \phi(t) \tag{6.10}$$

where the parameter errors is written in a vector form as

$$\phi(t) = [\phi_3(t) \quad \phi_1(t) \quad \phi_2(t) \quad \phi_0(t)]^T \tag{6.11}$$

Therefore the adaptive control law (6.9) can also be written as

$$u(t) = \theta^T(t)\omega(t) + \phi^T(t)\omega(t) \tag{6.12}$$

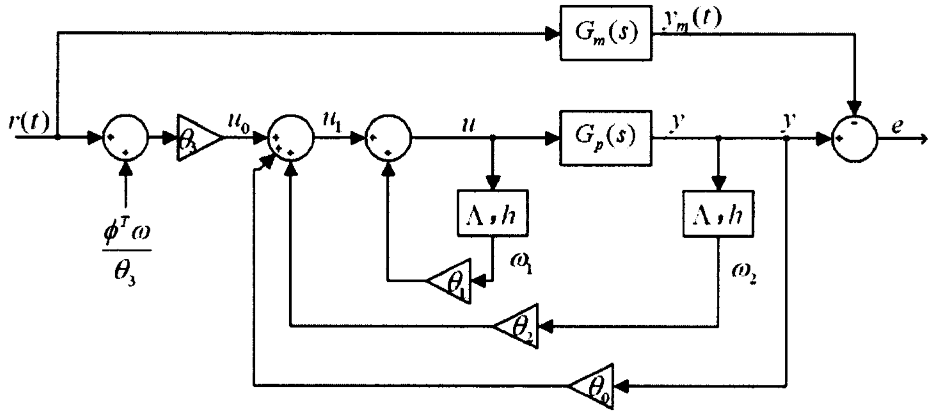


Fig.6.4 An equivalent configuration of an MRAC for an unknown linear plant

In order to choose an adaptation law so that the tracking error $e(t)$ converges to zero, it is necessary to investigate how the tracking error is related to the parameter error. With the adaptive control law (6.12), the MRAC system can be equivalently represented as shown in Fig.6.4 with $\frac{\phi^T \omega}{\theta_3}$ regarded as an external signal. Thus, the output of controlled plant is

$$y(t) = G_m(s)r + G_m(s)[\phi^T(t)\omega(t)/\theta_3] \tag{6.13}$$

with the desired output expressed as

$$y_m(t) = G_m(s)r \tag{6.14}$$

and the tracking error is related to the parameter error as

$$e(t) = y(t) - y_m(t) = G_m(s)[\phi^T(t)\omega(t)/\theta_3] \tag{6.15}$$

Since the tracking error (6.15) is the familiar equation seen in *Lemma 8.1* of reference [25], the following adaptive law is chosen

$$\dot{\hat{\theta}}(t) = \dot{\phi}(t) = -\mathrm{sgn}(g_p)\gamma e(t)\omega(t) \tag{6.16}$$

where γ is a positive number representing the adaptation gain, and the sign of θ_3 is the same as that of g_p due to the assumed positive sign of g_m.

In order to prove that the tracking error in the adaptive control system (6.16) converges to zero asymptotically, the following ***Lemma*** [25] is introduced.

Lemma: *Consider two signals $e(t)$ and $\phi(t)$ related by the following dynamic* equation

$$e(t) = H(s)[k\phi^T(t)v(t)] \tag{6.17}$$

where $e(t)$ is a scalar output signal. $H(s)$ is a strictly positive real transfer function. k is an unknown constant with known sign, $\phi(t)$ is an $m \times 1$ vector function of time, and $v(t)$ is a measurable $m \times 1$ vector. If the vector ϕ varies according to

$$\dot{\phi}(t) = -\operatorname{sgn}(k)\gamma\, e\, v(t) \tag{6.18}$$

with γ being a positive constant, then $e(t)$ and $\phi(t)$ are globally bounded. Furthermore, if v is bounded, then $e(t) \to 0$ as $t \to \infty$.

Proof: let the sate-space representation of (6.17) be

$$\begin{cases} \dot{X} = AX + b[k\phi^T v] \\ e = c^{\,T} X \end{cases} \tag{6.19}$$

Since $H(s)$ is SPR (strictly positive real transfer function), it follows from the ***Kalman-Yakubovich lemma*** stated in ***Appendix 5,*** that given a symmetric positive definite matrix Q, there exists another symmetric positive definite matrix P such that

$$Pb = c \tag{6.20}$$

and

$$PA + A^T P = -Q \tag{6.21}$$

Let V be a positive definite function of the form

$$V(X,\phi) = X^T PX + \frac{|k|}{\gamma}\phi^T\phi \geq 0 \tag{6.22}$$

Its time derivative along the trajectories of the system defined by (6.18) and (6.19) is

$$\begin{aligned} \dot{V} &= \dot{X}^T PX + X^T P\dot{X} + \frac{|k|}{\gamma}\dot{\phi}^T\phi + \frac{|k|}{\gamma}\phi^T\dot{\phi} \\ &= X^T A^T PX + b[k\phi^T v]^T PX + X^T PAX + X^T Pb[k\phi^T v] \\ &\quad - kev^T\phi - ke\phi^T v \end{aligned}$$

$$= X^T(A^TP + PA)X + 2X^TPb[k\phi^Tv] - 2\phi^T(kev) \quad (6.23)$$

Consider (6.19), (6.20) and (6.21), (6.23) can be rewritten as

$$\dot{V} = -X^TQX + 2X^Tc[k\phi^Tv] - 2\phi^T(kev)$$

$$= -X^TQX + 2ek\phi^Tv - 2\phi^T(kev)$$

$$= -X^TQX \leq 0 \quad (6.24)$$

Therefore, the system defined by (6.17) and (6.18) is globally stable. The equations (6.22) and (6.24) also imply that e and ϕ are globally bounded.

From (6.19), one has that $\dot{X}$ is bounded if the input $v(t)$ signal is bounded. This implies the uniform continuity of $\dot{V}$ since there is a bound of its derivative

$$\ddot{V} = -2XQ\dot{X}. \quad (6.25)$$

Application of ***Barbalat's lemma*** (see ***Appendix 5***) then indicates the asymptotic convergence of $\dot{V}$ to zero. Since Q is a positive defined matrix, the asymptotic convergence of $\dot{V}$ to zero as $t \to \infty$ leads $X \to 0$ as $t \to \infty$, Furthermore, results in $e \to 0$ as $t \to \infty$ since $e = c^TX$.

Based on the above ***Barbalat's lemma*** and through a straightforward procedure for establishing signal boundedness, one can see that the tracking error in the adaptive control system (6.16) converges to zero asymptotically.

6.3 MRAC Controller Design for Unknown Linear Systems with Higher Relative Degree

In the above section, the adaptive controller design for plants with relative one degree has been presented. In this section, the design of adaptive controller for plants with relative degree larger than one will be discussed. For the choice of control law, there is no difference from that for the plants with relative degree one, but the choice of the adaptive law is very different, which comes from the fact that the reference model now cannot be SPR.

Choice of adaptive law

When the plant parameters are unknown, the adaptive control law (6.9) can be used again. i.e.,

$$u(t) = \hat{\theta}^T(t)\omega(t) \tag{6.26}$$

with the $2n$ controller parameters in $\hat{\theta}(t)$ provided by the adaptation law. Using similar reasoning as before, the output y and the tracking error can be obtained again in the form of (6.13) and the form of (6.15), respectively. Therefore, one has

$$e(t) = G_m(s)[\phi^T(t)\omega(t)/\theta_3] \tag{6.27}$$

However, the choice of adaptation law given by (6.16) cannot be used, because now the reference model transfer function $G_m(s)$ is no longer SPR. In order to find an adaptation law, an augmented error is introduced as

$$\varepsilon(t) = e(t) + \alpha(t)\eta(t) \tag{6.28}$$

where $\alpha(t)$ is a time-varying parameter to be determined by adaptation, and $\alpha(t)$ is not a controller parameter, but only a parameter used in forming the new augmented error; $\eta(t)$ is an auxiliary error defined as

$$\eta(t) = \hat{\theta}^T(t)G_m(s)\omega(t) - G_m(s)[\hat{\theta}^T(t)\omega(t)] \tag{6.29}$$

which is shown in Fig.6.5. $\eta(t)$ can be computed on-line since the estimated parameter vector $\hat{\theta}(t)$ and the signal vector $\omega(t)$ are both available. When the estimated vector $\hat{\theta}(t)$ is replaced by its true parameter vector θ, the auxiliary error $\eta(t)$ becomes zero as

$$\eta(t) = \theta^T G_m(s)\omega(t) - G_m(s)[\theta^T\omega(t)] = 0$$

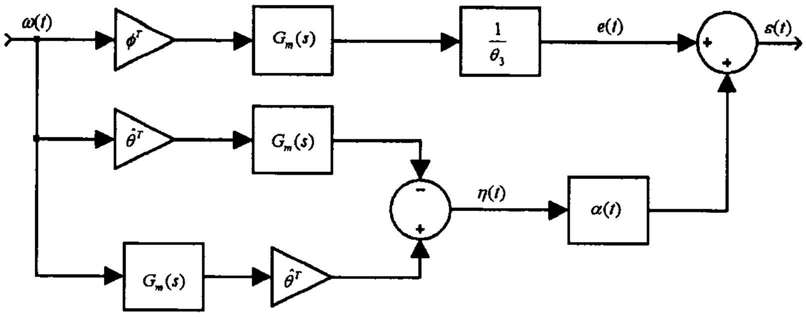


Fig.6.5 The augmented error

Defining $\alpha(t) = \frac{1}{\theta_3} + \phi_\alpha(t)$ and substituting (6.27) and (6.29) into (6.28) yield

$$\varepsilon(t) = \frac{1}{\theta_3}\phi^T(t)\varsigma(t) + \phi_\alpha(t)\eta(t) \tag{6.30}$$

where $\varsigma(t) \overset{\Delta}{=} G_m(s)\omega(t)$. This means that the augmented error $\varepsilon(t)$ is linearly parameterized by the parameter errors $\phi(t)$ and $\phi_\alpha(t)$. The above equation (6.30) represents a problem of system identification. The gradient method with normalization can be used to find the controller parameters and the parameter for forming the augmented error are updated by the adaptation law as

$$\begin{cases} \dot{\hat{\theta}}(t) = -\dfrac{\operatorname{sgn}(g_p)\gamma\varepsilon\varsigma}{1+\varsigma^T\varsigma} \\ \dot{\alpha}(t) = -\dfrac{\gamma\varepsilon\eta}{1+\varsigma^T\varsigma} \end{cases} \tag{6.31}$$

with

$$\eta(t) = \hat{\theta}^T(t)\varsigma(t) - G_m(s)[\hat{\theta}^T(t)\omega(t)],$$

$$\varsigma(t) = G_m(s)\omega(t),$$

$$\varepsilon(t) = \frac{1}{\theta_3}\phi^T(t)\varsigma(t) + \phi_\alpha(t)\eta(t),$$

and $$\phi_a(t) = \alpha(t) - \frac{1}{\theta_3}.$$

With the control law (6.26) and the adaptation law (6.31), global convergence of the tracking error can be guaranteed. The proof is omitted here due to much mathematical derivation involved [25].

6.4 Simulation Studies

In this section, the above methodology will be illustrated by applying it on a simple time-invariant linear plant with a transfer function as (5.59). The plant is unstable since it has a positive pole At $s = (3+\sqrt{13})/2 > 0$. The function (5.59) is given for representing the plant for simulation, but it cannot be used to design an MRC controller as explained in Chapter 5. Thus, the plant is assumed unknown since the poles, zero and gain of the plant are not exactly known, but only have some information about the sign of the gain, the order of $D_p(s)$ and the relative degree. The plant is preceded by a hysteretic actuator which in turn injects nonlinearity into the plant.

The control objective is to design a controller to force the output of the plant (5.59) to track the output of a reference model described by (5.60).

If the hysteresis of the actuator has been perfectly modeled, a perfect inverse model could be derived to cancel the nonlinearity from the actuator ($u = H(H^{-1}(u_d)) = u_d$). In this way, the tracking problem becomes the design of an MRAC configuration for an unknown linear time-invariant plant with linear input. However, the parameters of the hysteresis KP model are impossible to be identified exactly; thus, an inaccurate inverse compensation is usually carried out to reduce the hysteresis nonlinearity of the actuator. This inaccurate compensation of hysteresis in turn injects disturbance

($u = H(\hat{H}^{-1}(u_d)) = u_d + d(t)$) into the plant. Usually, the output of the actuator is not measurable if the hysteretic actuator is connected to the plant. Even if the output of actuator is measurable the sensor is a very expensive component of the control system, thus for simplicity of the system, the output of the actuator is still considered as unknown since the sensor is omitted. Thus the method of gradient adaptive control cannot be used to cancel the hysteresis. Fortunately, the MRAC method can reject some disturbance injected from the system input. This motivates the design of an MRAC system to control the unknown linear system with an inaccurate compensation of input hysteresis, rather than to design a gradient adaptive controller to cancel the hysteresis.

The initial parameters of the MRAC scheme are all assumed zero because there are not any other information about the plant except some assumptions stated in section 6.1. The parameters are calculated by the updating law (6.16), where the filtered signals ω_1 and ω_2 are still chosen as

$$\omega_1 = \frac{1}{s+4}u \quad \text{and} \quad \omega_2 = \frac{1}{s+4}y$$

since the transfer function of the model has only a zero at $s = -4$ with relative degree as $n^* = n - m = 2 - 1 = 1$. Since the sign of the plant gain is positive, $\mathrm{sgn}(g_p) = 1$. The adaptive gains are chosen upon the simulation effects in order to make the tracking error quickly approach zero and to avoid the MRAC gains to not have severe oscillations. This is because severe oscillation of MRAC gains introduces much dynamics into the system and in turn excites the unmodeled dynamics of the plant.

Case 1: First, assume that the hysteresis of the actuator is completely cancelled by its inverse hysteresis model, and then design an MRAC scheme for the output tracking problem of the unknown plant. In this situation, the output of the MRAC controller is directly connected to the input of the plant because $u = H(H^{-1}(u_d)) = u_d$. The simulation model for this case is shown in Fig.6.6. As the input command applied to the MRAC system is chosen as $r = -0.16[\cos(\pi t/3) + \cos(\sqrt{2}\pi t/3)]$, and the adaptive gains are chosen as $\gamma_1 = 10, \gamma_2 = 200, \gamma_3 = 200, \gamma_4 = 100$, the corresponding simulation results are shown in Fig.6.7, Fig.6.8 and Fig.6.9.

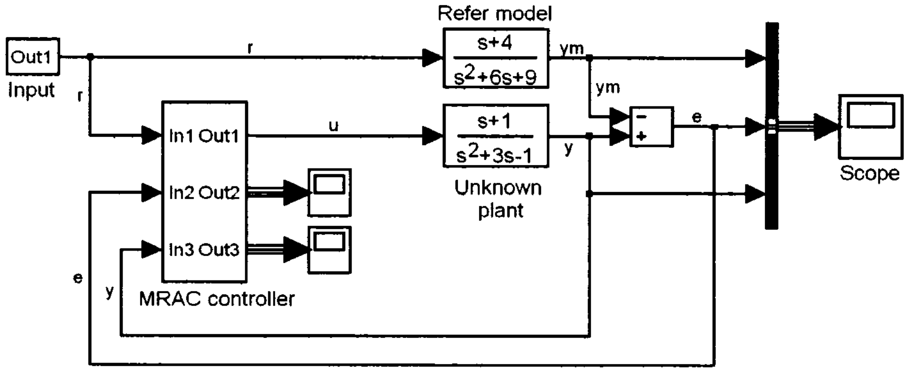


Fig.6.6 Simulation diagram for MRAC of an unknown linear plant

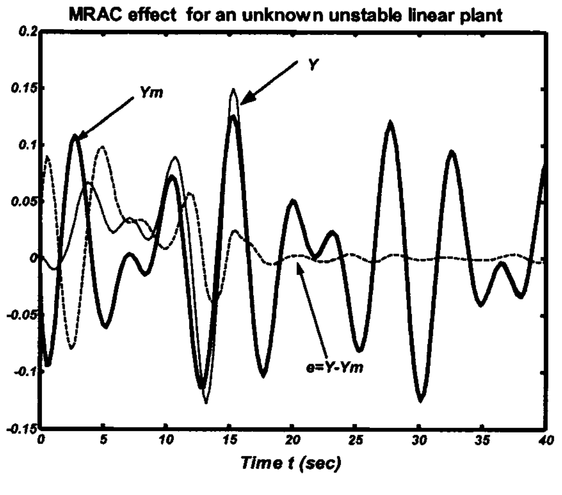


Fig.6.7 Tacking effect of MRAC for an unknown unstable linear plant

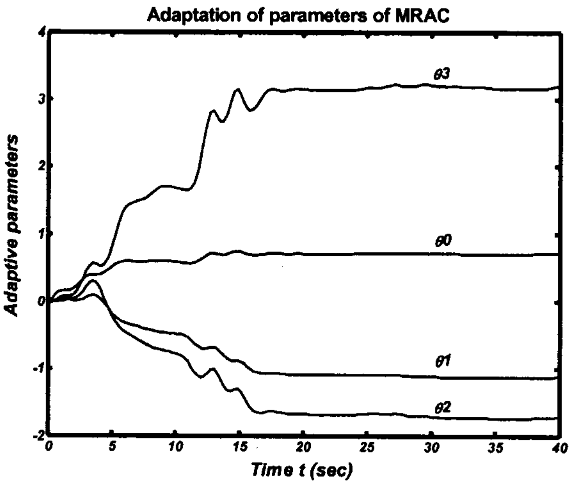


Fig.6.8 Convergences of adaptive parameters of MRAC for an unknown unstable linear plant

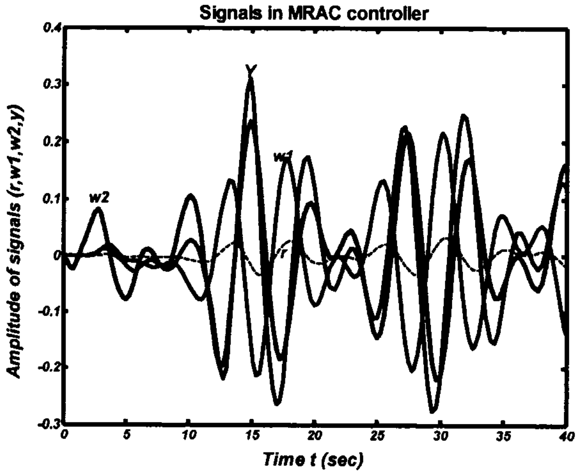


Fig.6.9 Signals in MRAC controller

Fig.6.7 shows that the output of the unknown linear plant perfectly tracks the desired output of the reference model after about 3 periods of input (15seconds). Also, from Fig.6.8, it can be observed that four MRAC gains converge to stable values after about 15 seconds. They are all initialized at zero because there is no any information about them. In the first 15 seconds, four MRAC gains promptly approach their stable values with some gentle oscillation. The relative larger adaptive gains $[\gamma_1, \gamma_2, \gamma_3, \gamma_4]$ are chosen, the faster the MRAC gains reach their stable values and the earlier the tracking error approaches zero. Too large a value of adaptive gains should not be chosen because it causes the MRAC gains to oscillate too much to be stable even if it can make the tracking error quickly approach zero. From Fig.6.9, one can see that the signals in the MRAC controller have reasonable multitudes for the choice of the adaptive gains as $\gamma_1 = 10, \gamma_2 = 200, \gamma_3 = 200, \gamma_4 = 100$.

Case 2: To reveal the hysteresis effects in the MRAC configuration, the hysteresis will not been compensated. In this case, the output of the MRAC controller is directly offered to the hysteresis, and then the hysteresis produces the input signal for the unknown plant. The simulation diagram is shown in Fig.6.10. As the input command is applied to the closed-loop system, the adaptive gains and filtered signals are selected as the same as in simulation case 1, and the parameters for modeling the hysteresis is still chosen as given in table 5.1; the simulation results are shown in Fig.6.11 and Fig.6.12.

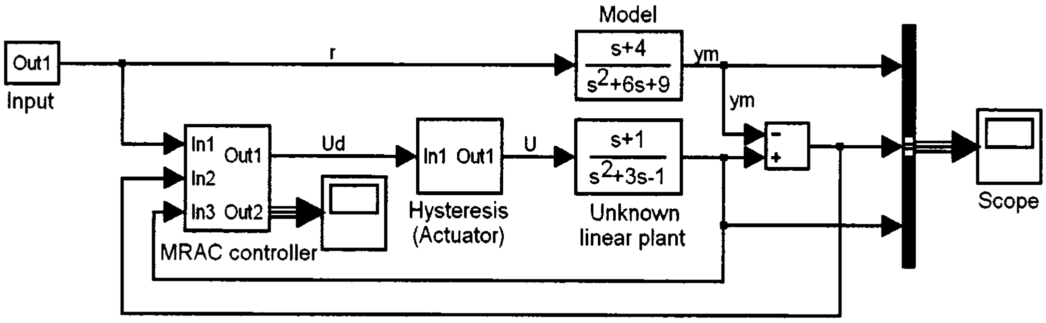


Fig.6.10 Simulation model for MRAC with input hysteresis of an unknown unstable linear plant

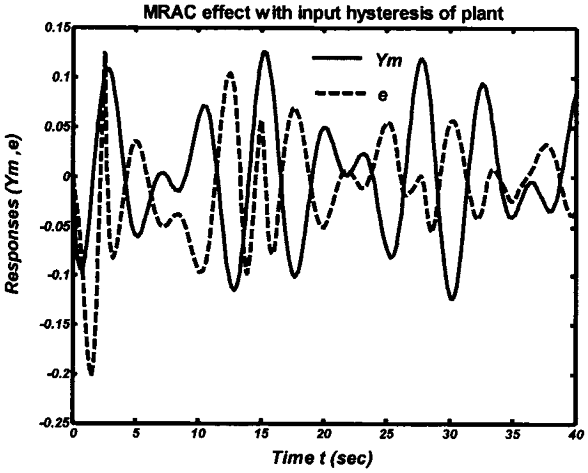


Fig.6.11 Tacking effect of MRAC (with input hysteresis of an unknown unstable linear plant)

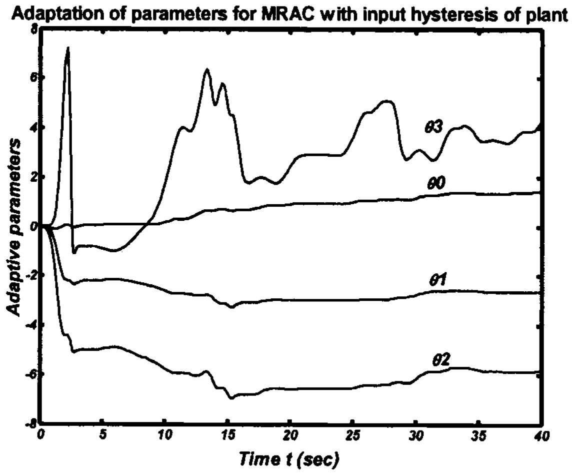


Fig.6.12 Adaptation of parameters
(MRAC with input hysteresis of an unknown unstable linear plant)

As shown Fig.6.11, the output of the unknown linear plant hardly tracks the desired output of the reference model during the entire simulation period. Also, from Fig.6.12, it is noticed that some MRAC gains cannot approach to certain stable values, and there is much oscillation of the MRAC gains. This means that the MRAC configuration cannot reject completely the disturbance injected from the hysteresis. Therefore, an inverse compensation to the hysteresis is necessary.

Case 3: Through the modeling of the actuator hysteresis, some of its properties can be investigated. Thus, an inverse hysteresis model can be designed to partly reduce the hysteresis effect. The precision of the hysteresis model is a determining factor upon the error $d(t) = u_d - H(\hat{H}^{-1}(u_d)) = u_d - u$. In this case, an inverse hysteresis model between the MRAC controller and the hysteretic actuator is added in the simulation diagram of case 2 to reject the remained disturbance. The simulation diagram is shown in Fig.6.13, and the simulation results are shown in Fig.6.14 and Fig.6.15. The simulation parameters for the hysteresis and the inverse hysteresis are chosen as those given in table 5.1, and the adaptive gains and filtered signals remain the same as in simulation case 1.

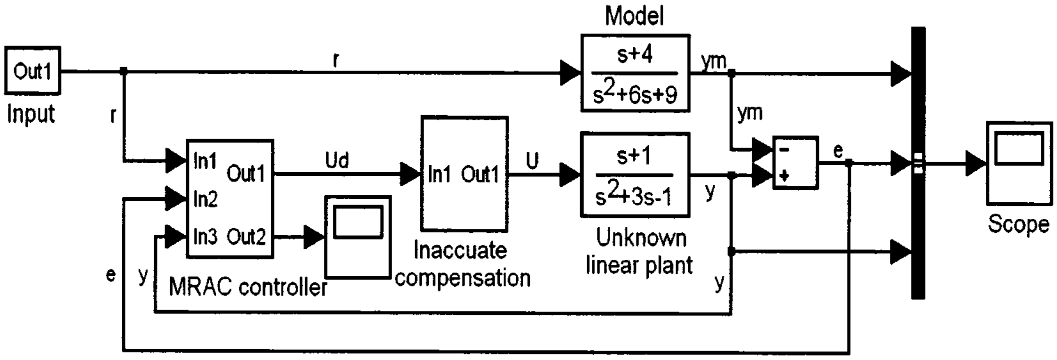


Fig.6.13 Simulation model for MRAC with inaccurate input hysteresis compensation of an unknown unstable linear plant

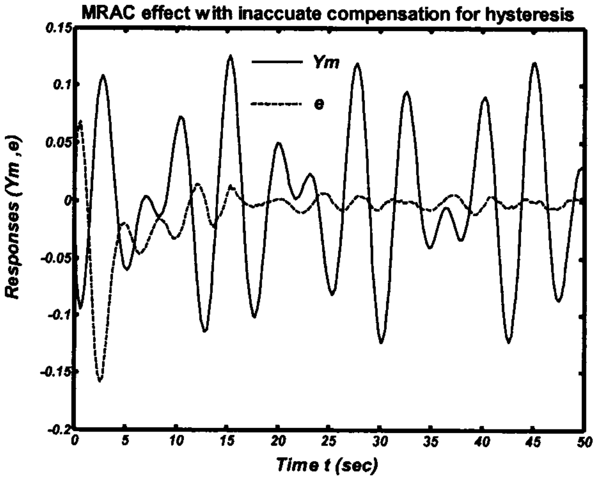


Fig.6.14 Tacking effect of MRAC with inaccurate input hysteresis compensation of an unknown unstable linear plant

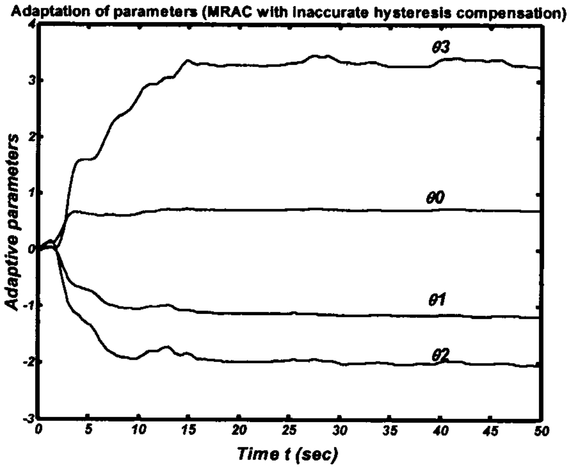


Fig.6.15 Adaptation of parameters of MRAC with inaccurate input hysteresis compensation of an unknown unstable linear plant

From Fig.6.14, the output of the unknown linear plant perfectly tracks the desired output of the reference model after about 2 periods of input (10 seconds). Also, as observed from Fig.6.15, four MRAC gains converge to stable values after about 10 seconds. In the first 10 seconds, four MRAC gains promptly approach their stable values smoothly. If relative larger adaptive gains $[\gamma_1, \gamma_2, \gamma_3, \gamma_4]$ are chosen, the tracking effect will be better. This means that combining with the relative accurate compensation of the hysteresis, the MRAC configuration can perfectly reject the disturbance injected from the hysteresis and also acheives perfect tracking.

6.5 Conclusions

From the above simulations, some conclusions can be drawn as follows: values of adaptive gains determine the approaching speed of the MRAC gains to their stable values, and also affect their approaching oscillation. For relatively larger adaptive gains selected, the faster the MRAC gains reach their stable values and the earlier the tracking error approaches zero. Too large a value of adaptive gains should not be chosen because it causes the MRAC gains to oscillate too much to be stable even if it can make the tracking error quickly approach zero. The MRAC configuration cannot by itself completely reject the disturbance injected from the hysteresis. Combining with the relatively accurate compensation of the hysteresis the MRAC configuration can perfectly reject the disturbance injected from the hysteresis and also achieves perfect tracking.

CHAPTER 7

ROBUST ADAPTIVE CONTROL OF NONLINEAR SYSTEMS WITH UNKNOWN INPUT HYSTERESIS

7.1 Problem Statement

Consider a nonlinear plant with a preceding hysteretic actuator which generates input signal for the plant. The hysteresis of the actuator can be expressed in an operator form as

$$u(t) = H[v](t) \tag{7.1}$$

where $v(t)$ denotes the input of the actuator, $u(t)$ represents its output. The nonlinear plant is described in canonical form as

$$x^{(n)}(t) + \sum_{i=1}^{r} a_i y_i(x(t), \dot{x}(t), ..., x^{(n-1)}(t)) = bu(t) \tag{7.2}$$

where $X = [x, \dot{x}, ..., x^{(n-1)}]^T$ is the state vector which is assumed measurable, y_i are known continuous nonlinear functions of the states and time, and the parameters a_i and the control gain b are unknown constant. Without losing generality, it is commonly to assume $b > 0$. The canonical form expression (7.2) describes a class of nonlinear systems since some general nonlinear systems can be transformed into this structure [27].

The control objective is to design a control law for $v(t)$ in (7.1), to ensure the plant state vector X to track a specified desired trajectory $X_d = [x_d, \dot{x}_d, ..., x_d^{(n-1)}]^T$, i.e., $X \rightarrow X_d$ as $t \rightarrow \infty$.

7.2 Adaptive Controller Design Using Compensator of KP Model

In case of the difficulties and estimation inaccuracies to model the actuator hysteresis in the control system, controlling the system through constructing an inverse hysteresis to compensate for the actuator hysteresis effects is a challenging task. Instead, the development of a direct control method by using currently available robust adaptive control techniques together with the properties of the hysteresis KP model is attempted in this section.

Different from some inverse compensation methods, which are adopted frequently in the literatures, a robust adaptive controller is proposed in this section to control a nonlinear plant as expressed by equation (7.2), which is preceded by an actuator with hysteresis nonlinearity described by the KP model of equation (7.1), so that the controlled system has global stability and the plant states track a specified desired trajectory within a desired precision.

By recalling equation (4.53), the hysteresis nonlinearity of the actuator described by the KP model can be expressed as

$$u(t) = cv(t) - \Gamma^T M + d(v(t)) \tag{7.3}$$

where $c = \| \Gamma \|$, and $d(v(t))$ is the modeling error of hysteresis in representation of the linearly parameterized KP mold. If the number of dividing lines increases to extremely large value, the term $d(v(t))$ can be considered as zero.

Substituting the hysteresis model (7.3) into the dynamic system (7.2) gives

$$x^{(n)}(t) + \sum_{i=1}^{r} a_i y_i(x(t), \dot{x}(t), \ldots, x^{(n-1)}(t)) = bcv(t) - b\Gamma^T M + bd(v(t)), \tag{7.4}$$

Dividing both sides of (7.4) by bc and rearranging terms results in

$$\frac{1}{bc} x^{(n)}(t) + \sum_{i=1}^{r} \frac{a_i}{bc} y_i(x(t), \dot{x}(t), \ldots, x^{(n-1)}(t)) + \frac{1}{c}\Gamma^T M = v(t) + \frac{1}{c} d(v(t)). \tag{7.5}$$

For the development of a robust adaptive control law, the following definitions and assumptions regarding the plant and the hysteresis are made:

$$\phi \overset{\Delta}{=} 1/(bc), \tag{7.6}$$

$$\Theta \overset{\Delta}{=} [\theta_1, \theta_2, ..., \theta_i, ..., \theta_r]^T \in R^r \text{, where } \theta_i \overset{\Delta}{=} a_i/(bc) \tag{7.7}$$

$$Y \overset{\Delta}{=} [y_1(X), y_2(X), ..., y_i(X), ..., y_r(X)]^T \in R^r \tag{7.8}$$

and
$$\Gamma_n \overset{\Delta}{=} \Gamma/c, \tag{7.9}$$

where Γ_n is a vector consisting of normalized densities of hysteresis as

$$\mu_n(p) \overset{\Delta}{=} \mu(p)/c = \mu(p)/\|\Gamma\|.$$

Substituting all these definitions (7.6) ~ (7.9) into (7.5) yields

$$\phi x^{(n)}(t) + \Theta^T Y + \Gamma_n{}^T M = v(t) + \frac{1}{c} d(v(t)). \tag{7.10}$$

For the development of a robust adaptive control law, the following definitions and assumptions are made:

Define the tracking error vector as $\widetilde{X}(t) = X(t) - X_d(t)$ (7.11)

and a filtered tracking error as $s(t) = (\frac{d}{dt} + \lambda)^{(n-1)} \widetilde{x}(t),\ \lambda > 0$ (7.12)

which can be rewritten as $s(t) = \Lambda^T \widetilde{X}(t)$ with $\Lambda^T = [\lambda^{(n-1)}, (n-1)\lambda^{(n-2)}, ..., 1]$.

Remark: It has been shown in [25] that the definition given in (7.12) has the following properties:

i) Equation $s(t) = 0$ defines a time-varying sliding surface in R^n, on which the tracking error vector $\widetilde{X}(t)$ decays exponentially to zero;

ii) If $\widetilde{X}(t_0)=0$ and $|s(t)| \le \varepsilon$, where ε is a positive constant, then

$$\widetilde{X}(t) \in \Omega_\varepsilon \overset{\Delta}{=} \{\widetilde{X}(t) \big| \ |\widetilde{X}_i| \le 2^{i-1}\lambda^{i-n}\varepsilon, i = 1, ..., n\} \text{ for } \forall t \ge 0;$$

iii) If $\widetilde{X}(0) \neq 0$ and $|s(t)| \leq \varepsilon$, then $\widetilde{X}(t)$ will converge to Ω_ε within a constant a time $(n-1)/\lambda$.

Rather than use the filtered error $s(t)$ to derive the adaptive law, a tuning error s_ε is introduced as

$$s_\varepsilon = s - \varepsilon sat(\frac{s}{\varepsilon}), \tag{7.13}$$

where ε is an arbitrary positive constant and $sat(\bullet)$ is the saturation function. The tuning error $s_\varepsilon = 0$ when the filtered error $s \leq \varepsilon$.

Assumption 1: The desired trajectory $X_d = [x_d, \dot{x}_d, ..., x_d{}^{(n-1)}]^T$ is continuous and available. Furthermore, $[X_d{}^T, x_d^n]^T \in \Omega_d \subset R^{n+1}$ with Ω_d a compact set;

Assumption 2: There exist known constants $0 < b_{\min} \leq b_{\max}$ such that the control gain b in (7.2) satisfies $b \in [b_{\min}, b_{\max}]$;

Assumption 3: $\Theta \in \Omega_\Theta \overset{\Delta}{=} \{\Theta |\ \theta_{\min} \leq \theta_i \leq \theta_{\max}, i = 1,...r\}$, where $\theta_{\min}$ and $\theta_{\max}$ are some known real numbers;

Assumption 4: $\Gamma_n \in \Omega_{\Gamma_n} \overset{\Delta}{=} \{\Gamma_n |\ 0 < \mu_n \leq \mu_{n\max}(p), p \in P\}$, where $\mu_{n\max}(p)$ are some known real constants.

Assumption 5: The bound B for the relation $\|d(v(t))\| \leq B$ is known.

Assumption 6: There exist known constants $0 < c_{\min} \leq c_{\max}$ such that the slope c in (7.3) satisfying $c \in [c_{\min}, c_{\max}]$.

Remark: Assumption 1 sets a restriction on the types of reference signals which may be used for design of the tracking controller. Assumption 2 is common for nonlinear controller designs [25]; In assumption 3 a new parameter vector Θ has been defined for the convenience of further development. This assumption implies that the range of the plant parameters, $a_i, i = 1,\ldots,r$, are known in advance. This is a reasonable assumption concerning the prior knowledge of the system; For assumption 4, based on the definition and property of the density function $\mu_n(p)$, it is reasonable to set an upper bound $\mu_{n\max}(p) > 0$ for $\mu_n(p)$. For the assumption 5, the bound B of modeling error of hysteresis is required as the modeling precision.

In presenting the developed robust adaptive control law, some parameter errors are defined as following:

$$\tilde{\Theta}(t) = \hat{\Theta}(t) - \Theta, \tag{7.14}$$

$$\tilde{\phi}(t) = \hat{\phi}(t) - \phi, \tag{7.15}$$

$$\tilde{\Gamma}_n = \hat{\Gamma}_n - \Gamma_n \tag{7.16}$$

where $\hat{\Theta}(t)$ is an estimate of Θ as given in Assumption 3, $\hat{\Gamma}_n$ is the estimate of the normalized density vector Γ_n due to the unknown vector Γ, and $\hat{\phi}(t)$ is an estimate of $\phi = 1/(bc) > 0$ due to the positive unknown plant parameters $b > 0$ and the uncertain constant $c = \| \Gamma \| > 0$ of hysteresis.

Given the plant and hysteresis model subjected to the assumptions described above, a control law is designed as following:

$$v = -k_d s(t) + \hat{\phi} v_{fd}(t) + \hat{\Theta}^T Y(X) + \hat{\Gamma}_n^T M - k_\varepsilon sat(\frac{s}{\varepsilon}) \tag{7.17}$$

with

$$\begin{aligned} v_{fd}(t) &= x_d^{(n)} - \dot{s}(t) \\ &= x_d^{(n)} - [0, \lambda^{(n-1)}, (n-1)\lambda^{(n-2)}, \ldots, (n-1)\lambda]^T \tilde{X}, \end{aligned} \tag{7.18}$$

and $k_d > 0$. k_ε is a control gain satisfying $k_\varepsilon > B / c_{\min}$. The parameters $\hat{\phi}(t)$, $\hat{\Theta}(t)$ and normalized density $\hat{\Gamma}_n$ will be updated by the following adaptation laws

$$\dot{\hat{\phi}}(t) = proj(\hat{\phi}, -\eta\, v_{fd} s_\varepsilon) \tag{7.19}$$

$$\dot{\hat{\Theta}}(t) = proj(\hat{\Theta}, -\gamma\, Y s_\varepsilon) \tag{7.20}$$

$$\dot{\hat{\Gamma}}_n = proj(\hat{\Gamma}_n, -q\, M s_\varepsilon) \tag{7.21}$$

where parameters η, γ and q are positive constants determining the rates of the adaptations, and the $proj(\cdot)$ are projection operators formulated as:

$$proj\,(\hat{\phi}, -\eta\, v_{fd} s_\varepsilon) = \begin{cases} 0 & if \quad \hat{\phi} = \phi_{\max}, \\ -\eta\, v_{fd} s_\varepsilon & if \quad \hat{\phi} \in [\phi_{\min}, \phi_{\max}] \\ & or \quad [\hat{\phi} = \phi_{\min}, and \quad \eta v_{fd} s_\varepsilon \le 0] \\ & or \quad [\hat{\phi} = \phi_{\max}, and \quad \eta v_{fd} s_\varepsilon \ge 0] \\ 0 & if \quad \hat{\phi} = \phi_{\min}, and \quad \eta v_{fd} s_\varepsilon > 0 \end{cases} \tag{7.22}$$

$$\{proj(\dot{\Theta}, -\gamma\, Y s_\varepsilon)\}_i = \begin{cases} 0 & if \quad \hat{\theta}_i = \theta_{i\max}, and \quad \gamma(Y s_\varepsilon) < 0 \\ -\gamma(Y s_\varepsilon)_i & if \quad \hat{\theta}_i \in [\theta_{i\min}, \theta_{i\max}] \\ & or \quad [\hat{\theta}_i = \theta_{i\min}, and \quad \gamma(Y s_\varepsilon)_i \le 0] \\ & or \quad [\hat{\theta}_i = \theta_{i\max}, and \quad \gamma(Y s_\varepsilon)_i \ge 0] \\ 0 & if \quad \hat{\theta}_i = \theta_{i\min}, and \quad \gamma(Y s_\varepsilon)_i > 0 \end{cases} \tag{7.23}$$

$$\{proj(\hat{\Gamma}_n, -q\, M s_\varepsilon)\}_j = \begin{cases} 0 & if \quad \hat{\theta}_i = \theta_{i\max}, and \quad q(M s_\varepsilon)_i < 0 \\ -q(M s_\varepsilon)_j & if \quad \hat{\theta}_i \in [\theta_{i\min}, \theta_{i\max}] \\ & or \quad [\hat{\theta}_i = \theta_{i\min}, and \quad q(M s_\varepsilon)_i \le 0] \\ & or \quad [\hat{\theta}_i = \theta_{i\max}, and \quad q(M s_\varepsilon)_i \ge 0] \\ 0 & if \quad \hat{\theta}_i = \theta_{i\min}, and \quad q(M s_\varepsilon)_i > 0 \end{cases} \tag{7.24}$$

with$i = 1 \sim r$, and $j = 1 \sim N$. The adaptive control model is shown as Fig. 7.1.

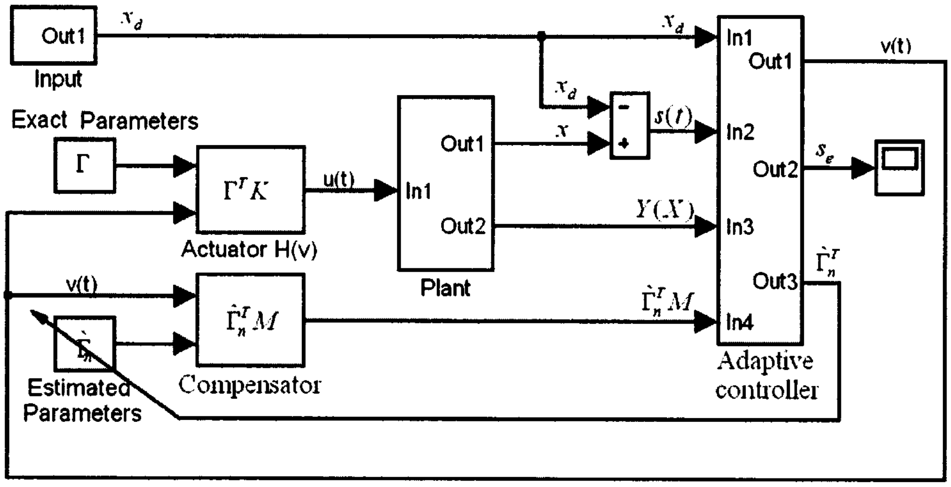


Fig.7.1. Adaptive control model for unknown nonlinear system with unknown input hysteresis

Remarks:

1) The projection operators for $\hat{\Theta}$ used in the above control law have the following properties [33]:

   i) $\hat{\Theta}(t) \in \Omega_\Theta$ if $\hat{\Theta}(t_0) \in \Omega_\Theta$;

   ii) $\| proj(\varpi, y) \| \leq \| y \|$ and

   iii) $-(\hat{\varpi} - \varpi)^T \Lambda proj(\hat{\varpi}, z) \geq -(\hat{\varpi} - \varpi)^T \Lambda z$, where Λ is a positive definite symmetric matrix. Note that these three properties are also valid for the projection operators defined for $\hat{\phi}$ and $\hat{\Gamma}_n(t, \hat{\varpi})$.

2) The projection operators require the upper and lower bounds of the parameters ϕ, Θ and Γ_n. Assumptions 2) ~ 4) are fundamental to these bounds. However, these parameters are only used to specify the variant ranges of the parameter for the projection operator. These ranges are not restricted as long as the estimated parameters are bounded.

3) The term $\hat{\Gamma}_n^T M$, which is associate with the compensation part $\Gamma^T M$ of the hysteresis, represents the dynamic effects of the actuator on its plant. The vectors M and $\hat{\Gamma}_n$ have $N = l(l+1)/2$ elements respectively. Here l denotes the number of lines to uniformly divide the Preisach plane P horizontally or vertically. The larger l leads to finer cell divisions of the Preisach plane P. The selection of the size of the cells depends on the accuracy requirement. As will be shown in the simulation example, extremely small sized cells are not necessary because it causes many calculations but only improves the accuracy slightly.

The stability of the closed-loop system described by equations (7.10), (7.17) and (7.19) ~ (7.24) is established in the following theorem:

Theorem: *For the plant given in (7.2) with the hysteresis (7.3), subject to Assumptions 1)~4), the robust adaptive controller specified by equations (7.17), (7.19)~(7.24) ensures that: if* $\hat{\Theta}(t_0) \in \Omega_\Theta$, $\hat{\Gamma}_n(t_0) \in \Omega_{\Gamma_n}$ *and* $\hat{\phi}(t_0) \in \Omega_\phi$, *all the closed-loop signals are bounded and the state vector* $X(t)$ *converges to* $\Omega_\varepsilon \overset{\Delta}{=} \{X(t) \,||\, \tilde{X}_i \,| \le 2^{i-1}\lambda^{i-n}\varepsilon, i = 1,\ldots,n\}$ *for* $\forall t \ge t_0$.

Proof: For system (7.10), considering (7.12), (7.17) and (7.18) the time derivative of the filtered error (7.12) is:

$$\dot{s}(t) = \frac{1}{\phi}[v(t) + \frac{1}{c}d(v(t)) - \Theta^T Y - \Gamma_n^T M - \phi v_{fd}(t)] \tag{7.25}$$

By applying the control law (7.17) to (7.25), it changes as

$$\dot{s}(t) = \frac{1}{\phi}[-k_d s(t) + (\hat{\phi} - \phi)v_{fd}(t) + (\hat{\Theta} - \Theta)^T Y + (\hat{\Gamma}_n - \Gamma_n)^T M - k_\varepsilon sat(\frac{s}{\varepsilon}) + \frac{1}{c}d(v)] \tag{7.26}$$

To establish global boundedness, a candidate *Lyapunov* function is defined as following:

$$V(t)=\frac{1}{2}[\phi s_\varepsilon{}^2+\frac{1}{\gamma}\tilde{\Theta}^T\tilde{\Theta}+\frac{1}{\eta}\tilde{\phi}^2+\frac{1}{q}\tilde{\Gamma}_n{}^T\tilde{\Gamma}_n]. \tag{7.27}$$

Since the discontinuity at $|s|=\varepsilon$ is of the first kind, the derivative $\dot{V}(t)$ exists for all s.

Applying the facts $s_\varepsilon \dot{s}_\varepsilon = s_\varepsilon \dot{s}$, $\tilde{\Theta}=\hat{\Theta}-\Theta$, $\dot{\tilde{\Theta}}=\dot{\hat{\Theta}}$, $\dot{\tilde{\Gamma}}_n=\dot{\hat{\Gamma}}_n$, $\tilde{\Gamma}_n=\hat{\Gamma}_n-\Gamma_n$, $\tilde{\phi}=\hat{\phi}-\phi$ and $\dot{\tilde{\phi}}=\dot{\hat{\phi}}$, one has as

$$\dot{V}(t)=\phi s_\varepsilon \dot{s}+\frac{1}{\gamma}(\hat{\Theta}-\Theta)^T\dot{\hat{\Theta}}+\frac{1}{\eta}(\hat{\phi}-\phi)\dot{\hat{\phi}}+\frac{1}{q}(\hat{\Gamma}_n-\Gamma_n)^T\dot{\hat{\Gamma}}_n \tag{7.28}$$

When $|s|\le\varepsilon$ and $s_\varepsilon=0$, applying adaptive control law (7.19) ~ (7.24) to (7.28) gives

$$\dot{V}(t)=\phi s_\varepsilon \dot{s}-\frac{1}{\gamma}(\hat{\Theta}-\Theta)^T\gamma Y s_\varepsilon-\frac{1}{\eta}(\hat{\phi}-\phi)\eta v_{fd}s_\varepsilon-\frac{1}{q}(\hat{\Gamma}_n-\Gamma_n)^T qMs_\varepsilon$$

$$=0 \quad for \quad |s|\le\varepsilon \tag{7.29}$$

When $|s|>\varepsilon$, applying (7.26) to (7.28), one has

$$\dot{V}(t)=s_\varepsilon[-k_d s(t)+(\hat{\phi}-\phi)v_{fd}(t)+(\hat{\Theta}-\Theta)^T Y+(\hat{\Gamma}_n-\Gamma_n)^T M-k_\varepsilon sat(\frac{s}{\varepsilon})+\frac{1}{c}d(v)]$$

$$+\frac{1}{\gamma}(\hat{\Theta}-\Theta)^T\dot{\hat{\Theta}}+\frac{1}{\eta}(\hat{\phi}-\phi)\dot{\hat{\phi}}+\frac{1}{q}(\hat{\Gamma}_n-\Gamma_n)^T\dot{\hat{\Gamma}}_n \tag{7.30}$$

Considering the definition of the filtered error as (7.12), one has

$$s_\varepsilon = s-\varepsilon sat(\frac{s}{\varepsilon})=\begin{cases} s-\varepsilon & for \quad s>\varepsilon \\ 0 & for \quad -\varepsilon\le s\le\varepsilon \\ s+\varepsilon & for \quad s<-\varepsilon \end{cases} \tag{7.31}$$

which leads to $|s|\ge|s_\varepsilon|>0$ while $|s|>\varepsilon$. Furthermore,

$$-k_d s s_\varepsilon \le -k_d s_\varepsilon{}^2 \tag{7.32}$$

With (7.32), (7.30) can be simplified as

$$\dot{V}(t) \le -k_d s_\varepsilon^{\ 2} + s_\varepsilon[(\hat{\phi}-\phi)v_{fd}(t) + (\hat{\Theta}-\Theta)^T Y + (\hat{\Gamma}_n - \Gamma_n)^T M - k_\varepsilon sat(\frac{s}{\varepsilon}) + \frac{1}{c}d(v)]$$

$$+\frac{1}{\gamma}(\hat{\Theta}-\Theta)^T \dot{\hat{\Theta}} + \frac{1}{\eta}(\hat{\phi}-\phi)\dot{\hat{\phi}} + \frac{1}{q}(\hat{\Gamma}_n - \Gamma_n)^T \dot{\hat{\Gamma}}_n \qquad (7.33)$$

Utilizing the adaptive laws given in (7.22) ~ (7.24) and the properties of the projection operators as

$$\frac{1}{\gamma}(\hat{\Theta}-\Theta) proj(\hat{\Theta}, -\gamma Y s_\varepsilon) \le -(\hat{\Theta}-\Theta)^T Y s_\varepsilon,$$

$$\frac{1}{q}(\hat{\Gamma}_n - \Gamma_n)^T proj(\hat{\Gamma}_n, -q M s_\varepsilon) \le -(\hat{\Gamma}_n - \Gamma_n)^T M s_\varepsilon$$

and
$$\frac{1}{\eta}(\hat{\phi}-\phi) proj(\hat{\phi}, -\eta v_{fd} s_\varepsilon) \le -(\hat{\phi}-\phi) v_{fd} s_\varepsilon,$$

one has
$$\dot{V}(t) \le -k_d s_\varepsilon^{\ 2} - k_\varepsilon s_\varepsilon sat(\frac{s}{\varepsilon}) + \frac{1}{c}d(v)s_\varepsilon \le 0$$

since $|s_\varepsilon| = s_\varepsilon sat(s/\varepsilon)$ for $|s| > \varepsilon$, the above inequality becomes

$$\dot{V}(t) \le -k_d s_\varepsilon^{\ 2} - k_\varepsilon |s_\varepsilon| + \frac{d(v)}{c} s_\varepsilon$$

$$\le -k_d s_\varepsilon^{\ 2} - k_\varepsilon |s_\varepsilon| + \frac{B}{c_{\min}} |s_\varepsilon|$$

$$\le -k_d s_\varepsilon^{\ 2} \quad \forall |s| > \varepsilon. \qquad (7.34)$$

Equations (7.27), (7.29) and (7.34) imply that $V(t)$ is a *Lyapunov* function which leads to global boundedness of s_ε, $\tilde{\Theta} = \hat{\Theta} - \Theta$, $\tilde{\Gamma}_n = \hat{\Gamma}_n - \Gamma_n$, and $\tilde{\phi} = \hat{\phi} - \phi$. From the definition of s_ε, $s(t)$ is bounded. It is easily shown that if $\tilde{X}(0)$ is bounded, then $\tilde{X}(t)$ is also bounded for all t, and since $X_d(t)$ is bounded by design, $X(t)$ must also be bounded. To complete the proof and establish asymptotic convergence of the tracking error, it is necessary to show that $s_\varepsilon \to 0$ as $t \to \infty$. This is accomplished by applying *Barbalat's lemma* [25, 27] to the continuous, nonnegative function

$$V_1(t) = V(t) - \int_0^t (\dot{V}(\tau) + k_d s_\varepsilon{}^2(\tau)) d\tau \geq 0$$

with
$$\dot{V}_1(t) = -k_d s_\varepsilon{}^2(\tau) \leq 0 \tag{7.35}$$

It can easily be shown that every term in (7.25) is bounded. It is required to show that $v(t) - \Gamma_n^T M$ in (7.25) is bounded. From (7.3) one has $u(t) = c[v(t) - \Gamma_n^T M]$ which is the output of an actuator of the system. Thus,

$$| v(t) - \Gamma_n^T M | = | \frac{1}{c} u(t) | = | \frac{1}{c} \Gamma^T K |$$

By choosing the extreme values for every KP kernels as $k_p = \pm 1$, the above equation changes to

$$| v(t) - \Gamma_n^T M | \leq | \frac{\| \Gamma \|}{c} | = 1.$$

Hence $\dot{s}(t)$, and $\dot{s}_\varepsilon(t)$ are bounded. Furthermore, $\ddot{V}_1(t) = -2k_d s_\varepsilon(t) \dot{s}_\varepsilon(t)$ is bounded. This implies that $\dot{V}_1(t)$ is a uniformly continuous function of time. Since $V_1(t) \geq 0$ and $\dot{V}_1(t) \leq 0$ for all t, applying *Barbalat's lemma* [25, 27] proves that $\dot{V}_1(t) \to 0$ as $t \to \infty$. Therefore, from (7.35) it can be demonstrated that $s_\varepsilon(t) \to 0$ as $t \to \infty$. The remark following equation (7.12) indicates that $\tilde{X}(t)$ will converge to Ω_ε.

Remark: It is now clear that the developed control strategy to deal with the hysteresis nonlinearities can be applied to many systems and may not necessarily be limited to the system described by (7.2). However, it should be emphasized that the goal is to develop a control strategy in a simpler setting that reveals its essential features.

7.3 Simulation Studies

To illustrate the control methodology introduced in section 7.2, in this section it is applied to control a nonlinear system with input hysteresis (see Fig.7.1). The nonlinear system is described by

$$\dot{x}(t) = a_1 x(t) + a_2 \frac{1-e^{-x(t)}}{1+e^{-x(t)}} + bu(t) \tag{7.36}$$

where $u(t)$ represents the output of an actuator with hysteresis nonlinearity. The actual parameter values of a_1 , a_2 , b and density function $\mu(p) = \mu(p_1, p_2)$ are assumed unknown, and some nominal values are $a_1 = -2$, $a_2 = 1$, $b = 6$. The nominal density function $\mu(p_1, p_2)$ will be specified later. Without control, i.e., $u(t) = 0$, the system is stable because $a_1 < 0$. The objective is to control the above system state $x(t)$ to follow a desired trajectory $x_d(t)$, which is given by a reference command as

$$x_d(t) = 2.2\sin(\pi t/3 - \pi/2) + 2.2\sin(\sqrt{2}\pi t/3 - \pi/2). \tag{7.37}$$

The system to be controlled is preceded by an actuator with hysteresis nonlinearity, which can be described by the linearly parameterized KP model as

$$u(t) = \Gamma^T K + d(v(t)) \tag{7.38}$$

with the parameters $N = [l(l+1)]/2,\ l = 20,$ and $p \in [v^-, v^+] = [-4, 4]$. The nominal density vector Γ and operator vector $K[v(t)]$ of the KP model are rearranged from a 2-dimensional upper-triangular matrix consisting of densities μ_{ij} and a 2-dimensional upper-triangular matrix consisting of outputs of the KP operators. These two upper-triangular matrices are only defined over the Preisach plane P. The elements μ_{ij} are calculated by a normalizing function as in equations (4.56) and (4.57) with $\sigma = 1.0$ and

$\overline{p}_1 = \overline{p}_2 = 0$. The normalized densities μ_{ij} over the Preisach plane P are illustrated in Fig.7.2. In equation (7.38), $\| \Gamma \| = 1$ since μ_{ij} are the normalized densities.

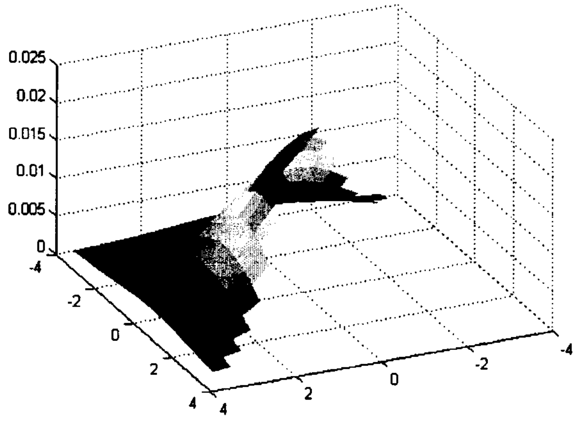


Fig. 7.2 Normalized densities μ_{ij} over the P plane

Without adaptive control, The responses of the dynamic system (7.36) including (7.38) subject to the input signal (7.37) with the initial conditions of $x_d(0) = -4.4$, $x(0) = -4.4$ and $u(0) = -1$ is shown in Fig.7.6. The hysteresis loops of the actuator are formed as in Fig.7.3. The relationship between output and input of the nonlinear compensator is shown in Fig.7.4, and its relationship with the actuator hysteresis is shown in Fig.7.5. Also, simulation studies show that different frequency values of input signal $x_d(t)$ result in hysteresis loops similar to that in Fig.7.3. This means that a fixed density distribution of the KP model combined with the KP operators fully determines the properties of the actuator, and the linearly parameterized KP model (7.38) can be used to describe the hysteresis nonlinearity of the actuator. In Fig.7.3 the maximum and minimum output values of the actuator equal to $+1$ and -1 respectively because the normalized densities are taken for the input hysteresis.

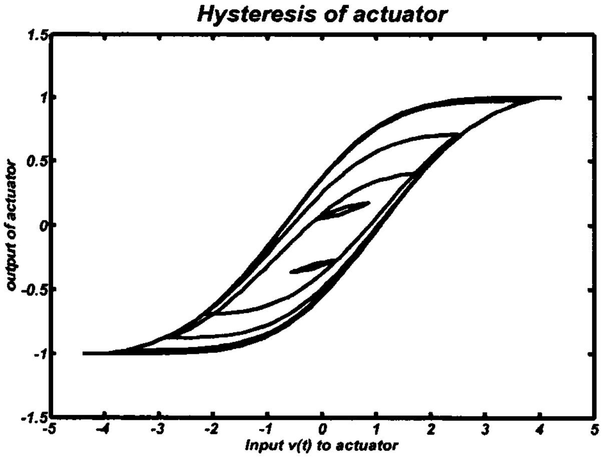


Fig.7.3 Hysteresis nonlinearity of actuator

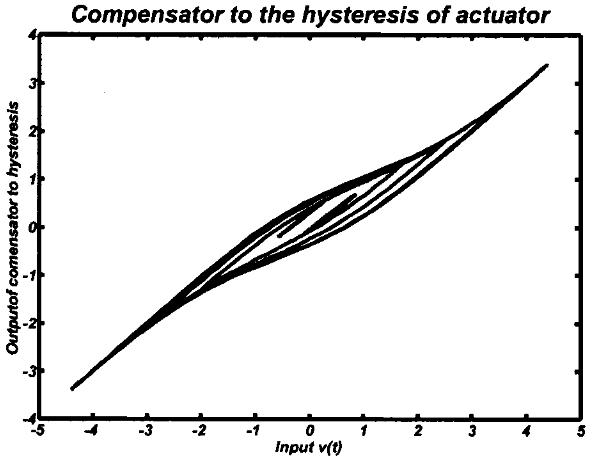


Fig.7.4 Hysteresis related compensator

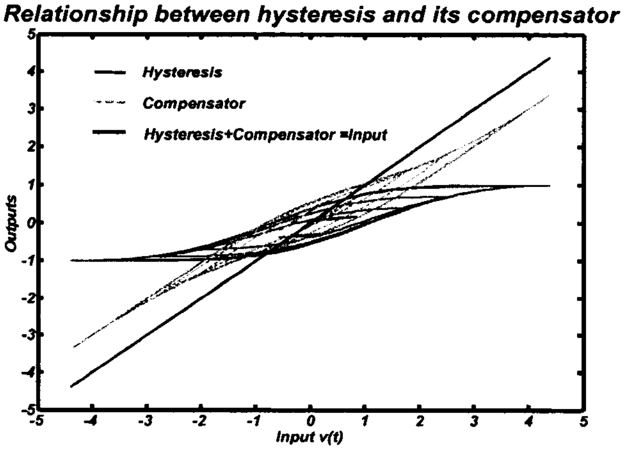


Fig.7.5 Relationship between hysteresis and its compensator

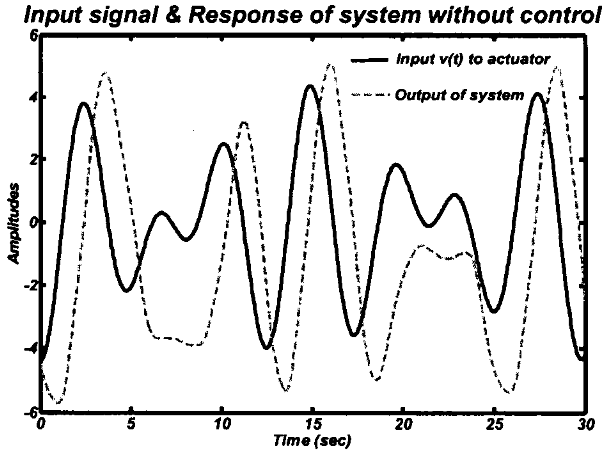


Fig.7.6 Responses of uncontrolled system

From Fig.7.6 it can be seen that without control, the output of the plant is quite different from the output of the reference model due to the nonlinearities of its actuator and the plant. If there is some parameter uncertainness in the real system, the response of the open-loop system becomes more different from the reference command. Thus, it is necessary to design a robust adaptive controller for the system to modify the input of the actuator, and furthermore to force the plant to generate an output to follow the reference command.

In the simulations, the robust adaptive control law (7.17) ~ (7.24) were used. So far, there has been no analytical approach developed for the selection of the control constants but only iterative simulation has been used. The control gain in the first term of the control law is set as $k_d = 10$, and the control gains to calculate the adaptive parameters $\hat{\theta}$, $\hat{\phi}$, and $\hat{\mu}_n$ are chosen as $\gamma = 2$, $\eta = 2$ and $q = 1.3$. The bound for modeling error of hysteresis is assumed as $B = 0.05$, which means the modeling error must less than 5% of the output multitude of the saturation states. $c_{\min} = 0.95$ and $c_{\max} = 1.0$. Thus, the control gain $k_\varepsilon = 0.053$. The minimum and maximum values of the normalized densities $\hat{\mu}_n$ are set as 0 and the nominal densities distribution μ_{ij} are calculated by equations (4.56) and (4.57). The initial distribution is chosen arbitrarily between these two extreme surfaces, 0 and μ_{ij}. Here the initial adaptive densities are chosen as $\hat{\mu}_{ij}(t_0) = 0.7\mu_{ij}$, and the initial normalized densities are calculated as

$$\hat{\mu}_{nij}(t_0) = \hat{\mu}_{ij}(t_0) / sum(\hat{\Gamma}(t_0)) = \hat{\mu}_{ij}(t_0) / (\sum_{i=1}^{l}\sum_{j=1}^{i}\hat{\mu}_{ij}(t_0)).$$

The initial parameters $\theta(t_0) = 1/3$, $\phi(t_0) = 1/4$, the initial state $x(0) = -4.4$, and the constraint to the error is chosen as $\varepsilon = 0.025$. The simulation sample time is 0.01 sec. From the simulation, it is observed that the system responses are more sensitive to control gains k_d and q. It is intuitive that q directly corrects the errors caused by the hysteresis.

To evaluate the effectiveness of the proposed adaptive robust control scheme, the simulation is also conducted without controlling the effects of the hysteresis, i.e., $\hat{\Gamma}_n^T M = 0$ in the controller $v(t)$ in equation (7.17). Simulation results of the tracking errors of the system state for the desired trajectory with and without controlling the effects of the hysteresis are shown in Fig.7.7, where the tracking error (solid line) has been kept as $s_1(t) = \tilde{x}(t) < 0.025$, while the tracking error $s_2(t)$ for $\hat{\Gamma}_n^T M = 0$ shown by the dash-dot line is larger than $s_1(t)$ and cannot satisfy the design requirement $s(t) \le \varepsilon = 0.025$. Fig.7.7 shows that the signal $\hat{\Gamma}_n^T M$ which is designed to reduce the hysteresis effects is necessary and effective, and also verifies the excellent tracking performance of the proposed algorithm.

It should be mentioned that simulations for several different desired trajectories with various parameter values and initial conditions have also been conducted. Results show that they all displayed similar behaviors as the ones shown. The simulation for $l = 30$, which divides the Preisach plane P into smaller cells, gives almost identical results but involves larger cost of calculation. This further verifies that the developed control algorithm is repeatable and computationally implementable. This means that dividing the Preisach plane P into extremely small sized cells is not necessary because it causes too much calculation but only marginally improves the accuracy.

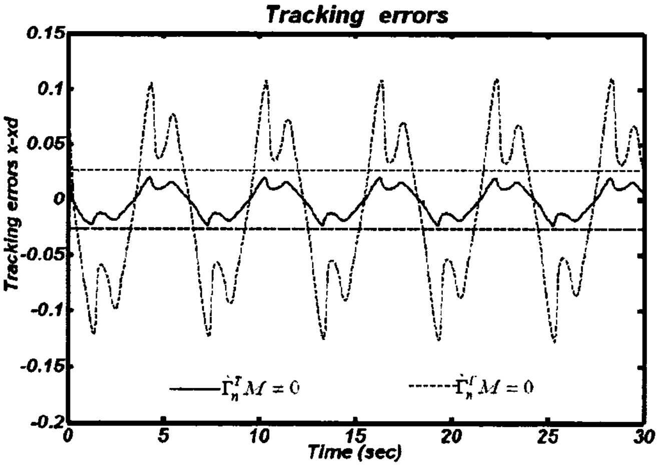


Fig.7.7 Tracking errors of state with the control term $\hat{\Gamma}_n^T M \neq 0$ and with $\hat{\Gamma}_n^T M = 0$

7.4 Conclusions

In this chapter, a modeling approach to actuators with hysteresis by employing the parameterized Krasnosel'skii–Pokrovkii model is systemically presented, and then a robust adaptive control architecture is proposed for a class of continuous-time nonlinear dynamic systems preceded by the actuators with hysteresis. By showing the properties of the hysteresis model, a robust adaptive control scheme is developed without constructing the hysteresis inverse. The new adaptive control law ensures global stability of the controlled system and achieves both stabilization and trajectory tracking within desired precision. Simulation results performed on a simple nonlinear system illustrated and further validated the effectiveness of the proposed approach.

CHAPTER 8

A NOVEL HYSTERESIS MODEL

In the previous chapters, the classical Preisach hysteresis model and the KP model have been presented. The classical Preisach model is the basis for the KP model. The elementary operator of the KP model is a continuous kernel, which can be expressed by the classical Preisach model. Even though both models can be rewritten in linearly parameterized forms, the linearly parameterized KP model is more suitable to be combined with the adaptive control methodology for systems. But, both models have the same drawbacks such as zero-initial-slope of the reverse curves and no capability to describe reversible parts of the hysteresis actuators. This fact motives further research to define a new hysteresis model to overcome these shortcomings. In this chapter, the definition of the new hysteresis model will be introduced, and the application of the model in the adaptive control of nonlinear systems will be illustrated.

8.1 Definition for the New Hysteresis Model

The new hysteresis model is defined as an integration of weighted elementary operators over a specific domain as

$$\begin{aligned} u(t) &= \int_P k_p[v,\psi,\xi_p](t)\mu(p)dp \\ &= \int_{v^-}^{v^+}\int_{v^-}^{p_2} k_{p_1 p_2}[v,\psi,\xi_{p_1 p_2}](t)\cdot\mu(p_1,p_2)dp_1 dp_2 \end{aligned} \qquad (8.1)$$

Also it can be simply expressed in operator form as

$$u(t) = H[v](t) \qquad (8.2)$$

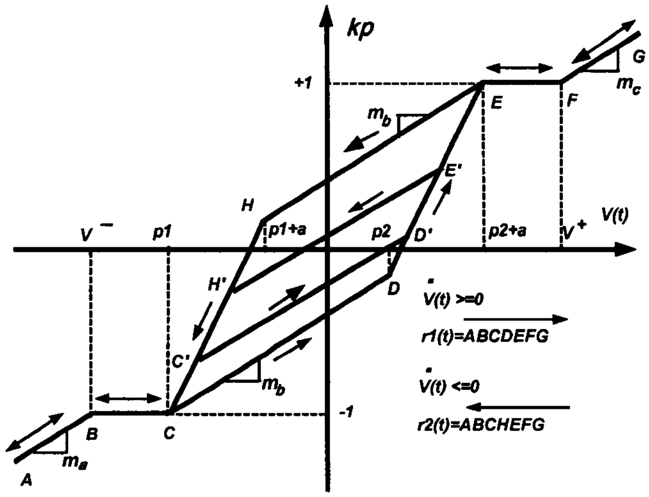


Fig. 8.1 a new elementary hysteresis operator k_p

where $v(t)$ is the hysteresis input; $u(t)$ is the hysteresis output; $H(\cdot)$ is the hysteresis operator; $k_p[v,\psi,\xi_p](t)$ as shown in Fig. 8.1 is called the *operator function* which defines the output values of a specific elementary operator with a pair of parameters $p=(p_1,p_2)\in P$ as it is subjected to the input $v(t)$; $\mu(p)$, a function of parameters $p=(p_1,p_2)\in P,$ is the *weighting factor* of the elementary operator $k_p[v,\psi,\xi_p](t)$; and the triangular region P shown in Fig. 8.2 is the *hysteresis plane* over which the hysteresis occurs, and it can be defined as

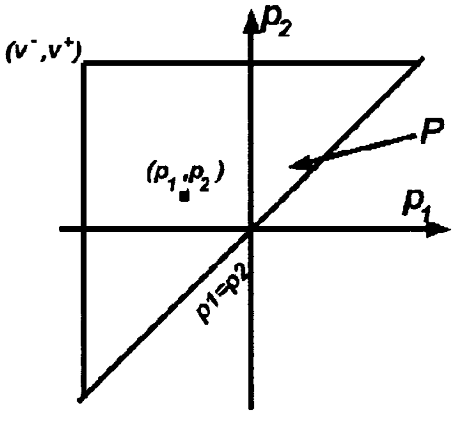


Fig. 8.2 Hysteresis Plane P

$$P=\{p(p_1,p_2)\in R^2 : v^+ - a \geq p_2 \geq p_1 \geq v^-\} \tag{8.3}$$

where v^- and v^+ are the lower and upper hysteresis input limits, respectively, and a is a rise-constant of hysteresis elementary operator which will be introduced later.

Mathematically, the operator function $k_p[v,\psi,\xi_p](t)$ can be expressed by

$$k_p[v,\psi,\xi_p](t)=\begin{cases}\max(\ \xi_p(t), r_1[v](t)) & for \quad \dot{v}\geq 0\\ \\ \min(\ \xi_p(t), r_2[v](t)) & for \quad \dot{v}\leq 0\end{cases} \tag{8.4}$$

where k_p is the output of the elementary operator corresponding to its input $v(t)$; the $\xi_p(t)$ is the output coordinate of turning point where input $v(t)$ changes its varying trend when $v(t)=\psi$, and the pair of $(\psi,\xi_p(t))$ is defined by

$$\psi=\begin{cases}v(t_0) & if \ t=t_0\\ \\ \psi(t_i) & if \ t=t_i>t_{i-1} and\ sign\ (\dot{v}(t^+))=-sign\ (\dot{v}(t^-))\\ \\ \psi(t_{i-1}) & if \ t_i\geq t>t_{i-1} and\ sign\ (\dot{v}(t^+))=sign\ (\dot{v}(t^-))\end{cases} \tag{8.5a}$$

$$\xi_p(t)=\begin{cases}u(t_0) & if \ t=t_0\\ \\ k_p[v(t_i),\psi(t_{i-1}),\xi_p(t_{i-1})] & if \ t=t_i>t_{i-1} and\ sign\ (\dot{v}(t^+))=-sign\ (\dot{v}(t^-))\\ \\ m_b[v(t)-\psi(t_{i-1})]+\xi_p(t_{i-1}) & if \ t_i\geq t>t_{i-1} and\ sign\ (\dot{v}(t^+))=sign\ (\dot{v}(t^-))\end{cases} \tag{8.5b}$$

where i is the sequence of the turning points.

The two boundary functions $r_1[v](t): ABCDEFG$ and $r_2[v](t): ABCHEFG$ forming the major loop of the elementary operator $k_p[v,\psi,\xi_p](t)$ (see Fig.8.1) are defined by

$$r_1[v](t) = \begin{cases} m_a(v(t) - v^-) - 1 & \mathit{if} \;\; v(t) < v^- \\ -1 & \mathit{if} \;\; v^- \le v(t) < p_1 \\ m_b(v(t) - p_1) - 1 & \mathit{if} \;\; p_1 \le v(t) < p_2 \\ \upsilon(v(t) - p_2 - a) + 1 & \mathit{if} \;\; p_2 \le v(t) \le p_2 + a \\ 1 & \mathit{if} \;\; p_2 + a < v(t) \le v^+ \\ m_c(v(t) - v^+) + 1 & \mathit{if} \;\; v(t) > v^+ \end{cases} \tag{8.6}$$

$$r_2[v](t) = \begin{cases} m_a(v(t) - v^-) - 1 & \mathit{if} \;\; v(t) < v^- \\ -1 & \mathit{if} \;\; v^- \le v(t) < p_1 \\ \upsilon(v(t) - p_1) - 1 & \mathit{if} \;\; p_1 \le v(t) < p_1 + a \\ m_b(v(t) - p_2 - a) + 1 & \mathit{if} \;\; p_1 + a \le v(t) \le p_2 + a \\ 1 & \mathit{if} \;\; p_2 + a < v(t) \le v^+ \\ m_c(v(t) - v^+) + 1 & \mathit{if} \;\; v(t) > v^+ \end{cases} \tag{8.7}$$

where $\upsilon = 2 - m_b(p_2 - p_1)]/a$, m_a is the slope of the half line AB; m_b is the slope of both segments CD and EH ; m_c is the slope of the half line FG ; the horizontal segments BC and EF have end points with coordinates as $B(v^-,-1), C(p_1 - 1), E(p_1 + 1)$ and $F(v^+,+1)$; the segments CH and DE are parallel to each other with slope as $\upsilon = [2 - m_b(p_2 - p_1)]/a$ so that their end points D and H have horizontal coordinates as p_2 and $p_1 + a$ respectively.

To explain the formation (8.4, 8.5) of the elementary operator as the input $v(t)$ varies, the input is assumed starting from a value $v(t) < v^-$. As the input $v(t)$ keeps increasing, the point of input-output pair $(v(t), k_p)$ traces along the boundary $r_1 : ABCDEFG$. And then the input $v(t)$ changes its varying tendency and keeps decreasing after it increases to $v(t) > p_2 + a$; this decreasing input variation results in the input-output point $(v(t), k_p)$ to slide down along the boundary $r_2 : ABCHEFG$. If the turning point E' with coordinates (ψ, ξ_p) is on the segment DE, the point $(v(t), k_p)$ slides down along the segment $E'H'$ and then along the segment $H'CBA$. After that, if the input $v(t)$ changes its varying tendency again and keeps increasing after it decreases to $p_1 + a > v(t) > p_1$, the point $(v(t), k_p)$ goes up along the segment $C'D'$ and then along the segment $D'EFG$. And the segments $E'H'$, $H'C'$, $C'D'$ and $D'E'$ form an inner loop of the elementary operator. From this explanation, the coordinates of the turning points E' and C' are critical for the operator function to determine its loops.

The weighting factor $\mu(p) = \mu(p_1, p_2)$ is called the density of the elementary operator k_p. Each point $p(p_1, p_2)$ in the P plane associates with an elementary operator k_p and has its specific density factor $\mu(p_1, p_2)$. The function $\mu(p_1, p_2)$ to describe the densities of points in the entire Preisach plane P is called the density function or density distribution of the newly defined hysteresis model.

The integral model of hysteresis (8.1) can be interpreted as a parallel connection of an infinite number of weighted elementary operators as shown in Fig.8.3.

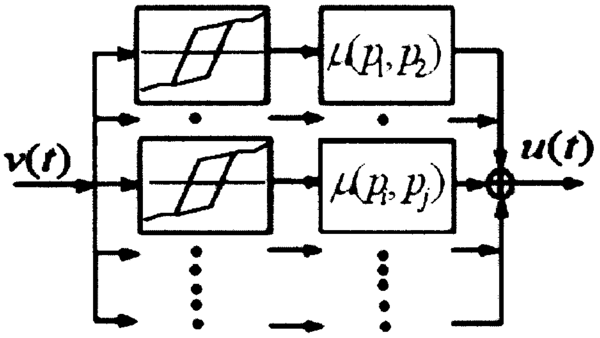


Fig. 8.3 Parallel connection form of new defined Hysteresis Model

8.2 Definition of Compensator of the New Hysteresis Model

Referring to the definition of the newly defined operator function $k_p[v, \psi, \xi_p](t)$ one can construct a elementary compensator $m_p[v, \psi, \varsigma_p](t)$ as shown in Fig.8.4

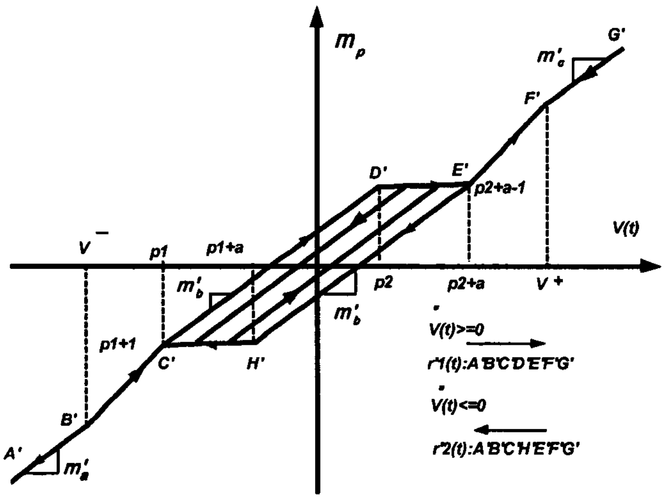


Fig. 8.4 Compensator m_p of the newly defined elementary operator k_p

so that

$$k_p[v,\psi,\xi_p](t)+m_p[v,\psi,\varsigma_p](t)=v(t) \tag{8.7}$$

The mathematic definition of the elementary compensator $m_p[v,\psi,\varsigma_p](t)$ related to the operator function $k_p[v,\psi,\xi_p](t)$ is expressed by

$$m_p[v,\psi,\varsigma_p](t)=\begin{cases}\min(\varsigma_p(t),r_1'[v](t)) & for\ \dot{v}(t)\geq 0\\ \\ \max(\varsigma_p(t),r_2'[v](t)) & for\ \dot{v}(t)\leq 0\end{cases} \tag{8.8}$$

where m_p is the output of the elementary compensator corresponding to its input $v(t)$; the $\zeta_p(t)$ is the output coordinate of the turning point where input $v(t)$ changes its varying trend when $v(t)=\psi$, and the coordinates $(\psi,\zeta_p(t))$ of the turning point is defined by

$$\psi=\begin{cases}v(t_0)\ \ if\,t=t_0\\ \\ \psi(t_i)\ \ if\,t=t_i>t_{i-1}\,and\ \ sign\ \ (\dot{v}(t^+))=-sign\ \ (\dot{v}(t^-))\\ \\ \psi(t_{i-1})if\,t_i\geq t>t_{i-1}\,and\ \ sign\ \ (\dot{v}(t^+)=sign\ \ (\dot{v}(t^-))\end{cases} \tag{8.9a}$$

and $$\varsigma_p(t)=\begin{cases} v(t_0)-u(t_0) & if\, t=t_0 \\ m_p[v(t_i),\psi(t_{i-1}),\varsigma_p(t_{i-1})] & if\, t=t_i>t_{i-1}\, and\ sign\ (\dot{v}(t^+))=-sign\ (\dot{v}(t^-)) \\ m_b'[v(t)-\psi(t_{i-1})]+\varsigma_p(t_{i-1}) & if\, t_i\ge t>t_{i-1}\, and\ sign\ (\dot{v}(t^+)=sign\ (\dot{v}(t^-)) \end{cases} \quad (8.9b)$$

where i is the sequence of the turning points, and t_i is the instance of time at the ith turning point.

The two boundary functions $r_1'[v](t): A'B'C'D'E'F'G'$ and $r_2'[v](t): A'B'C'H'E'F'G'$ enveloping the major loop of elementary compensator $m_p[v,\psi,\varsigma_p](t)$ (see Fig.8.4) are defined by

$$r_1'[v](t)=\begin{cases} m_a'(v(t)-v^-)+v^-+1 & if\ \ v(t)<v^- \\ v(t)+1 & if\ \ v^-\le v(t)<p_1 \\ m_b'(v(t)-p_1)+p_1+1 & if\ \ p_1\le v(t)<p_2 \\ \vartheta[v(t)-(p_2+a)]+p_2+a-1 & if\ \ p_2\le v(t)\le p_2+a \\ v(t)-1 & if\ \ p_2+a<v(t)\le v^+ \\ m_c'(v(t)-v^+)+v^+-1 & if\ \ v(t)>v^+ \end{cases} \quad (8.10)$$

$$r_2'[v](t)=\begin{cases} m_a'(v(t)-v^-)+v^-+1 & if\ \ v(t)<v^- \\ v(t)+1 & if\ \ v^-\le v(t)<p_1 \\ \vartheta(v(t)-p_1)+p_1+1 & if\ \ p_1\le v(t)<p_1+a \\ m_b'[v(t)-(p_2+a)]+p_2+a-1 & if\ \ p_1+a\le v(t)\le p_2+a \\ v(t)-1 & if\ \ p_2+a<v(t)\le v^+ \\ m_c'(v(t)-v^+)+v^+-1 & if\ \ v(t)>v^+ \end{cases} \quad (8.11)$$

where $\vartheta=[a-2+(1-m_b')(p_2-p_1)]/a$, m_a' is the slope of the half line $A'B'$; m_b' is the slope of both segments $C'D'$ and $E'H'$; m_c' is the slope of the half line $F'G'$; the slopes of segments $B'C'$ and $E'F'$ have the same value as 1. The coordinates of key points are $B'\ (v^-,v^-+1)$, $C'\ (p_1,p_1+1)$, $E'\ (p_2+a,p_2+a-1)$, and $F'\ (v^+,v^+-1)$. The segments $C'H'$ and $D'E'$ are parallel to each other with the same slope as ϑ and their end points

D' and H' have horizontal coordinates as p_2 and p_1+a, respectively. Thus, the key points D' and H' are determined by $C'H'$ and $D'E'$ as $D'\,(p_2, p_1+1+m_b'(p_2-p_1))$ and $H'\,(p_1+a, p_2+a-1-m_b'(p_2-p_1))$.

The formation principle of the inner loops in the compensator is similar with that of the newly defined elementary operator except for different choices of the operator "*max*" or "*min*" in equations (8.4) and (8.8). This can be explained as followings. As the input $v(t)$ changes its variation trend from increasing to decreasing $(from \quad \dot{v}(t)\geq 0 \quad to \quad \dot{v}(t)\leq 0)$, the curves of the compensator inner loop are above the decreasing boundary $r_2'[v](t): A'B'C'H'E'F'G'$, but in contrary the curves of the inner loop elementary operator are below its decreasing boundary $r_2[v](t): ABCHEFG$. Thus, the "*max*" operator is taken in (8.8) but the "*min*" operator is selected in (8.4). Similarly, the "*min*" operator is taken in (8.8) but the "*max*" operator is selected in (8.4) if the input $v(t)$ changes its variation tendency from decreasing to increasing (from $\dot{v}(t)\leq 0$ to $\dot{v}(t)\geq 0$).

The weighting factor $\mu(p)=\mu(p_1,p_2)$ is the same density of the elementary operator k_p. Thus, to model a hysteresis, either operator k_p or compensator m_p is used in the newly defined hysteresis model, and the same density function is used.

There are relationships between the slopes of the segments of he compensator m_p and that of the corresponding segments of the elementary operator k_p, as

$$m'_a = 1 - m_a \tag{8.12}$$

$$m'_b = 1 - m_b \tag{8.13}$$

$$m'_c = 1 - m_c; \tag{8.14}$$

Also there exists a relationship between the vertical coordinates $\varsigma_p(t_i)$ and $\xi_p(t_i)$ of the $i\,th$ turning point of the compensator m_p and it corresponding operator k_p as

$$\varsigma_p(t_i) + \xi_p(t_i) = v(t_i) \tag{8.15}$$

With these relationships (8.12) ~ (8.15) and the definitions of the compensator $m_p[v,\psi,\varsigma_p](t)$ and its operator $k_p[v,\psi,\xi_p](t)$, it can be easy proven that the relationship between the compensator and its related elementary operator is expressed as equation (8.7), and their relationship is shown in Fig.8.5.

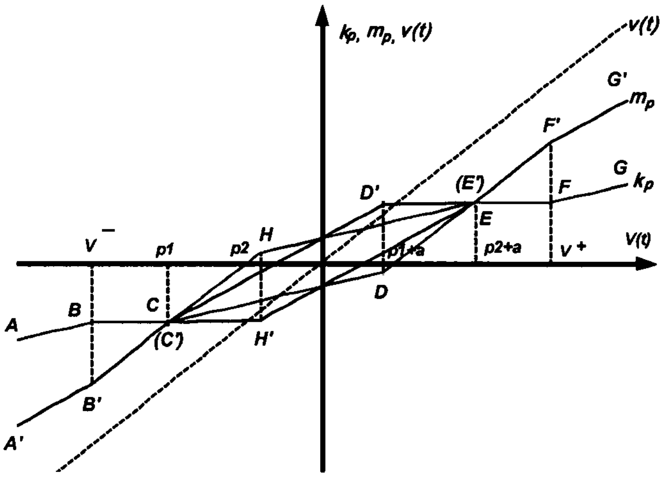


Fig. 8.5 Relationship between k_p and m_p

Thus, by defining the compensator m_p of the operator k_p, the newly defined hysteresis model can be alternatively described by an integral of the compensator over the hysteresis plane P as

$$\begin{aligned} u(t) = H[v](t) &= \int_P (v(t) - m_p[v,\psi,\zeta_p](t))\mu(p)dp \\ &= v(t)\int_P \mu(p)dp - \int_P m_p[v,\psi,\zeta_p](t)\mu(p)dp \\ &= cv(t) - \int_{v^-}^{v^+}\int_{v^-}^{p_2} m_{p_1p_2}[v,\psi,\varsigma_{p_1p_2}](t)\cdot\mu(p_1,p_2)dp_1dp_2 \end{aligned} \tag{8.16}$$

where $c = \int_P \mu(p)dp = \int_{v^-}^{v^+}\int_{v^-}^{p_2} \mu(p_1,p_2)dp_1dp_2$ is a constant for a particular hysteresis.

8.3 Linearly Parameterized Hysteresis Model

The hysteresis plane $P=\{p(p_1,p_2)\in R^2: v^+-a\geq p_2\geq p_1\geq v^-\}$ can be uniformly divided by l horizontal lines and l vertical lines into $N=(l+2)(l+1)/2$ small cells (see Fig.8.6) with coordinates (v_i,v_j) of their lower-left node as

$$\begin{cases} v_i = v^- + (i-1)\Delta v \\ v_j = v^- + (j-1)\Delta v \end{cases} \tag{8.17}$$

where $\Delta v = \dfrac{v^+ - v^-}{l+1}$, $j \geq i$, $i,j = 1,2,\ldots,l+1$;

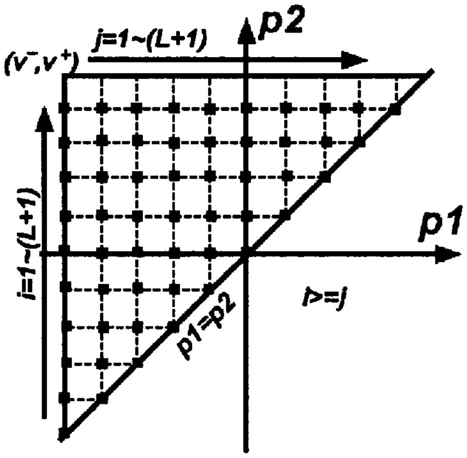


Fig. 8.6 Dividing of Preisach plane P

And then the total contribution of the elementary operators of each cell to the newly defined hysteresis model (8.1) is lumped into that of the operator corresponding to the lower-left node of the cell. Thus, one has

$$k_{p_{ij}}(t)\cdot\bar{\mu}_{p_i p_j} = \int_{\text{the } ij \text{ cell}} k_p(t)\cdot\mu_p dp \tag{8.18}$$

where $\bar{\mu}_{p_i p_j}$ is called the lumped density of a cell to its lower-left node (v_i,v_j). Finally, with (8.18) the newly defined hysteresis model (8.1) can be linearly parameterized as

$$u(t) = H[v](t) = \sum_{i=1}^{l+1} \sum_{j=1}^{i} \int_{the\ ij\ cell} k_p(t) \cdot \mu_p dp$$

$$= \sum_{i=1}^{l+1} \sum_{j=1}^{i} k_{p_{ij}}(t) \cdot \overline{\mu}_{p_i p_j} + d(v(t))$$

$$= \Gamma^T K + d(v(t)) \tag{8.19}$$

where $$K^T = [k_{p_{11}}, k_{p_{12}}, k_{p_{22}}, \ldots, k_{p_{1j}}, k_{p_{2j}}, \ldots k_{p_{jj}}, \ldots, k_{p_{(l+1)(l+1)}}] \tag{8.20}$$

and $$\Gamma^T = [\overline{\mu}_{p_{11}}, \overline{\mu}_{p_{12}}, \overline{\mu}_{p_{22}}, \ldots, \overline{\mu}_{p_{1j}}, \overline{\mu}_{p_{2j}}, \ldots \overline{\mu}_{p_{jj}}, \ldots, \overline{\mu}_{p_{(l+1)(l+1)}}] \tag{8.21}$$

and where K^T and Γ^T are $N = (l+2)(l+1)/2$ dimensional vectors. The term $d(v(t))$ in (8.19) is modeling error by the linearly parameterization of the model. If the number of dividing lines increases to extremely large value, the term $d(v(t))$ can be considered as zero.

Alternatively, the linearly parameterized newly defined hysteresis model (8.19) can also be expressed using the compensator m_p as

$$u(t) = H[v](t) = cv(t) - \sum_{i=1}^{l+1} \sum_{j=1}^{i} \int_{the\ ij\ cell} m_p(t) \cdot \mu_p dp$$

$$= cv(t) - \sum_{i=1}^{l+1} \sum_{j=1}^{i} m_{p_{ij}}(t) \cdot \overline{\mu}_{p_i p_j} + d(v(t))$$

$$= cv(t) - \Gamma^T M + d(v(t)) \tag{8.22}$$

where $$M^T = [m_{p_{11}}, m_{p_{12}}, m_{p_{22}}, \ldots, m_{p_{1j}}, m_{p_{2j}}, \ldots m_{p_{jj}}, \ldots, m_{p_{(l+1)(l+1)}}] \tag{8.23}$$

is an $N = (l+2)(l+1)/2$ dimensional vector consisting of output values of all the compensators associated with all lower-left nodes of all small cells.

Equations (8.19) and (8.22) are called the linearly parameterized hysteresis model. Although they can be modeled respectively, these two models are dependent and compensative to each other because they have the same density distribution (same density vector Γ) and have the relationship as

$$u(t) = cv(t) - \Gamma^T M + d(v(t)) = \Gamma^T K + d(v(t)) \tag{8.24}$$

The term $\Gamma^T M$ in the linearly parameterized hysteresis model (8.22) is very useful since it directly models the input-output error of hysteresis. And this error is critical for nonlinear robust adaptive control design since it can be treated as plant dynamics while the hysteretic actuator is considered as a linear transducer.

In order to conduct simulations of systems with input hysteresis which can be described by the newly defined hysteresis model (8.19) or (8.22), a simulation model has been developed in Simulink and is shown as in Appendix 6.

8.4 Simulation Studies

To illustrate the control methodology introduced in the section 2 of Chapter 7, in this section it is applied to a nonlinear system as

$$\begin{cases} \dot{x}(t) = \beta_1 x(t) + \beta_2 \dfrac{1 - e^{-x(t)}}{1 + e^{-x(t)}} + bu(t) \\ y = x \end{cases} \tag{8.25}$$

where $u(t)$ represents the output of an actuator with hysteresis. For the simulation, $\beta_1 = -2$, $\beta_2 = 1$ and $b = 6$ are chosen as their nominal values. The hysteresis model parameters ρ_{ij} for describing the input hysteresis of the nonlinear system are unknown, but their nominal values are assumed to be obtained from the functions

$$\rho_{ji} = \mu_{ji} / (\sum_{i}^{l+1} \sum_{j=1}^{i} \mu_{ji}) \tag{8.26}$$

$$\mu_{ji} = \frac{\gamma(v_j, v_i)}{\sum_{i=1}^{l+1} \sum_{j=1}^{i} \gamma(v_j, v_i)} \qquad (8.27)$$

and $$\gamma(v_j, v_i) = c_1 \exp\ (c_2((v_j - \bar{p}_1)^2 + (v_i - \bar{p}_2)^2)) \qquad (8.28)$$

with parameters $\sigma = 3.8$, $\bar{p}_1 = 0$, $\bar{p}_2 = 0$. The input region of hysteresis is $p \in [v^-, v^+] = [-6, 6]$. The hysteresis plane is divided by $l = 18$ horizontal lines and $l = 18$ vertical lines into $N = 190$ nodes. The coordinates v_j and v_i in (8.27) and (8.28) are calculated by (8.17). The slopes of the hysteresis are measured as $m_a = m_b = m_c = 0.1$. The sample time of the simulation is $t_s = 0.01$ sec and the simulation time $t = 30$ secs.

The objective is to control the above system state $x(t)$ to follow a desired trajectory $x_d(t)$ as
$$x_d(t) = -3.1[\cos(\pi t / 3) + \cos(\sqrt{2} \pi t / 3)]. \qquad (8.29)$$

Without adaptive control, the first investigation is the response of the open-loop system which includes the nonlinear plant (8.25) with the initial conditions of $x(0) = 0$ and the hysteretic actuator (8.19) as it is subjected to an input signal given by (8.29), i.e., $v(t) = x_d(t)$. The hysteresis loops of the actuator are formed as shown in Fig.8.7. The nonlinear compensator and its relationship with the hysteresis are shown in Fig. 8.8. Also, simulation studies show that the hysteresis loops generated by different frequencies of input signal $v(t)$ are similar to that in Fig.8.7. This means that the linearly parameterized hysteresis model with a fixed density distribution fully determines the property of the hysteretic actuator. The response of the open-loop system is shown in Fig.8.9, from which it is observed that the output $y(t)$ of the nonlinear plant is quite different from the command $v(t)$. In fact, the presence of the input hysteresis and the parameter uncertainties of the nonlinear plant make the response of the open-loop system very different from that of the reference command. Thus, it is necessary to design a robust adaptive controller to guarantee that the output of the nonlinear plant will follow the reference command.

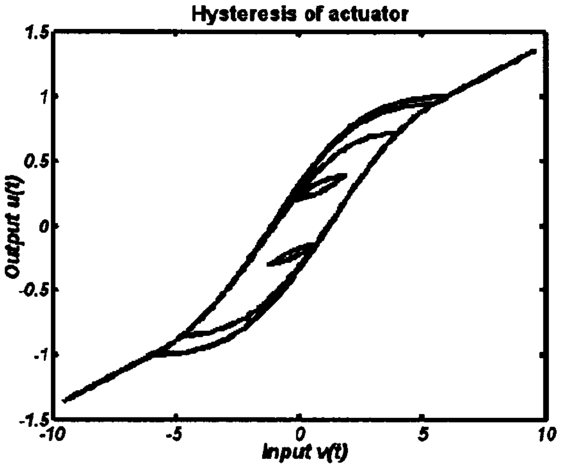


Fig. 8.7 Hysteresis of actuator

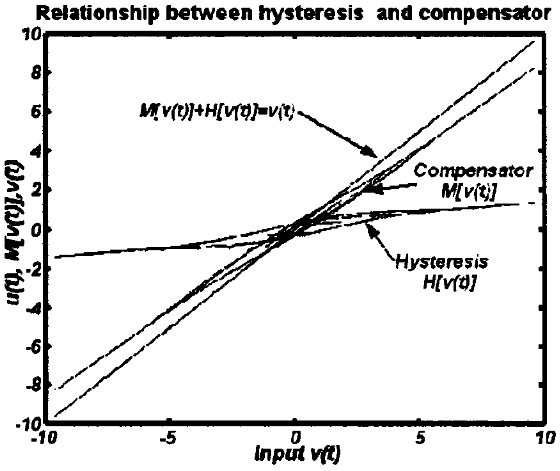


Fig. 8.8 Relationship between hysteresis and compensator

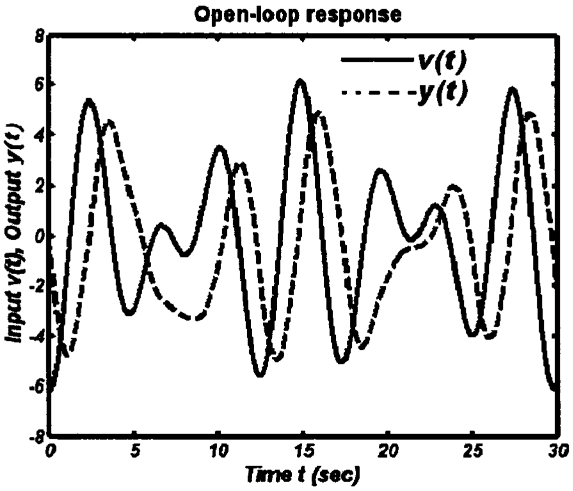


Fig. 8.9 Open-loop response

In the simulations of the closed-loop system, the robust adaptive control law used are those given by equations (7.17) ~ (7.24) as were designed in Chapter 7. The initial condition of the nonlinear plant is also chosen as $x(0)=0$. So far, no analytical approach has been developed for the selection of the control constants, but they are selected through iterative simulation. The control gain in the first term of the control law is selected as $k_d=10$, and the other control gains are set as $\gamma=2$, $\eta=2$, and $q=6$ to calculate the adaptive parameters θ, ϕ and ρ_n. The minimum and maximum values of the normalized densities ρ_n are set as 0 and 1. The bound for modeling error of hysteresis is assumed as $B=0.05$, which means the modeling error must less than 5% of the output multitude of the saturation states. $c_{\min}=0.95$ and $c_{\max}=1.0$. Thus, the control gain $k_\varepsilon=0.053$. The initial density distribution is chosen arbitrarily between these two extreme surfaces as $\rho_i=0$ and $\rho_j=1$where $i,j=1\sim N$. The initial values for the parameters $\theta(t)$and $\phi(t)$are set as $\theta(t_0)=1/3$and $\phi(t_0)=1/4$. The constraints for the tracking error are chosen as $\varepsilon=0.025$ and $\lambda=0.1$. The tracking performance using the robust adaptive control is shown in Fig.8.10. Simulation studies show that the system responses are more sensitive to control gains k_d and q. It is intuitive that q directly corrects the errors caused by the hysteresis.

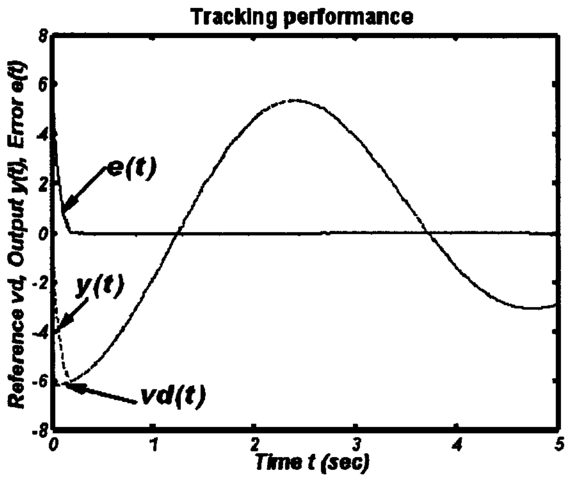


Fig.8.10 Tracking performance of closed loop

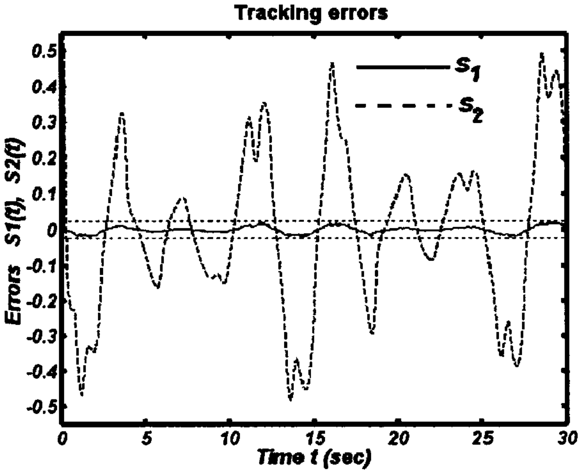


Fig.8.11 Tracking errors

To evaluate the effectiveness of the proposed adaptive robust control scheme, the simulation is also conducted for the closed-loop configuration without controlling the effects of the hysteresis, i.e., $\Gamma_n^T M=0$ in the controller (7.17). The tracking errors for the robust adaptive control with and without controlling the effects of the hysteresis are shown in Fig.8.11, where the tracking error (solid line) has been kept as

$s_1(t) = \widetilde{x}(t) < 0.025$, while the tracking error $s_2(t)$ (the dash-dot line) for $\Gamma_n^T M=0$ is larger than $s_1(t)$ and cannot satisfy the design requirement $s(t) \leq \varepsilon = 0.025$. Fig.8.11 shows that the signal $\Gamma_n^T M$ designed to reduce the hysteresis effects is necessary and effective, and also verifies the excellent tracking performance of the proposed algorithm.

It should be mentioned that simulations for several different desired trajectories with various parameter values and initial conditions have also been conducted for the applications of the newly defined hysteresis model and the robust adaptive control method introduced in Chapter7. Results show that they displayed similar behaviors as these shown in the simulations presented. The simulation for $l = 28$, which divides the hysteresis plane P into smaller cells, gives almost identical results but involves larger cost of calculation. This further verifies that the developed control algorithm is repeatable and computationally applicable. And it means that dividing the hysteresis plane P into extremely small sized cells is not necessary because it causes much calculation but only slightly improves the accuracy.

8.5 Conclusions

In this chapter, a modeling approach to the input hysteresis of systems is systemically presented, and a robust adaptive control architecture is proposed for a class of continuous-time nonlinear systems preceded by the hysteretic actuators. By showing the properties of the hysteresis model, a robust adaptive control scheme is developed without constructing the hysteresis inverse. The adaptive control law ensures global stability of the controlled system and achieves both stabilization and trajectory tracking within the desired precision. Simulation results performed on a simple nonlinear system illustrate and further validate the effectiveness of the proposed approach.

CHAPTER 9

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

9.1 Conclusions

This thesis has been dedicated to modeling and control of hysteresis of SMA actuators connected to systems. Extensive simulations have been conducted to validate the modeling approach and the control schemes.

The contribution of this thesis in the modeling aspect is the proposal of a linearly parameterized Preisach model, an alternative form of the linearly parameterized KP model, and a novel hysteresis model which overcomes the drawbacks of the Preisach model and the KP model.

The classical Preisach hysteresis model is widely used in the existing literatures. However, to implement the model in integration form, a suitable density distribution function for the hysteresis must be formulated in advance and the memory interface line for an applied input is required to be determined at every instance of time. These facts hinder the effective applications of the model. Motivated by the linear parameterized KP model defined by Banks, Kurdila and Webb in [41, 42], the Preisach model is rewritten as a linearly parameterized model, and this linear form of the Preisach model is suitable to be applied in adaptive control schemes. Thus, it becomes possible to conduct online identification and compensation of the hysteresis using the Preisach hysteresis model.

The linear parameterized KP mode is a linear combination of nonlinear elementary operators, called kernels. It is convenient to describe the Preisach class of hysteresis and to compensate the hysteresis using an inverse hysteresis model based on the linearly parameterized KP model. However, the findings of this thesis research shows that the newly proposed robust adaptive control of systems with input hysteresis, the inverse hysteresis model is not required to be constructed. Instead, an alternative form of the linearly parameterized KP model is defined to describe the error introduced by hysteretic actuators. In this alternative form of the KP model, an elementary compensator associating with the elementary hysteresis operator, kernel, is defined. The KP model can be expressed in the integral form of either weighted kernels or weighted elementary compensators. Also, the relationship between the KP hysteresis model and the Preisach model is revealed in this thesis. This relationship guarantees that the KP model has the same properties as the Preisach model and can describe the Preisach class of hysteresis.

Both the classical Preisach hysteresis model and the KP model have some drawbacks. Both models cannot describe the hysteresis with reversible parts. Also, the reverse curves of the hysteresis loops described by the models can only have zero initial slopes. To overcome these shortcomings of the models, a novel hysteresis model has been proposed in this thesis. This newly defined hysteresis model can easily describe the reversible parts of hysteresis, and can precisely model reverse curves with non-zero initial slopes. This new model is expressed in a linearly parameterized form so that it can be conveniently integrated with the adaptive control techniques.

The problems of control systems with input hysteresis have been pursued along three different paths: inverse compensation, gradient adaptive control and robust adaptive control for linear and nonlinear systems.

The idea of inverse compensation is to construct an inverse operator to eliminate or cancel the hysteretic nonlinearity in actuators. The inverse hysteresis model has been traditionally constructed by using either the classical Preisach operator method, the linearly parameterized Preisach operator method, or by the KP operator method. Inverse compensation of hysteresis is open-loop in nature and its performance is susceptible to model uncertainties and to errors introduced in the inversion process. For the classical Preisach hysteresis model, it is difficult to update the model parameters to adapt the effects caused from the uncertainties and dynamics of the actuator. For the linearly parameterized Preisach model, it is possible to incorporate adaptive control schemes to update model parameters. However, due to the discontinuity of the elementary relay operator, this model has extremely large amounts of parameters to be updated. Thus, the adaptive compensation would involve a large amount of computation and would affect the application of the control scheme in systems with more uncertainties and dynamics. The linearly parameterized KP operator method based on the inverse hysteresis model is more suitable for adaptive compensation. Due to the lack of effective inverse algorithms, a large amount of calculations and computation errors still exist in this inverse compensation framework.

To avoid the recursive approximation in the inverse algorithms, disturbances injected into systems by the hysteretic actuator is modeled by an elementary compensator. This modeled disturbance can be treated as the dynamics of the linear or nonlinear systems while the actuator is assumed as a linear transducer. Thus, the existing adaptive techniques can be used to control the dynamic systems. This alternative control framework has been successfully combined with the linearly parameterized KP model and the newly defined model to control nonlinear systems with input hysteresis. This scheme avoids the direct cancellation of the hysteresis nonlinearity of the actuator. In the adaptive control schemes, analysis for system stability and convergence of trajectory tracking has been conducted.

9.2 Recommendations for Future Work

In this thesis, hysteresis of SMA actuators is considered to be rate independent if the operation frequency or the change rate of input is low enough, for example, less than 10 Hz. However, in some realistic cases, the operation frequency may be considerably high and vary with time; consequently, the hysteresis of SMA actuators is a rate dependent nonlinearity. Thus, in the future work, the problem of defining dynamic hysteresis models to describe the rate dependent hysteresis will be involved, and adaptive control techniques combined with the dynamic hysteresis model will be examined to control systems preceded by dynamic hysteretic actuators.

APPENDICES

Appendix 1

```
%%%%%%%%%%%%%%% main function %%%%%%%%%%%%%%%%%
%% to find interface lines and draw the hysteresis loops %%
clear all
close

%% acquits input value sequence %%
%p= [-185 145 -145  145  -145 105 -105 75 -75 45 -45 20 -30];  %case1%
%p= [-185 185 -185 145 -145  145  -145 105 -105 75 -75 45 -45 20];      %case2%
p=[-185 185 -185 145 -145 105 -100 75 -45 80 -30 20];                   %case3%

Umin=-185;
Umax=185;
yy0=-17;
m=1000;
yti=0;
ytd=0;

% call function 'interface_loops'%
interface_loops(p,Umax,Umin,yy0,m,yti,ytd);
%%%%%%%%%%%%%%% end of main function %%%%%%%%%%%%%%%

%%%%%% interface_loops%%%%%%%%%%%%%
function interface_loops(p,Umax,Umin,yy0,m,yti,ytd)
if p(1)<=p(2)
   %call function ' start_from_negative_state'%
   [AA,BB]=start_from_negative_state(p);
else
   %call function ' start_from_positive_state'%
   [AA,BB]=start_from_positive_state(p);
end

subplot(1,2,1)
memory(AA,BB,Umax,Umin);

subplot(1,2,2)
%call function 'loops'%
loops(AA,BB,Umax,Umin,yy0,m,yti,ytd);
%%%%%% end of function interface_loops %%%%%

%%%%% start_from_positive_state  %%%%%
function [AA,BB]=start_from_positive_state(p)
[lp,kp]=size(p);
en=p(kp);
kkp=kp;
for ih=1:kkp
   [CA,IA]=max(p);
   AA(ih)=CA;
   if AA(ih)==en
      break;
   end
   for i=1:kp-IA
      pp(i)=p(IA+i);
   end
   p=pp;
   pp=NaN;
   [lp,kp]=size(p);

   [CB,IB]=min(p);
   BB(ih)=CB;
   if BB(ih)==en
      break;
   end
   for i=1:kp-IB
```

```
      pp(i)=p(IB+i);
   end
   p=pp;
   pp=NaN;
   [lp,kp]=size(p);
end
%%%% end of function "start_from_negative_state" %%%%%%%%

%%%%% start_from_negative_state  %%%%%
function [AA,BB]=start_from_negative_state(p)
[lp,kp]=size(p);
en=p(kp);
kkp=kp;
for ih=1:kkp
   [CB,IB]=min(p);
   BB(ih)=CB;
   if BB(ih)==en
      break;
   end
   for i=1:kp-IB
      pp(i)=p(IB+i);
   end
   p=pp;
   pp=NaN;
   [lp,kp]=size(p);

   [CA,IA]=max(p);
   AA(ih)=CA;
   if AA(ih)==en
      break;
   end
   for i=1:kp-IA
      pp(i)=p(IA+i);
   end
   p=pp;
   pp=NaN;
   [lp,kp]=size(p);
end

%check the end element is an>bn to decide add a bn+1 to p vector
[AAl,AAk]=size(AA);
[BBl,BBk]=size(BB);
if AAk==BBk
   BB(BBk+1)=AA(AAk);
end
%%%% end of function "start_from_negative_state" %%%%%%%%

%%%% function memory%%%%%%%%%%%%%%
function memory(AA,BB,Umax,Umin)
av=[Umin,Umin,Umax,Umin];
bv=[Umin,Umax,Umax,Umin];
% draw a plot of memory curve in support triangle
line(av,bv, 'linewidth', 4, 'color',[1 0 0]);
hold;
grid on

[l,k]=size(AA);
BX(2*k+1)=BB(k+1);
AY(2*k+1)=BB(k+1);

for i=1:k
    BX(2*i-1)=BB(i);
    BX(2*i)=BB(i+1);
end

for i=1:k
    AY(2*i-1)=AA(i);
    AY(2*i)=AA(i);
end
line(BX, AY, 'linewidth', 2, 'color',[0 1 0]);
```

```
%%%% end of function memory %%%%%%%%%%

%%%%%%%% function yab(a,b) %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function y=yab(a,b)
y=34.7239*(exp(-0.033*(-185+38.473))-exp(-0.033*(b+38.473)))/((1+exp(-0.026*(a-16.773)))*(1+exp(-
0.033*(b+38.473)))*(1+exp(-0.033*(-185+38.473))))-17;
%%%%%%%% end of function yab(a,b) %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%% function loops %%%%%%%%%%%%%
function loops(AA,BB,Umax,Umin,yy0,m,yti,ytd)
[lA,kA]=size(AA);
[lB,kB]=size(BB);

for ip=1:kA
   p(2*ip-1)=BB(ip);
   p(2*ip)=AA(ip);
end
p(2*kA+1)=BB(kB);
[pl,pk]=size(p);

      %% draw major loop %%%%%%%%%%%%%%%%
      u=Umin:(Umax-Umin)/m:Umax;
      for jj=1:m+1
          yyi(jj)=yab(u(jj),u(jj));
      end
      plot(u,yyi,'-', 'linewidth', 2);
      hold;
      grid on;
      u=Umax:(Umin-Umax)/m:Umin;

      for jj=1:m+1
         yyd(jj)=yab(Umax,u(jj));
      end
      plot(u,yyd,'-', 'linewidth', 2);
     %% end of draw major loop %%%%%%%%%%%%%

if p(pk)==p(pk-1)
   for kk=1:(pk-1)/2-1
      u=p(2*kk-1):(p(2*kk)-p(2*kk-1))/m:p(2*kk);
      for jj=1:m+1
         yyi(jj)=yyd(m+1)+yab(u(jj),u(jj))-yab(u(jj),p(2*kk-1));
      end
      plot(u,yyi,'-', 'linewidth', 2);

      u=p(2*kk):(p(2*kk+1)-p(2*kk))/m:p(2*kk+1);
      for jj=1:m+1
         yyd(jj)=yyi(m+1)-yab(p(2*kk),p(2*kk))+yab(p(2*kk),u(jj));
      end
      plot(u,yyd,'-', 'linewidth', 2);
   end

   kk=(pk-1)/2;
   u=p(2*kk-1):(p(2*kk)-p(2*kk-1))/m:p(2*kk);
   for jj=1:m+1
      yyi(jj)=yyd(m+1)+yab(u(jj),u(jj))-yab(u(jj),p(2*kk-1));
   end
   plot(u,yyi,'r-', 'linewidth', 2);

      %% calculate the output at end of process %%%%%%%%%%%%%%%%%%%%%%
      for kk=1:(pk-1)/2
          yt(kk)=yab(p(2*kk),p(2*kk+1))-yab(p(2*kk),p(2*kk-1));
      end
      yt=yy0+sum(yt)
      plot(u(m+1),yt,'b*');
     %% end of calculate the output at the end of process%%%%%%%%%%%

else
   for kk=1:(pk-1)/2
      u=p(2*kk-1):(p(2*kk)-p(2*kk-1))/m:p(2*kk);
      for jj=1:m+1
      yyi(jj)=yyd(m+1)+yab(u(jj),u(jj))-yab(u(jj),p(2*kk-1));
    end
    plot(u,yyi,'-', 'linewidth', 2);

    u=p(2*kk):(p(2*kk+1)-p(2*kk))/m:p(2*kk+1);
    for jj=1:m+1
      yyd(jj)=yyi(m+1)-yab(p(2*kk),p(2*kk))+yab(p(2*kk),u(jj));
    end
    plot(u,yyd,'-', 'linewidth', 2);
  end

  %%calculate the output at end of process%%%%%%%%%%
  for kk=1:(pk-1)/2
    yt(kk)=yab(p(2*kk),p(2*kk+1))-yab(p(2*kk),p(2*kk-1));
  end
  yt=yy0+sum(yt)
  plot(u(m+1),yt,'r*');
  %%end of calculate the output at end of process%%%%%%%%

end
%%%%%%% end of function loops %%%%%%%%%%%%%%%
```

Appendix 2

```
%%%%%%%%%%%%%%%%%%%%%%main part%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%find input when knowing desired output%%%%%%%%%%
clear all;
close
global umax umin stp y

load yab_999lines;
m=169;
n=2;

tim=0:n:m;
yd=17*(-0.7602+185*(0.5*sin(pi*(tim/30-0.5))+0.5*cos(pi*(tim/30/2^0.5-1))))/185.7602;
yd(1)=-17;

u=find_process_input(yd);
[ul,uk]=size(u);
ytt(1)=-17;

for tt=2:uk
p=NaN;
p=u(1:tt);
[AA,BB]=start_from_negative_state(p);
ytt(tt)=yt_end_loops(AA,BB);
end

subplot(2,2,1);
plot(tim,u);
grid on;

subplot(2,2,2);
plot(tim, ytt,'b-', 'linewidth', 1);
grid on;

subplot(2,2,3);
plot(yd,ytt);
grid on;

subplot(2,2,4);
plot(u,ytt,'r-', 'linewidth', 1);
grid on;
%%%%%%%%%%%%%%%%% end of main part%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%
```

```
function u=find_process_input(yd)
u(1)=-185;
[lPd, iPd]=size(yd);

for tt=2:iPd
   Pi=yd(1:tt);
   [AA,BB]=start_from_negative_state(Pi);
   [Laa,Kaa]=size(AA);
    P=NaN;

   if AA(Kaa)==BB(Kaa+1)
     for Pj=1:Kaa
        P(2*Pj-1)=BB(Pj);
        P(2*Pj)=AA(Pj);
     end

   else

     for Pj=1:Kaa
        P(2*Pj-1)=BB(Pj);
        P(2*Pj)=AA(Pj);
     end
     P(2*Kaa+1)=BB(Kaa+1);

   end
   [lp,i_p]=size(P);
   u(tt)=find_moment_input(P);
end


%%%%%%%%%%%%%%%%%%%%
function a=find_moment_input(P)
[lp,i_p]=size(P);
if i_p==2
  bn=-185;
  yn=P(1);    %yn=-16;
  ytt=P(2);    %ytt=18;
  [y_t,a_t]=up(yn,bn,ytt);
  a=a_t;
elseif i_p==3
  bn=-185;
  yn=P(1);    %yn=-16;
  ytt=P(2);    %ytt=18;
  [y_t,a_t]=up(yn,bn,ytt);

  an=a_t;
  yn=P(2);
  ytt=P(3);
  [y_t,b_t]=down(an, yn, ytt);
  a=b_t;
elseif i_p>3
  bn=-185;
  yn=P(1);    %yn=-16;
  ytt=P(2);    %ytt=18;
  [y_t,a_t]=up(yn,bn,ytt);

  an=a_t;
  yn=P(2);
  ytt=P(3);
  [y_t,b_t]=down(an, yn, ytt);

  for ip=2:round((i_p-1)/2)
     bn=b_t;
     yn=P(2*ip-1);
     ytt=P(2*ip);
     [y_t,a_t]=up(yn,bn,ytt);
     if 2*ip==i_p
       a=a_t;
       break;
     end
    an=a_t;
    yn=P(2*ip);
    ytt=P(2*ip+1);
    [y_t,b_t]=down(an, yn, ytt);
    a=b_t;
  end
end


%%%%%%%%%%%%%%%%%%%%%%%%%%
function [AA,BB]=start_from_negative_state(p)
%clear all;
%close
%p=[185 -184 105 -100 75 -45 80 -30]
[lp,kp]=size(p);
en=p(kp);
kkp=kp;
      for ih=1:kkp
      [CB,IB]=min(p);
      BB(ih)=CB;
      if BB(ih)==en
        break;
      end
      for i=1:kp-IB
         pp(i)=p(IB+i);
      end
      p=pp;
      pp=NaN;
      [lp,kp]=size(p);

      [CA,IA]=max(p);
      AA(ih)=CA;
      if AA(ih)==en
        break;
      end
      for i=1:kp-IA
        pp(i)=p(IA+i);
      end
      p=pp;
      pp=NaN;
      [lp,kp]=size(p);
   end

%check the end element is an>bn to decide add a bn+1 to p vector
[AAl,AAk]=size(AA);
[BBl,BBk]=size(BB);
if AAk==BBk
   BB(BBk+1)=AA(AAk);
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [y_t,a_t]=up(yn,bn,ytt)
%load yab_999lines;
a=umax:-stp:umin;
b=umin:stp:umax;
ytt=yn:(ytt-yn)/99:ytt; % discrete the expected outputs;
deta_y=ytt-yn;

ii=round(1-(umin-bn)/stp);
jj=1001-ii;
for i_up=1:100
   while y(jj,1001-jj)-y(jj,ii)<deta_y(i_up)
      jj=jj-1;
   end
   an(i_up)=a(jj);
   yt_t(i_up)=yab(an(i_up),an(i_up))-yab(an(i_up),bn)+yn;
end
y_t=ytt(100);
a_t=an(100);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [yt]=yt_end_loop(AA,BB)
yy0=-17;
m=1000;
yti=0;
ytd=0;
[lA,kA]=size(AA);
[lB,kB]=size(BB);

for ip=1:kA
   p(2*ip-1)=BB(ip);
   p(2*ip)=AA(ip);
end
p(2*kA+1)=BB(kB);
[pl,pk]=size(p);

%%%Calculate the output at end of process%%%
for kk=1:(pk-1)/2
yt(kk)=yab(p(2*kk),p(2*kk+1))-yab(p(2*kk),p(2*kk-1));
end
yt=yy0+sum(yt);
```

Appendix 3

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all;
close
%n=209.9;
n=40;
m=0.1;
t=0:m:n;
%v=-2*(cos(pi*t/15)+cos(pi*2^0.5 *t/15));
v=-4*cos(pi*t/20);
plot(t,v);

[kl,ki]=size(v);
vv(1)=v(1);
vv(2)=v(1);
for i=3:ki+2
   vv(i)=v(i-2);
end
v=vv;
kn=5;
a=4/kn;
%s1=-2:4/(kn-1):2;
s1=-2:4/kn:(2-a);
s2=s1;
for i=1:kn
   for j=1:i
      ru=1.0;
      c1=1/(2*pi*ru^2);
      c2=-1/(2*ru^2);
      gamma(i,j)=c1*exp(c2*(s1(j)^2+s2(i)^2));
   end
end
sum_gamma=sum(sum(gamma),2);
for i=1:kn
   for j=1:i
      gamma(i,j)=gamma(i,j)/sum_gamma;
   end
end
for i=1:kn
   for j=1:i
      ww(i,j,:)=element(s1(j),s2(i),a,v);
   end
```

```
end
for j1=1:kn
   for j2=1:j1
      for ji=1:n/m+3
         AA(ji)=ww(j1,j2,ji);
      end
      subplot(kn,kn,(kn-j1)*kn+j2)    %draw elements in support triangle
      plot(v,AA,'-','linewidth',2);
      axis([-4 4 -1.1 1.1])
      AA=0;
   end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all;
close
%n=209.9;
n=40;
m=0.1;
t=0:m:n;
%v=-2*(cos(pi*t/15)+cos(pi*2^0.5 *t/15));
v=-4*cos(pi*t/20);

[kl,ki]=size(v);
vv(1)=v(1);
vv(2)=v(1);

for i=3:ki+2
    vv(i)=v(i-2);
end

v=vv;
l=4;
kn=l+1;
a=4/kn;
s1=-2:4/kn:(2-a);
s2=s1;

for i=1:kn
    for j=1:i
        ru=1.0;
        c1=1/(2*pi*ru^2);
        c2=-1/(2*ru^2);
        gamma(i,j)=c1*exp(c2*(s1(j)^2+s2(i)^2));
    end
end
sum_gamma=sum(sum(gamma),2);

for i=1:kn
    for j=1:i
          gamma(i,j)=gamma(i,j)/sum_gamma;
     end
end

for i=1:kn
    for j=1:i
        ww(i,j,:)=gamma(i,j)*element(s1(j),s2(i),a,v);
    end
end

for j1=1:kn
    for j2=1:j1
        for ji=1:n/m+3
            AA(ji)=ww(j1,j2,ji);
        end
        AA=0;
    end
end
```

```
w=cumsum(cumsum(ww),2);
for ji=1:n/m+3
    wa(ji)=w(kn,kn,ji);
end
plot(v,wa,'r-', 'linewidth', 2);
axis([-4 4 -1.1 1.1]);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

S-function

Loading parameters by calling m.file “gamma_kpsf”

```
%%% %%%%%%%%%% gamma_kpsf.m %%%%%%%%%%%
%%%used to calculate and upload parameters for s-function %%%
%%%%%%%%"HYSTERESIS" in the simulation block%%%%%
clear all;
close;

%%%give conditions about hysteresis and density function and discretization level%%
umin=-2;        %give negative saturation input value;
umax=2;         %give positive saturation input value;
kn=22;          %discretization level;
ru=1.0;         %parameter of density distrbution function;

[s1,a]=discretization(kn,umax,umin);%call function 'discretization' to calculate 's1','s2', 'a';
[gamma]=density(kn,s1,ru);%call function 'density' to calculate normalized density 'gamma';

%%%%%%%%% end of m.file “gamma_kpsf” for loading parameter %%%%%%%%%%%
```

Two subroutines called by the m.file “gamma_kpsf”:

```
%%##############################################%%
function [s1,a]=discretization(kn,umax,umin)
s1=umin:(umax-umin)/(kn-1):umax;
a=(umax-umin)/(kn-1);
%%##############################################%%

%%##################################################################%%
function [gamma]=density(kn,s1,ru)
c1=1/(2*pi*ru^2);
c2=-1/(2*ru^2);

for i=1:kn
  for j=1:i
    gamma(i,j)=c1*exp(c2*(s1(j)^2+s1(i)^2));%calculation of density distribution
    %gamma(i,j)=1;
  end
end

sum_gamma=sum(sum(gamma),2);%summation of all density factors

for i=1:kn
  for j=1:i
    gamma(i,j)=gamma(i,j)/sum_gamma;%normalized density distribution
  end
end
%%#################################################################%%
```

S-function: H

```
function [sys,x0,str,ts] = H(t,x,u,flag,kn,gamma,umin,a,s1)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                  H                %
% This S-function computes an output value of a hysteresis system.  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Dispatch the flag. The switch function controls the calls to
% S-function routines at each simulation stage of the S-function.
v=[0];
switch flag,
   %%%%%%%%%%%%%%%%%%
   % Initialization %
   %%%%%%%%%%%%%%%%%%
   % Initialize the states, sample times, and state ordering strings.
  case 0
       [sys,x0,str,ts]=mdlInitializeSizes;


   %%%%%%%%%%%
   % Outputs %
   %%%%%%%%%%%
   % Return the outputs of the S-function block.
  case 3
     sys=mdlOutputs(t,x,u,kn,gamma,umin,a,s1);

   %%%%%%%%%%%%%%%%%%%
   % Unhandled flags %
   %%%%%%%%%%%%%%%%%%%
   % There are no termination tasks (flag=9) to be handled.
   % Also, there are no continuous or discrete states,
   % so flags 1,2, and 4 are not used, so return an emptyu
   % matrix
   case { 1, 2, 4, 9 }
     sys=[];

   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   % Unexpected flags (error handling)%
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   % Return an error message for unhandled flag values.
   otherwise
   error(['Unhandled flag = ',num2str(flag)]);
end
% end timestwo

%=============================================================
% mdlInitializeSizes
% Return the sizes, initial conditions, and sample times for the S-function.
%=============================================================
function [sys,x0,str,ts] = mdlInitializeSizes()

sizes = simsizes;
sizes.NumContStates  = 0;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = -1;  % dynamically sized
sizes.NumInputs      = -1;  % dynamically sized
sizes.DirFeedthrough = 1;   % has direct feedthrough
sizes.NumSampleTimes = 1;

sys = simsizes(sizes);
str = [];
x0  = [];
ts  = [-1 0];   % inherited sample time
% end mdlInitializeSizes

%==========================================
% mdlOutputs
% Return the output vector for the S-function
%==========================================
function sys = mdlOutputs(t,x,u,kn,gamma,umin,a,s1)

%store sampling time vector and input vector
if t==0
  v=u; % pass current input 'u' from outside to the inside input variabl 'v'
  t=0;
  tt=t;
  save('t1','t','-ascii','-double');% save 't' as 't1'
```

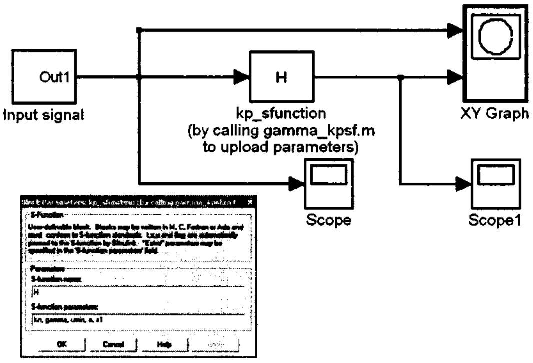
```
else
  v=load('vva');
  [vl,vk]=size(v);

  if (u<=umin)&(vk>=2)&(v(vk-1)>v(vk))&(v(vk)<u);
    v=u;
    save('vva','v','-ascii','-double'); % save 'v' as 'vva'
    v=load('vva');
    save('t1','t','-ascii','-double');  %save 't' as 't1'
  end

  t1=load('t1');
  tt=t-t1;

  if tt==0
    v=u;
  else
    v=[v,u];
  end

end
save('vva','v','-ascii','-double'); %save 'v' as'vva'

%use stored time vector and input vector to calculate the operators' output
sys=hh(v,tt,kn,gamma,a,s1);
% end mdlOutputs

%%%%%%% end of the s-function "H"%%%%%%%%%%%%%%%
```

Some subroutines called by the S-function "H"

```
%%##################################################%%
function sys=hh(v,t,kn,gamma,a,s1)
m=1;
[kl,ki]=size(v);
vv(1)=v(1);
vv(2)=v(1);
for i=3:ki+2
   vv(i)=v(i-2);
end
v=vv;

for i=1:kn
   for j=1:i
      ww(i,j,:)=gamma(i,j)*element(s1(j),s1(i),a,v);
   end
end

for j1=1:kn
   for j2=1:j1
       for ji=1:t/m+3
          AA(ji)=ww(j1,j2,ji);
       end
       AA=0;
    end
end
w=cumsum(cumsum(ww),2);

for ji=1:t/m+3
   wa(ji)=w(kn,kn,ji);
end

for iv=3:ki+2
   [lwa,kwa]=size(wa);
   waa(iv-2)=wa(iv);
end
 [il,ki]=size(waa);
```

```
sys=waa(ki);
%%##############################################%%

%%##############################################%%
function w=element(s1,s2,a,v)
[kl,ki]=size(v);
%w(1)=1;
%w(2)=1;
%xi_s=1;
w(1)=-1;
w(2)=-1;
xi_s=-1;

for i=3:ki
  if ((v(i)>=v(i-1))&(v(i-1)<=v(i-2)))|((v(i)<=v(i-1))&(v(i-1)>=v(i-2)))
    xi_s=w(i-1);
  end

  if v(i)<v(i-1)
    r1=r(s1,a,v(i));
    w(i)=min(xi_s, r1);

  elseif v(i)>=v(i-1)
    r2=r(s2,a,v(i));
    w(i)=max(xi_s, r2);
  end
end
%%##############################################%%
```

"H" block by calling the s-function " H"



S-Function

User-definable block. Blocks may be written in M, C, Fortran or Ada and must conform to S-function standards. t,x,u and flag are automatically passed to the S-function by Simulink. "Extra" parameters may be specified in the 'S-function parameters' field.

Parameters

S-function name: H

S-function parameters: kn, gamma, umin, a, s1

Fig.1A "H" block by calling the s-function " H"

Appendix 4

4.1 Simulation model of "KP hysteresis"

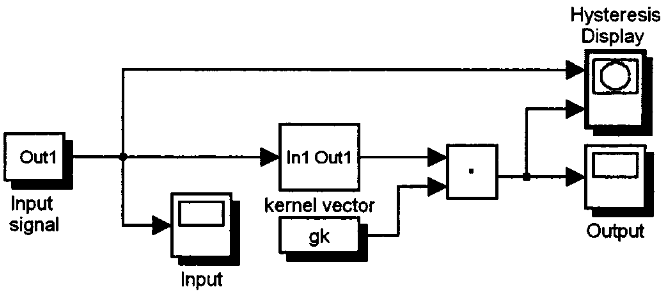


Fig. 2A Simulation model of hysteresis with KP representation

where the "kernel vector" block can be combined with the operator "dot production" as a "Hysteresis" block.

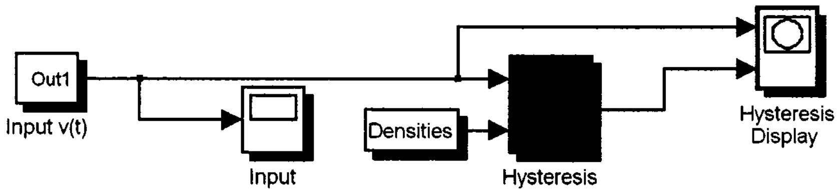


Fig. 2A Simulation model of hysteresis with KP representation

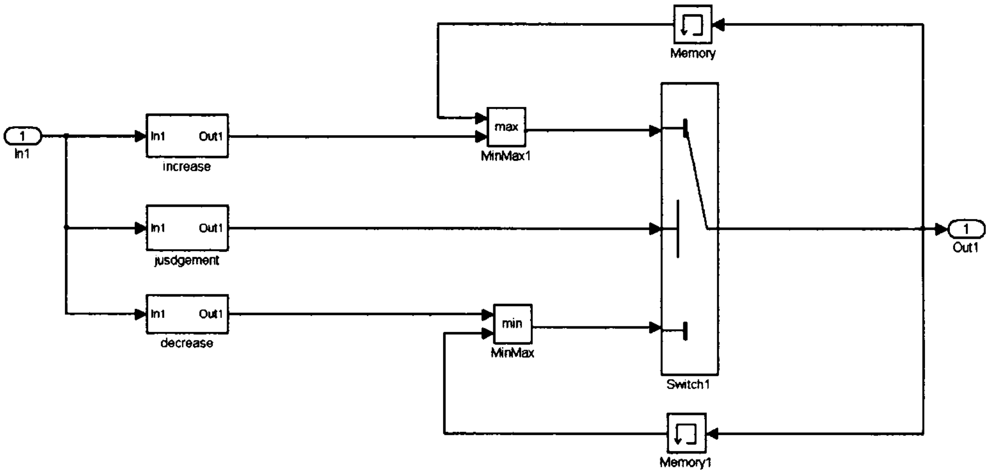


Fig. 3A the "kernel vector subsystem" in Fig.2A

The kernel vector subsystem (Fig.3A) includes following subsystems:

a. Ascending ridge function:

$$r_2 = \begin{cases} -1 & v(t) < p_2 \\ -1 + \dfrac{2}{a} \cdot [v(t) - p_2] & p_2 \le v(t) \le p_2 + a \\ 1 & v(t) > p_2 + a \end{cases}$$

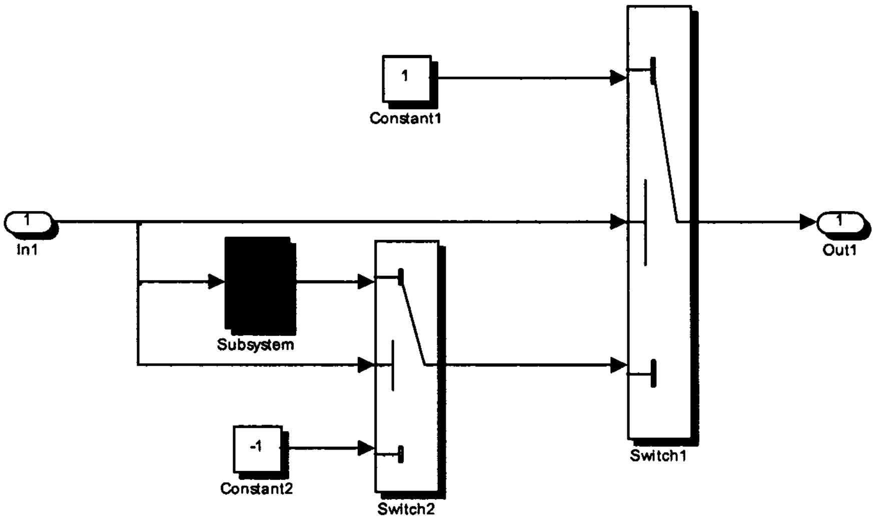


Fig. 4A the ascending boundary of a kernel

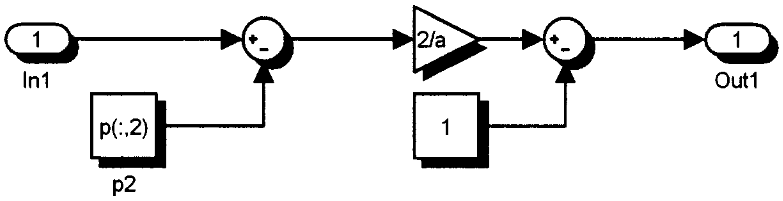


Fig. 5A the "Subsystem" in Fig. 4A

b. Descending ridge function:

$$r_1 = \begin{cases} -1 & v(t) < p_1 \\ -1 + \dfrac{2}{a}\cdot[v(t) - p_1] & p_1 \le v(t) \le p_1 + a \\ 1 & v(t) > p_1 + a \end{cases}$$

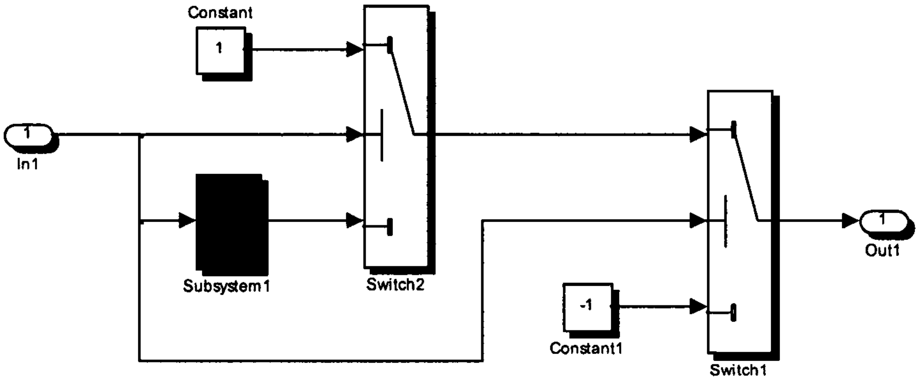


Fig. 6A the descending boundary of a kernel

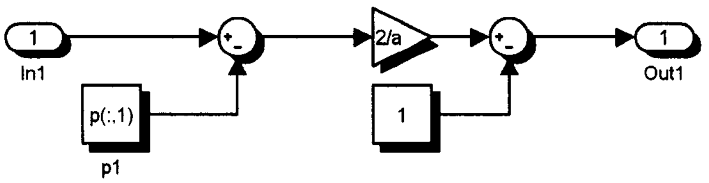


Fig. 7A the "Subsystem 1" in Fig. 6A

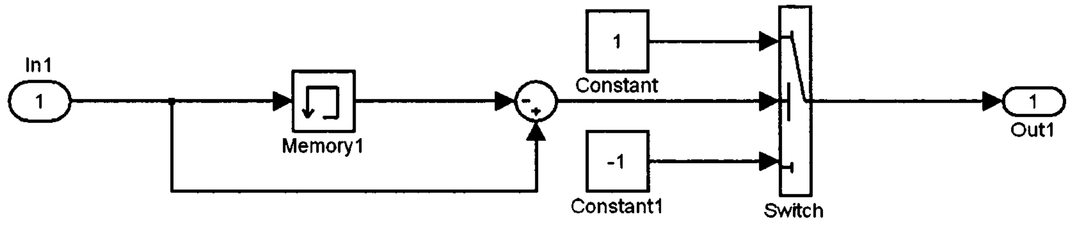


Fig. 8A Judgment subsystem in Fig. 3A
(To judge the ascending or descending trend of input)

4.2 Simulation model of "inverse KP hysteresis"

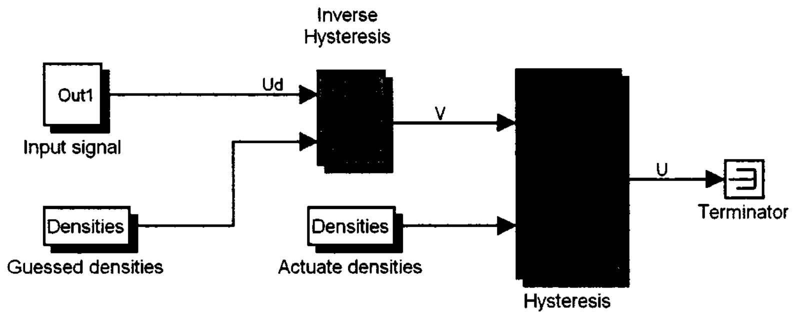


Fig. 9A Simulation model for open-loop compensation with inverse KP hysteresis model

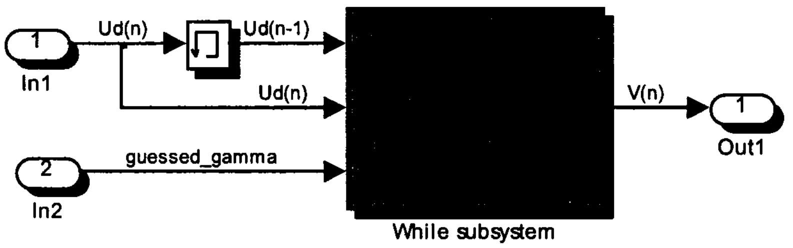


Fig. 10A Simulation model of the subsystem "inverse hysteresis" in Fig. 9A

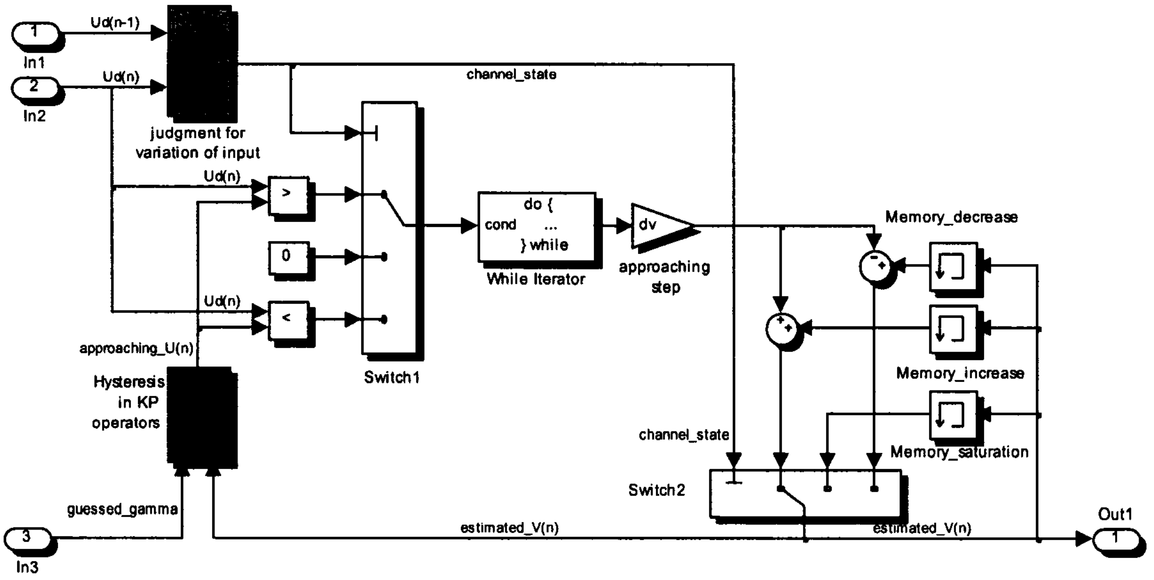


Fig. 11A The "while subsystem" in Fig. 10A

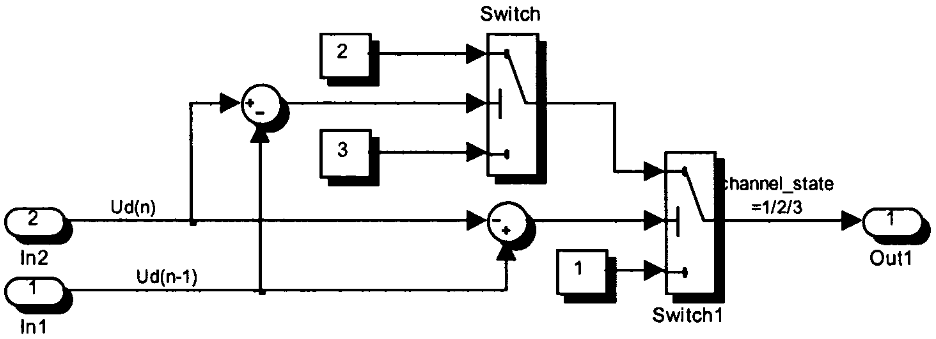


Fig. 12A The "judgment subsystem" in Fig. 11A

Appendix 5

5.1 *Norms and L_p Spaces*

Definition 5.1.1 *The* **norm** $|x|$ *of a vector* x *is a real valued function with the following properties:*

(i) $|x| \geq 0$ *with* $|x| = 0$ *if and only if* $x = o$

(ii) $|\alpha x| = |\alpha| \, |x|$ *for any scalar* α

(iii) $|x + y| \leq |x| + |y|$ *(triangle inequality)*

The norm $|x|$ of a vector x can be thought of as the size or length of the vector x. Similarly, $|x - y|$ can be thought of as the distance between the vectors x and y.

An $m \times n$ matrix A represents a linear mapping from n-dimensional space R^n into m-dimensional space R^m. The induced norm of A is defined as follows:

Definition 5.1.2 *Let* $|\cdot|$ *be a given vector norm. Then for each matrix* $A \in R^{m \times n}$, *the quantity* $\|A\|$ *defined by*

$$\|A\| \overset{\Delta}{=} \sup_{x \neq 0, x \in R^n} \frac{|Ax|}{|x|} = \sup_{|x| \leq 1} |Ax| = \sup_{|x|=1} |Ax|$$

is called the **induced (matrix) norm** *of* A *corresponding to the vector norm* $|\cdot|$.

The induced matrix norm satisfies the properties (i) to (iii) of Definition 5.1.1. Some other properties of the **induced norm** are summarized as follows:

(a) $|Ax| \leq \|A\| \, |x|, \; \forall x \in R^n$

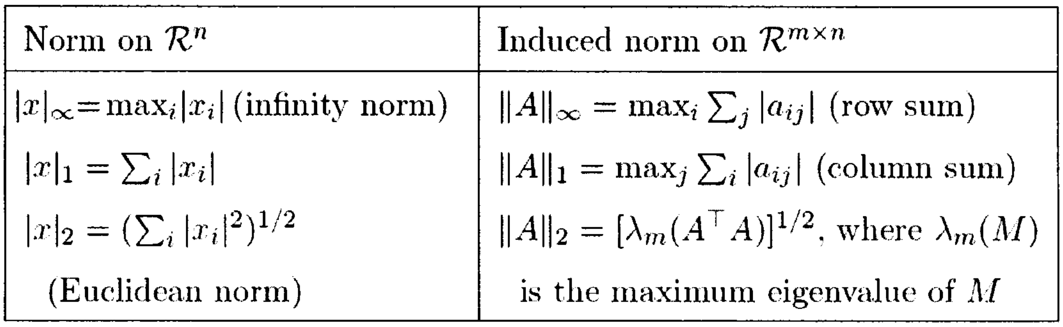
(b) $\|A + B\| \leq \|A\| + \|B\|$

(c) $\|AB\| \leq \|A\| \|B\|$

where A, B are arbitrary matrices of compatible dimensions. Table 1A shows some of the most commonly used norms on R^n.

It should be noted that the function $\| A \|_s = \max_{ij} | a_{ij} |$, where $A \in R^{m \times n}$ and a_{ij} is the (i, j) element of A satisfies the properties (i) to (iii) of Definition 5.1.1. It is not, however, an induced matrix norm because no vector norm exists such that $\| \cdot \|_s$ is the corresponding induced norm. Note that $\| \cdot \|_s$ does not satisfy property (c).

Table 1A Commonly used norms



Norm on $\mathcal{R}^n$	Induced norm on $\mathcal{R}^{m \times n}$
$\lvert x \rvert_\infty = \max_i \lvert x_i \rvert$ (infinity norm)	$\lVert A \rVert_\infty = \max_i \sum_j \lvert a_{ij} \rvert$ (row sum)
$\lvert x \rvert_1 = \sum_i \lvert x_i \rvert$	$\lVert A \rVert_1 = \max_j \sum_i \lvert a_{ij} \rvert$ (column sum)
$\lvert x \rvert_2 = (\sum_i \lvert x_i \rvert^2)^{1/2}$	$\lVert A \rVert_2 = [\lambda_m(A^\top A)]^{1/2}$, where $\lambda_m(M)$
(Euclidean norm)	is the maximum eigenvalue of M

Definition 5.1.3 L_p **norm**: For functions of time, the L_p norm is defined as follow

$$\| x \|_p \overset{\Delta}{=} \left(\int_0^\infty | x(\tau) |^p d\tau \right)^{1/p}$$

for $p \in [1, \infty)$ and say that $x \in L_p$ when $\| x \|_p$ exists (i.e., when $\| x \|_p$ is finite).

Definition 5.1.4 L_∞ **norm**: The L_∞ norm is defined as

$$\| x \|_\infty \overset{\Delta}{=} \sup_{t \geq 0} | x(t) |$$

and it can be said that $x \in L_\infty$ when $\| x \|_\infty$ exists.

In the above L_p, L_∞ norm definitions, $x(t)$ can be a scalar or a vector function. If x is a scalar function, then $|\cdot|$ denotes the absolute value. If x is a vector function in R^n then $|\cdot|$ denotes any norm in R^n.

Similarly, for sequences the l_p norm can be defined as

$$\| x \|_p \overset{\Delta}{=} (\sum_{i=1}^{\infty} | x_i |^p)^{1/p}, \quad 1 \le p < \infty$$

and the l_∞ norm as

$$\| x \|_\infty \overset{\Delta}{=} \sup_{i\ge 1} | x_i |$$

where $x = (x_1, x_2, ...)$ and $x_i \in R$. It can be said that $x \in l_p$ (respectively $x \in l_\infty$) if $\| x \|_p$ (respectively $\| x \|_\infty$) exists.

To handle classes of functions of time that do not belong to L_p. It is necessary to define the L_{pe} norm.

Definition 5.1.5 L_{pe} norm: the L_{pe} norm is defined as follow

$$\| x_t \|_p \overset{\Delta}{=} (\int_0^t | x(\tau) |^p \, d\tau)^{1/p}$$

for $p \in [1, \infty)$ and it can be said that $x \in L_{pe}$ when $\| x_t \|_p$ exists for any finite t.

Definition 5.1.6 $L_{\infty e}$ norm : The norm $L_{\infty e}$ is defined as follow

$$\| x_t \|_\infty \overset{\Delta}{=} \sup_{0 \le \tau \le t} | x(\tau) |$$

The function t^2 does not belong to L_p but $t^2 \in L_{pe}$. Similarly, ***any continuous function of time belongs to L_{pe} but it may not belong to L_p.***

Definition 5.1.7** L_p **space and** L_{pe} **space: For each $p \in [1,\infty)$, the set of functions that belong to L_p (respectively L_{pe}) form a linear vector space called L_p ***space*** (respectively L_{pe} ***space***) [43].

If the truncated function f_t is defined as

$$f_t(\tau) \overset{\Delta}{=} \begin{cases} f(\tau) & 0 \le \tau \le t \\ 0 & \tau > t \end{cases}$$

for all $t \in [0,\infty)$, then it is clear that for any $p \in [1,\infty)$, $f \in L_{pe}$ implies that $f_t \in L_{pe}$ for any finite t. The L_{pe} space is called the ***extended*** L_p ***space*** and is defined as the set of all functions f such that $f_t \in L_p$.

It can be easily verified that the L_p and L_{pe} norms satisfy the properties of the norm given by Definition 5.1.1. It should be understood, however, that elements of L_p and L_{pe} are *equivalent classes* [43], i.e., if $f,g \in L_p$ and $\| f-g \|_p = 0$, the functions f and g are considered to be the same element of L_p even though $f(t) \neq g(t)$ for some values of t. The following lemmas give some of the properties of L_p and L_{pe} spaces.

***Lemma 5.1.1** (**Holder's Inequality**): If* $p,q \in [1,\infty)$ *and* $\frac{1}{p}+\frac{1}{q}=1$ *then* $f \in L_p, g \in L_q$ *imply that* $fg \in L_1$ *and*

$$\| fg \|_1 \le \| f \|_p \| g \|_q$$

When $p=q=2$, the Holder's inequality becomes the *Schwartz inequality*, i.e.,

$$\| fg \|_1 \le \| f \|_2 \| g \|_2 \qquad \text{(A5.1)}$$

Lemma 5.1.2 (Minkowski Inequality): For $p \in [1, \infty)$, $f, g \in L_p$ imply that $f + g \in L_p$ and

$$\| f + g \|_p \leq \| f \|_p + \| g \|_p \qquad \text{(A5.2)}$$

The proofs of **Lemma 5.1.1** and **5.1.2** can be found in reference books [44, 45].

It should be noted that the above lemmas also hold for the truncated functions f_t, g_t of f, g, respectively, provided $f, g \in L_{pe}$. For example, if f and g are continuous functions, then $f, g \in L_{pe}$, i.e., $f_t, g_t \in L_p$ for any finite $t \in [0, \infty)$ and from (A5.1) one has $\| (fg)_t \|_1 \leq \| f_t \|_2 \| g_t \|_2$, i.e.,

$$\int_0^t | f(\tau) g(\tau) | d\tau \leq \left(\int_0^t | f(\tau) |^2 d\tau \right)^{1/2} + \left(\int_0^t | g(\tau) |^2 d\tau \right)^{1/2} \qquad \text{(A5.3)}$$

which holds for any finite $t \geq 0$.

5.2 Properties of Functions

Definition* 5.2.1*Continuity: *A function* $f : [0, \infty) \to R$ *is* ***continuous*** *on* $[0, \infty)$ *if for any given* $\varepsilon_0 > 0$ *there exists a* $\delta(\varepsilon_0, t_0) > 0$ *such that* $\forall t_0, t \in [0, \infty)$ *for which* $| t - t_0 | < \delta(\varepsilon_0, t_0)$ *one has* $| f(t) - f(t_0) | < \varepsilon_0$.

Definition 5.2.2 Uniform Continuity *A function* $f : [0, \infty) \to R$ *is* ***uniformly continuous*** *on* $[0, \infty)$ *if for any given* $\varepsilon_0 > 0$ *there exists a* $\delta(\varepsilon_0) > 0$ *such that* $\forall t_0, t \in [0, \infty)$ *for which* $| t - t_0 | < \delta(\varepsilon_0)$ *one has* $| f(t) - f(t_0) | < \varepsilon_0$.

In other words, $f(t)$ is uniformly continuous if one can always find an ε which does not depend on the specific point t_0 , and in particular, such that ε does not shrink as $t_0 \to \infty$.

Definition 5.2.3 Piecewise Continuity: *A function $f:[0,\infty)\to R$ is piecewise continuous on $[0,\infty)$ if f is continuous on any finite interval $[t_0,t_1)\subset[0,\infty)$ except for a finite number of points.*

Definition 5.2.4 Absolute Continuity: *A function $f:[a,b]\to R$ is **absolutely continuous** on $[a,b]$ iff, for any $\varepsilon_0>0$, there is a $\delta>0$ such that*

$$\sum_{i=1}^{n}|f(\alpha_i)-f(\beta_i)|<\varepsilon_0$$

for any finite collection of subintervals (α_i,β_i) of $[a,b]$ with $\sum_{i=1}^{n}|\alpha_i-\beta_i|<\delta$.

Definition 5.2.5 Lipschitz: *A function $f:[a,b]\to R$ is **Lipschitz** on $[a,b]$ if*

$$|f(x_1)-f(x_2)|\le k|x_1-x_2|\quad \forall x_1,x_2\in[a,b],$$

*where $k\ge 0$ is a constant referred to as the **Lipschitz constant.***

Please be noted that a ***uniformly continuous function*** is also ***continuous.*** A function f with $\dot{f}\in L_\infty$ is ***uniformly continuous*** on $[0,\infty)$. Therefore, an easy way of checking the ***uniform continuity*** of $f(t)$ is to check the boundedness of $\dot{f}$. If f is ***Lipschitz*** on $[a,b]$, then it is ***absolutely continuous.***

The following facts about functions are important in the analysis of adaptive systems.

Fact 1 $\lim_{t\to\infty}\dot{f}(t)=0$ does not imply that $f(t)$ has a limit as $t\to\infty$.

Fact 2 $\lim_{t\to\infty}f(t)=c$ for some constant $c\in R$ does not imply that $\dot{f}(t)\to 0$ *as* $t\to\infty$.

Some important lemmas that are frequently used in the analysis of adaptive schemes are the following:

Lemma 5.2.1 *The following is true for scalar-valued functions:*

(i) A function $f(t)$ that is bounded from below and is nonincreasing has a limit as $t \to \infty$.

(ii) Consider the nonnegative scalar functions $f(t)$, $g(t)$ defined for all $t \geq 0$. If $f(t) \leq g(t)$ $\forall t \geq 0$ and $g \in L_p$, then $f \in L_p$ for all $p \in [1,\infty)$.

Proof (i) Because $f(t)$ is bounded from below, its infimum f_m exists, i.e.,

$$f_m = \inf_{o \leq t \leq \infty} f(t)$$

which implies that there exists a sequence $\{t_n\} \in R^+$ such that $\lim_{n\to\infty} f(t_n) = f_m$. This, in turn, implies that given any $\varepsilon_0 > 0$ there exists an integer $N > 0$ such that

$$| f(t_n) - f_m | < \varepsilon_0, \forall n \geq N$$

Because f is nonincreasing, there exists an $n_0 \geq N$ such that for any $t \geq t_{n_0}$ and some $n_0 \geq N$ one has

$$f(t) \leq f(t_{n_0})$$

and

$$| f(t) - f_m | \leq | f(t_{n_0}) - f_m | < \varepsilon_0$$

for any $t \geq t_{n_0}$. Because $\varepsilon_0 > 0$ is any given number, it follows that $\lim_{t\to\infty} f(t) = f_m$.

(ii) One has

$$z(t) \overset{\Delta}{=} (\int_0^t f^p(\tau)d\tau)^{\frac{1}{p}} \leq (\int_0^\infty g^p(\tau)d\tau)^{\frac{1}{p}} < \infty, \forall t \geq 0$$

Because $0 \leq z(t) < \infty$ and $z(t)$ is nondecreasing, one can establish, as in (i), that $z(t)$ has a limit, i.e., $\lim_{t\to\infty} z(t) = \bar{z} < \infty$, which implies that $f \in L_p$. For $p = \infty$, the proof is straightforward.

Lemma 5.2.1 (i) does not imply that $f \in L_\infty$. For example, the function $f(t) = 1/t$ with $t \in (0,\infty)$ is bounded from below, i.e., $f(t) \geq 0$ and is nonincreasing, but it becomes unbounded as $t = 0$. If, however, $f(0)$ is finite, then it follows from the nonincreasing property $f(t) \leq f(0) \forall t \geq 0$ that $f \in L_\infty$. A special case of Lemma 5.2.1 which is often used in adaptive analysis is when $f(t) \geq 0$ and $\dot{f}(t) \leq 0$.

Lemma 5.2.2 *Let $f,\ V : [0,\infty) \to R$. Then*

$$\dot{V} \leq -\alpha V + f, \qquad \forall t \geq t_0 \geq 0$$

implies that $V(t) \leq e^{-\alpha(t-t_0)} V(t_0) + \int_{t_0}^{t} e^{-\alpha(t-\tau)} f(\tau) d\tau,\ \forall t \geq t_0 \geq 0$ for any finite constant α.

Proof: Let $w(t) \overset{\Delta}{=} \dot{V} + \alpha V - f$. One has $w(t) \leq 0$ and

$$\dot{V} = -\alpha V + f + w(t)$$

implies that

$$V(t) = e^{-\alpha(t-t_0)} V(t_0) + \int_{t_0}^{t} e^{-\alpha(t-\tau)} f(\tau) d\tau + \int_{t_0}^{t} e^{-\alpha(t-\tau)} w(\tau) d\tau$$

Because $w(t) \leq 0\ \forall t \geq t_0 \geq 0$, one has

$$V(t) \leq e^{-\alpha(t-t_0)} V(t_0) + \int_{t_0}^{t} e^{-\alpha(t-\tau)} f(\tau) d\tau, \quad \forall t \geq t_0 \geq 0$$

Lemma 5.2.3 *If $f, \dot{f} \in L_\infty$ and $f \in L_p$ for some $p \in [1,\infty)$, then $f(t) \to 0$ as $t \to \infty$.*

*The result of Lemma 5.2.3 is a special case of a more general result given by **Barbalat's Lemma** stated below.*

Lemma 5.2.4 (Barbalat's Lemma [46]*)* *If* $\lim_{t\to\infty}\int_0^t f(\tau)d\tau$ *exists and is finite, and* $f(t)$ *is a uniformly continuous function, then* $\lim_{t\to\infty} f(t)=0$.

Proof: Assume that $\lim\limits_{t\to\infty} f(t)=0$ does not hold, i.e., either the limit does not exist or it is not equal to zero. This implies that there exists an $\varepsilon_0>0$ such that for every $T>0$, one can find a sequence of numbers $t_i>T$ such that $|f(t_i)|>\varepsilon_0$ for all i.

Because f is uniformly continuous, there exists a number $\delta(\varepsilon_0)>0$ such that

$$|f(t)-f(t_i)|<\frac{\varepsilon_0}{2}\quad \forall t\in[t_i,t_i+\delta(\varepsilon_0)]$$

Hence, for every $t\in[t_i,t_i+\delta(\varepsilon_0)]$, one has

$$|f(t)|=|f(t)-f(t_i)+f(t_i)|\geq|f(t_i)|-|f(t)-f(t_i)|\geq\varepsilon_0-\frac{\varepsilon_0}{2}=\frac{\varepsilon_0}{2}\quad\forall t\in[t_i,t_i+\delta(\varepsilon_0)]$$

which implies that

$$\left|\int_{t_i}^{t_i+\delta(\varepsilon_0)} f(\tau)d\tau\right|=\int_{t_i}^{t_i+\delta(\varepsilon_0)}|f(\tau)|\,d\tau>\frac{\varepsilon_0\delta(\varepsilon_0)}{2}\qquad\text{(A5.4)}$$

where the first equality holds because $f(t)$ retains the same sign for $t\in[t_i,t_i+\delta(\varepsilon_0)]$. On the other hand, $g(t)\overset{\Delta}{=}\int_0^t f(\tau)d\tau$ has a limit as $t\to\infty$ implies that there exists a $T(\varepsilon_0)>0$ such that for any $t_2>t_1>T(\varepsilon_0)$ one has

$$|g(t_1)-g(t_2)|<\frac{\varepsilon_0\delta(\varepsilon_0)}{2}$$

i.e.,

$$\int_{t_1}^{t_2} f(\tau)d\tau<\frac{\varepsilon_0\delta(\varepsilon_0)}{2}$$

which for $t_2=t_1+\delta(\varepsilon_0), t_1=t_i$ contradicts (A5.4), and, therefore, $\lim_{t\to\infty} f(t)=0$.

The proof of Lemma 5.2.3 follows directly from that of Lemma 5.2.4 by noting that the function $\lim_{t\to\infty}\int_0^t |f(\tau)|\,d\tau = \lim_{t\to\infty}\int_0^t f(\tau)d\tau = f(\infty)$ exists for $f(t)\geq 0$ and $p=1$. $f(t)$ is uniformly continuous for any $p\in[1,\infty)$ because $f(t),\dot{f}(t)\in L_\infty$.

5.3 Lyapunov Stability

Consider systems described by ordinary differential equations of the form

$$\dot{x} = f(t,x), \quad x(t_0) = x_0 \tag{A5.5}$$

where $x\in R^n$, $f: J\times B(r)\mapsto R$, $J=[t_0,\infty)$ and $B(r)=\{x\in R^n \| x| < r\}$. It is assumed that f is of such nature that for every $x_0\in B(r)$ and every $t_0\in R^+$, the differential equation (A5.5) possesses one and only one solution $x(t,t_0,x_0)$.

Definition of Stability:

Definition 5.3.1 *A state x_e of a system is said to be an* **equilibrium state** *of the system described by (A5.5) if*

$$f(t,x_e)\equiv 0 \textit{ for all } t\geq t_0$$

Definition 5.3.2 *The equilibrium state x_e is said to be* **stable (in the sense of Lyapunov)** *if for arbitrary t_0 and $\varepsilon>0$ there exists a $\delta(\varepsilon,t_0)$ such that $|x_0-x_e|<\delta$ implies $|x(t;t_0,x_0)-x_e|<\varepsilon$ for all $t\geq t_0$.*

Definition 5.3.3 *The equilibrium state x_e is said to be* **uniformly stable (u.s.)** *if it is stable and if $\delta(\varepsilon,t_0)$ in Definition 5.3.2 does not depend on t_0.*

Definition 5.3.4 *The equilibrium state x_e is said to be* **asymptotically stable (a.s.)** *if (i) it is stable, and (ii) there exists a $\delta(t_0)$ such that $|x_0-x_e|<\delta(t_0)$ implies* $\lim_{t\to\infty}|x(t;t_0,x_0)-x_e|=0$.

Definition 5.3.5 *The set of all* $x_0 \in R^n$ *such that* $x(t;t_0,x_0) \to x_e$ *as* $t \to \infty$ *for some* $t_0 \geq 0$ *is called the* **region of attraction** *of the equilibrium state* x_e. *If condition (ii) of Definition 5.3.4 is satisfied, then the equilibrium state* x_e *is said to be* **attractive**.

Definition 5.3.6 *The equilibrium state* x_e *is said to be* **uniformly asymptotically stable (u.a.s.)** *if (i) it is uniformly stable, (ii) for every* $\varepsilon > 0$ *and any* $t_0 \in R^+$, *there exist a* $\delta_0 > 0$ *independent of* t_0 *and* ε *and a* $T(\varepsilon) > 0$ *independent of* t_0 *such that* $|x(t;t_0,x_0) - x_e| < \varepsilon$ *for all* $t \geq t_0 + T(\varepsilon)$ *whenever* $|x_0 - x_e| < \delta_0$.

Definition 5.3.7 *The equilibrium state* x_e *is* **exponentially stable (e.s.)** *if there exists an* $\alpha > 0$, *and for ever* $\varepsilon > 0$ *there exists a* $\delta(\varepsilon) > 0$ *such that*

$$|x(t;t_0,x_0) - x_e| < \varepsilon e^{-\alpha(t-t_0)} \textit{ for all } t \geq t_0$$

whenever $|x_0 - x_e| < \delta(\varepsilon)$.

Definition 5.3.8 *The equilibrium state* x_e *is said to be* **unstable** *if it is not stable.*

Equation (A5.4) possesses a unique solution for each $x_0 \in R^n$ and $t_0 \in R^+$. The following definitions for the global characterization of solutions are needed.

Definition 5.3.9 *A solution* $x(t;t_0,x_0)$ *of (A5.4) is* **bounded** *if there exists a* $\beta > 0$ *such that* $|x(t;t_0,x_0)| < \beta$ *for all* $t \geq t_0$, *where* β *may depend on each solution.*

Definition 5.3.10 *The solutions of (A5.4) are* **uniformly bounded (u.b.)** *if for any* $\alpha > 0$ *and* $t_0 \in R^+$, *there exists a* $\beta = \beta(\alpha)$ *independent of* t_0 *such that if* $|x_0| < \alpha$, , *then* $|x(t;t_0,x_0)| < \beta$ *for all* $t \geq t_0$.

Definition 5.3.11 *If* $x(t;t_0,x_0)$ *is a solution of* $\dot{x} = f(t;x)$, *then the trajectory* $x(t;t_0,x_0)$ *is said to be* **stable (u.s., a.s., u.a.s., e.s., unstable)** *if the equilibrium point* $z_e = 0$ *of the differential equation*

$$\dot{z} = f(t;z + x(t;t_0,x_0)) - f(t;x(t;t_0,x_0))$$

is **stable (u.s., a.s., u.a.s., e.s., unstable, respectively).**

Definition 5.3.12 *A continuous function* $\varphi:[0,r] \mapsto R^+$ *(or a continuous function* $\varphi:[0,\infty) \mapsto R^+$*) is said to belong to class K, i.e.,* $\varphi \in K$ *if*

(i) $\varphi(0) = 0$

(ii) φ *is strictly increasing on* $[0,r]$ *(or on* $[0,\infty)$*).*

Definition 5.3.13 *A continuous function* $\varphi:[0,\infty) \mapsto R^+$ *is said to belong to class KR, i.e.,* $\varphi \in KR$ *if*

(i) $\varphi(0) = 0$

(ii) φ *is strictly increasing on* $[0,\infty)$

(iii) $\lim_{r\to\infty} \varphi(r) = \infty.$

Definition 5.3.14 *Two functions* $\varphi_1,\varphi_2 \in K$ *defined on* $[0,r]$ *(or on* $[0,\infty)$*) are said to be of the same order of magnitude if there exist positive constants kl:k2 such that*

$$k_1\varphi_1(r_1) \le \varphi_2(r_1) \le k_2\varphi_1(r_1), \quad \forall r_1 \in [0,r] \quad (or \quad \forall r_1 \in [0,\infty))$$

Definition 5.3.15 *A function* $V(t,x): R^+ \times B(r) \mapsto R$ *with* $V(t,0) = 0 \quad \forall t \in R^+$ *is* **positive definite** *if there exists a continuous function* $\varphi \in K$ *such that* $V(t,x) \ge \varphi(|x|) \quad \forall t \in R^+$ $x \in B(r) = \{x \in R^n \| x |< r\}$ *and som* $r > 0$. $V(t,x)$ *is called* **negative definite** *if* $-V(t,x)$ *is positive definite.*

Definition 5.3.16 *A function $V(t,x): R^+ \times B(r) \mapsto R$ with $V(t,0) = 0 \quad \forall t \in R^+$ is said to be* **positive (negative) semidefinite** *if $V(t,x) \geq 0$ ($V(t,x) \leq 0$) for all $t \in R^+$ and $\forall x \in B(r)$ for some $r > 0$.*

Definition 5.3.17 *A function $V(t,x): R^+ \times B(r) \mapsto R$ with $V(t,0) = 0 \quad \forall t \in R^+$ is said to be* **decrescent** *if there exists $\varphi \in K$ such that $|V(t,x)| \leq \varphi(|x|) \quad \forall t \geq 0$ and $\forall x \in B(r)$ for some $r > 0$.*

Definition 5.3.18 *A function $V(t,x): R^+ \times R^n \mapsto R$ with $V(t,0) = 0 \quad \forall t \in R^+$ is said to be* **radially unbounded** *if there exists $\varphi \in KR$ such that $V(t,x) \geq \varphi(|x|)$ for all $x \in R^n$ and $t \in R^+$.*

It is clear from Definition 5.3.18 that if $V(t,x)$ is radially unbounded, it is also positive definite for all $x \in R^n$ but the converse is not true.

Definition 5.3.19 *Persistence of Excitation (PE) Signal: A piecewise continuous signal vector $\phi(t): R^+ \mapsto R^n$ is PE in R^n with a level of excitation $\alpha_0 > 0$ if there exist constants α_1, $T_0 > 0$ such that*

$$\alpha_1 I \geq \frac{1}{T_0} \int_t^{t+T_0} \phi(\tau)\phi^T(\tau) d\tau \geq \alpha_0 I, \quad \forall t \geq 0$$

Second method of Lyapunov

Without loss of generality, $x_e = 0$ is assumed as an equilibrium point of (A5.4) and $\dot{V}$ is defined to be the time derivative of the function $V(t,x)$ along the solution of (A5.4), i.e.,

$$\dot{V} = \frac{\partial V}{\partial t} + (\nabla V)^T f(t,x) \tag{A5.5}$$

where $\nabla V = [\frac{\partial V}{\partial x_1}, \frac{\partial V}{\partial x_2}, ..., \frac{\partial V}{\partial x_n}]^T$ is the gradient of $V(t,x)$ with respect to x. The second method of Lyapunov is summarized by the following theorem.

Theorem 5.3.1 *Suppose there exists a positive definite function $V(t,x): R^+ \times B(r) \mapsto R$ for some $r > 0$ with continuous first-order partial derivatives with respect to x, t, and $V(t,0) = 0 \quad \forall t \in R^+$. Then the following statements are true:*

(i) If $\dot{V} \leq 0$, then $x_e = 0$ is **stable.**

(ii) If V is decrescent and $\dot{V} \leq 0$, then $x_e = 0$ is **u.s.**

(iii) If V is decrescent and $\dot{V} < 0$, then $x_e = 0$ is **u.a.s.**

(iv) If V is decrescent and there exist $\varphi_1, \varphi_2, \varphi_3 \in K$ of the same order of magnitude such that $\varphi_1(|x|) \leq V(t,x) \leq \varphi_2(|x|)$, $\dot{V}(t,x) \leq -\varphi_3(|x|)$ for all $\forall x \in B(r)$ and $t \in R^+$, then $x_e = 0$ is **e.s.**

5.4 Stability of Linear Systems

Positive Definite Matrices

A square matrix $A \in R^{n \times n}$ is called *symmetric* if $A = A^T$. A symmetric matrix A is called *positive semidefinite* if for every $x \in R^n$, $x^T A x \geq 0$ and *positive definite* if $x^T A x > 0 \quad \forall x \in R^n$ with $|x| \neq 0$. It is called *negative semidefinite* (*negative definite*) if $-A$ is positive semidefinite (positive definite).

A positive semidefinite matrix A *is usually* written as $A \geq 0$, and a positive definite matrix A is written as $A > 0$. If $A - B > 0$ and $A - B \geq 0$ they can be written as $A > B$ and $A \geq B$, respectively.

A symmetric matrix $A \in R^{n\times n}$ is positive definite if and only if any one of the following conditions holds:

(i) $\lambda_i(A) > 0, i = 1,2,...,n$ where $\lambda_i(A)$ denotes the ith eigenvalue of A, which is real because $A = A^T$.

(ii) There exists a nonsingular matrix A_1 such that $A = A_1 A_1^T$.

(iii) Every principal minor of A is positive.

(iv) $x^T A x \geq \alpha |x|^2$ for some $\alpha > 0$ and $\forall x \in R^n$.

The decomposition $A = A_1 A_1^T$ in (ii) is unique when A_1 is also symmetric. In this case, A_1 is positive definite, it has the same eigenvectors as A, and its eigenvalues are equal to the square roots of the corresponding eigenvalues of A. This unique decomposition A_1 of A can be specified by denoting as $A^{\frac{1}{2}}$, i.e., $A = A^{\frac{1}{2}} A^{\frac{T}{2}}$ where $A^{\frac{1}{2}}$ is a positive definite matrix and $A^{\frac{T}{2}}$ denotes the transpose of $A^{\frac{1}{2}}$.

Lemma 5.4.1 (Kalman-Yakubovich) Consider a controllable linear time-invariant system

$$\dot{X} = AX + BU$$
$$Y = C^T X$$

The transfer function is

$$H(s) = C^T [sI - A]^{-1} B$$

and is strictly positive real if, and only if, there exists a positive definite matrices P and Q such that

$$PA + A^T P = -Q$$

$$PB = C$$

Appendix 6

Optimization Techniques

In this appendix, two simple optimization techniques, the ***gradient method*** and the ***gradient projection method*** for constrained minimization problems will be introduced.

6.1 Notation and Mathematical Background

A real-valued function $f : R^n \mapsto R$ is said to be continuously differentiable and is written as $f \in C^1$ if the partial derivatives $\frac{\partial f(x)}{\partial x_1},...,\frac{\partial f(x)}{\partial x_n}$ exist for each $x \in R^n$ and are continuous functions of x. More generally, if all partial derivatives of order m of the function $f : R^n \mapsto R$ exist and are continuous functions of x, the function is written as $f \in C^m$.

If $f \in C^1$, the **gradient** of f at a point $x \in R^n$ is defined to be the column vector

$$\nabla f(x) \overset{\Delta}{=} \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} & \cdots & \frac{\partial f(x)}{\partial x_n} \end{bmatrix}^T$$

If $f \in C^2$, the **Hessian** of f at x is defined to be the symmetric $n \times n$ matrix having $\partial^2 f(x) / \partial x_i \partial x_j$ as the ij th element, i.e.,

$$\nabla^2 f(x) \overset{\Delta}{=} \left[\frac{\partial^2 f(x)}{\partial x_i \partial x_j} \right]_{n \times n}$$

A subset S of R^n is said to be **convex** if for every $x, y \in S$ and $\alpha \in [0,1)$, there exists

$$\alpha x + (1-\alpha) y \in S.$$

A function $f : S \mapsto R$ is said to be **convex** over the convex set S if for every $x, y \in S$ and $\alpha \in [0,1)$ there exists

$$f[\alpha x+(1-\alpha)y]\leq \alpha f(x)+(1-\alpha)f(y).$$

Let $f\in C^1$ over an open convex set S, then f is convex over S iff

$$f(y)\geq f(x)+(\nabla f(x))^T(y-x)\forall x,y\in S \tag{6.1a}$$

If $f\in C^2$ over S and $\nabla^2 f(x)\geq 0\quad \forall x\in S$, then f is convex over S.

Now consider the following unconstrained minimization problem

$$\min_{\theta\in R^n}(J(\theta)) \tag{6.2a}$$

where $J: R^n \mapsto R$ is a given function. The vector θ^* is said to be a global minimum for (6.2a) if

$$J(\theta^*)\leq J(\theta)\quad \forall\theta\in R^n$$

A necessary and sufficient condition satisfied by the global minimum θ^* is given by the following lemma.

Lemma 6.1 Assume that $f\in C^1$ and is convex over R^n. Then θ^* is a global minimum for (6.2a) iff

$$\nabla J(\theta^*)=0$$

The proof of Lemma 6.1 can be found in [47, 48]

A vector $\bar{\theta}$ is called a regular point of the surface $S_\theta=\{\theta\in R^n \mid g(\theta)=0\}$ if $\nabla g(\bar{\theta})\neq 0$. At a regular point $\bar{\theta}$, the set

$$M(\bar{\theta})=\{\theta\in R^n \mid \theta^T\nabla g(\bar{\theta})=0\}$$

is called the ***tangent plane*** of g at $\bar{\theta}$.

6.2 The Method of Steepest Descent (Gradient Method)

The gradient method is one of the oldest and most widely known methods for solving the unconstrained minimization problem (6.2a). The method proceeds from an initial approximation θ_0 for the minimum θ^* to successive points $\theta_1, \theta_2, \ldots$, in R^n in an iterative manner until some stopping condition is satisfied. Given the current point θ_k, the point θ_{k+1} is obtained by a linear search in the direction d_k where

$$d_k = -\nabla J(\theta_k)$$

It can be shown [48] that d_k is the direction from θ_k in which the initial rate of decrease of $J(\theta)$ is the greatest. Therefore, the sequence $\{\theta_k\}$ is defined by

$$\theta_{k+1} = \theta_k - \lambda_k \nabla J(\theta_k) \tag{6.3a}$$

where θ_0 is given and λ_k, known as the *step size* or *step length*, is determined by the linear search method, so that θ_{k+1} minimizes $J(\theta)$ in the direction d_k from θ_k. A simpler expression for θ_{k+1} can be obtained by setting $\lambda_k = \lambda, \forall k$, i.e.,

$$\theta_{k+1} = \theta_k - \lambda \nabla J(\theta_k) \tag{6.4a}$$

In this case, the linear search for λ_k is not required, though the choice of the step length λ is a compromise between accuracy and efficiency.

Considering infinitesimally small step lengths, (6.4a) can be converted to the continuous time differential equation

$$\dot{\theta} = -\nabla J(\theta(t)), \quad \theta(t) = \theta_0 \tag{6.5a}$$

whose solution $\theta(t)$ is the descent path in the time domain starting from $t = t_0$.

The direction of steepest descent $d = -\nabla J$ can be scaled by a constant positive definite matrix $\Gamma = \Gamma^T$ as follows: let $\Gamma = \Gamma_1 \Gamma_1^T$ where Γ_1 is an $n \times n$ nonsingular matrix and consider the vector $\bar{\theta} \in R^n$ given by

$$\Gamma_1 \bar{\theta} = \theta$$

Then the minimization problem (6.2a) is equivalent to

$$\min_{\bar{\theta} \in R^n} (\bar{J}(\bar{\theta})) \overset{\Delta}{=} J(\Gamma_1 \bar{\theta}) \tag{6.6a}$$

If $\bar{\theta}^*$ is a minimum of $\bar{J}$, the vector $\theta^* = \Gamma_1 \bar{\theta}^*$ is a minimum of J. The steepest descent for (6.6a) is given by

$$\bar{\theta}_{k+1} = \bar{\theta}_k - \lambda \nabla \bar{J}(\bar{\theta}_k) \tag{6.7a}$$

Because $\nabla \bar{J}(\bar{\theta}) = \partial J(\Gamma_1 \bar{\theta}) / \partial \bar{\theta} = \Gamma_1^T \nabla J(\theta)$ and $\Gamma_1 \bar{\theta} = \theta$ it follows from (6.7a) that

$$\theta_{k+1} = \theta_k - \lambda \Gamma_1 \Gamma_1^T \nabla J(\theta_k) \tag{6.8a}$$

Setting $\Gamma = \Gamma_1 \Gamma_1^T$, one obtains the scaled version for the steepest descent algorithm

$$\theta_{k+1} = \theta_k - \lambda \Gamma \nabla J(\theta_k) \tag{6.9a}$$

The continuous-time version of (6.9a) is now given by

$$\dot{\theta} = -\Gamma \nabla J(\theta) \tag{6.10a}$$

The convergence properties of (6.3a), (6.4a), (6.8a) for different step lengths are given in any standard book on optimization such as [47, 48]. The algorithms (6.5a), (6.10a) for various cost functions $J(\theta)$ are used in Chapters 3 and Chapter 5 where the design and analysis of adaptive laws is considered.

6.3 Gradient Projection Method

In sections 6.2, the search for the minimum of the function $J(\theta)$ given in (6.2a) was carried out for all $\theta \in R^n$. In some cases, $\theta$is constrained to belong to a certain convex set

$$S \overset{\Delta}{=} \{\theta \in R^n \mid g(\theta) \leq 0\} \tag{6.11a}$$

in $R^n$where $g(\cdot)$ is a scalar-valued function if there is only one constraint and a vector-valued function if there are more than one constraints. In this case, the search for the minimum is restricted to the convex set defined by (6.11a) instead of R^n.

First a simple case which has an equality constraint is considered, namely,

$$\min_{subject\ to\ g(\theta)=0} J(\theta) \tag{6.12a}$$

where$g(\theta)$ is a scalar-valued function. One of the most common techniques for handling constraints is to use a descent method in which the direction of descent is chosen to reduce the function $J(\theta)$ by remaining within the constrained region. Such method is usually referred to as the ***gradient projection method.***

The parameter vector θ is selected starting from a point θ_0 which satisfies the constraint $g(\theta_0) = 0$. To obtain an improved vectorθ_1, the negative gradient of J is projected atθ_0 i.e.,$-\nabla J(\theta_0)$ is projected onto the tangent plane$M(\theta_0)=\{\theta \in R^n \mid \nabla g^T(\theta_0)\theta=0\}$ to obtain the direction vector$proj(\theta_0)$. Then θ_1 is taken as $\theta_0 + \lambda_0 proj(\theta_0)$ where λ_0 is chosen to minimize$J(\theta_1)$. The general form of this iteration is given by

$$\theta_{k+1} = \theta_k + \lambda_k proj(\theta_k) \tag{6.13a}$$

where λ_k is chosen to minimize $J(\theta_k)$ and $proj(\theta_k)$ is the new direction vector after projecting $-\nabla J(\theta_k)$ onto $M(\theta_k)$. The explicit expression for $proj(\theta_k)$ can be obtained as follows: The vector $-\nabla J(\theta_k)$ can be expressed as a linear combination of the vector $proj(\theta_k)$ and the normal vector $Norm(\theta_k) = \nabla g(\theta_k)$ to the tangent plane $M(\theta_k)$ at θ_k, i.e.,

$$-\nabla J(\theta_k) = \alpha \nabla g(\theta_k) + proj(\theta_k) \tag{6.14a}$$

for some constant α. Because $proj(\theta_k)$ lies on the tangent plane $M(\theta_k)$, one also has $\nabla g^T(\theta_k) proj(\theta_k) = 0$ which together with (6.14a) implies that

$$-\nabla g^T \nabla J = \alpha \nabla g^T \nabla g$$

i.e.,

$$\alpha = -(\nabla g^T \nabla g)^{-1} \nabla g^T \nabla J$$

Hence, from (6.14a), one can obtain

$$proj(\theta_k) = -[I - \nabla g(\nabla g^T \nabla g)^{-1} \nabla g^T] \nabla J \tag{6.15a}$$

Here $proj(\theta_k)$ is referred as the projected direction onto the tangent plant $M(\theta_k)$. The gradient projection method is illustrated in Fig.13A.

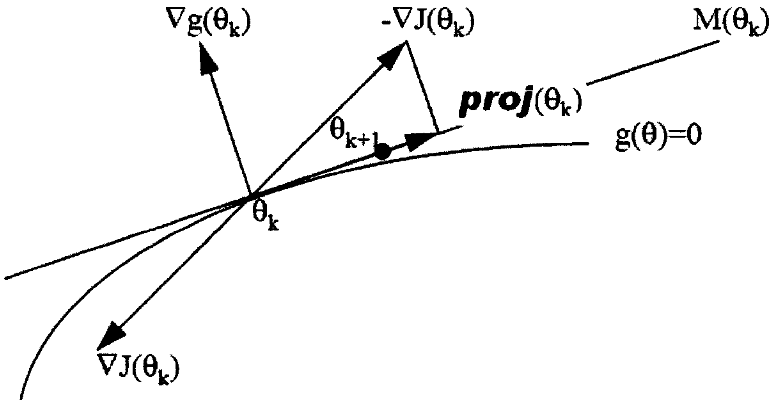


Fig.13A Gradient projection method

It is clear from Fig.6.1a that when $g(\theta)$ is not a linear function of θ, the new vector θ_{k+1} given by (6.13a) may not satisfy the constraint, so it must be modified. There are several successive approximation techniques that can be employed to move θ_{k+1} from $M(\theta_k)$ to the constraint surface $g(\theta)=0$ [47, 48]. One special case, which is often encountered in adaptive control applications, is when θ is constrained to stay inside a ball with a given center and radius, i.e., $g(\theta)=(\theta-\theta_0)^T(\theta-\theta_0)-R_b{}^2$ where θ_0 is a fixed constant vector and $R_b>0$ is a scalar. In this case, the discrete projection algorithm which guarantees that $\theta_k \in S \quad \forall k$ is

$$\overline{\theta}_{k+1}=\theta_k+\lambda_k\nabla J$$

$$\theta_{k+1}=\begin{cases}\overline{\theta}_{k+1} & if \quad |\overline{\theta}_{k+1}-\theta_0|\leq R_b\\ \theta_0+\dfrac{\overline{\theta}_{k+1}-\theta_0}{|\overline{\theta}_{k+1}-\theta_0|}R_b & if \quad |\overline{\theta}_{k+1}-\theta_0|>R_b\end{cases} \tag{6.16a}$$

Letting the step length λ_k become infinitesimally small, one obtains the continuous time version of (6.13a), i.e.,

$$\dot{\theta}=proj(\theta)=-[I-\nabla g(\nabla g^T\nabla g)^{-1}\nabla g^T]\nabla J \tag{6.17a}$$

Because of the sufficiently small step length, the trajectory $\theta(t)$, if it exists, will satisfy $g(\theta(t))=0 \quad \forall t\geq 0$, provided $\theta(0)=\theta_0$ satisfies $g(\theta_0)=0$.

The scaled version of the gradient projection method can be obtained by using the change of coordinates $\Gamma_1\overline{\theta}=\theta$ where Γ_1 is a nonsingular matrix that satisfies $\Gamma=\Gamma_1\Gamma_1{}^T$ and Γ is the scaling positive definite constant matrix. Following a similar approach as in section 6.2, the scaled version of (6.17a) is given by

$$\dot{\theta}=proj(\Gamma_1\overline{\theta})=-[I-\nabla g(\nabla g^T\nabla g)^{-1}\nabla g^T]\nabla J$$

The minimization problem (6.12a) can now be extended to

$$\min_{subject\ to\ g(\theta)\ \leq\ 0} J(\theta) \tag{6.18a}$$

where $S = \{\theta \in R^n \mid g(\theta) \leq 0\}$ is a convex subset of R^n.

The solution to (6.18a) follows directly from that of the unconstrained problem and (6.12a). Starting from an initial point $\theta_0 \in S$. If the current point is in the interior of S, defined as $S_o = \{\theta \in R^n \mid g(\theta) < 0\}$, then the unconstrained algorithm is used. If the current point is on the boundary of S, defined as $\delta(S) \overset{\Delta}{=} \{\theta \in R^n \mid g(\theta) = 0\}$ and the direction of search given by the unconstrained algorithm is pointing away from S, then we use the gradient projection algorithm. If the direction of search is pointing inside S then we keep the unconstrained algorithm. In view of the above, the solution to the constrained optimization problem (6.18a) is given by

$$\dot{\theta} = \begin{cases} -\nabla J(\theta) & if\ \ \theta \in S_o\ \ or\ \ \theta \in \delta(S)\ \ and\ \ -\nabla J^T \nabla g \leq 0 \\ -\nabla J + \dfrac{\nabla g \nabla g^T}{\nabla g^T \nabla g} \nabla J & otherwise \end{cases} \tag{6.19a}$$

where $\theta(0) \in S$ or with the scaling matrix

$$\dot{\theta} = \begin{cases} -\Gamma \nabla J(\theta) & if\ \ \theta \in S_o\ \ or\ \ \theta \in \delta(S)\ \ and\ \ -(\Gamma \nabla J)^T \nabla g \leq 0 \\ -\Gamma \nabla J + \Gamma \dfrac{\nabla g \nabla g^T}{\nabla g^T \Gamma \nabla g} \Gamma \nabla J & otherwise \end{cases} \tag{6.20a}$$

Appendix 7

Simulation model for newly defined hysteresis model

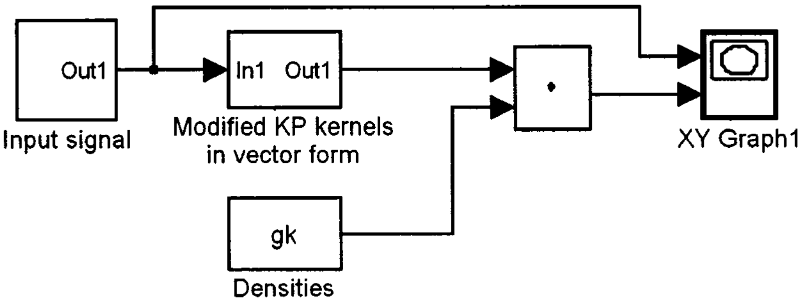


Fig.14A Simulation model of hysteresis in the newly defined hysteresis model

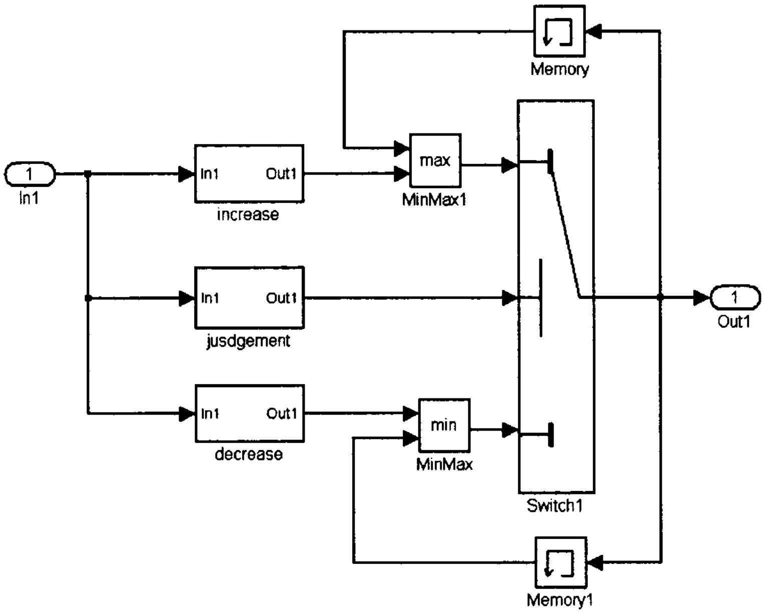


Fig.15A Subsystem of "an elementary operator of the newly defined hysteresis model in vector form"

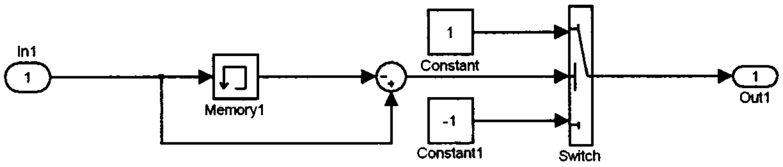


Fig.16A Subsystem of "judgment" in Fig.15A

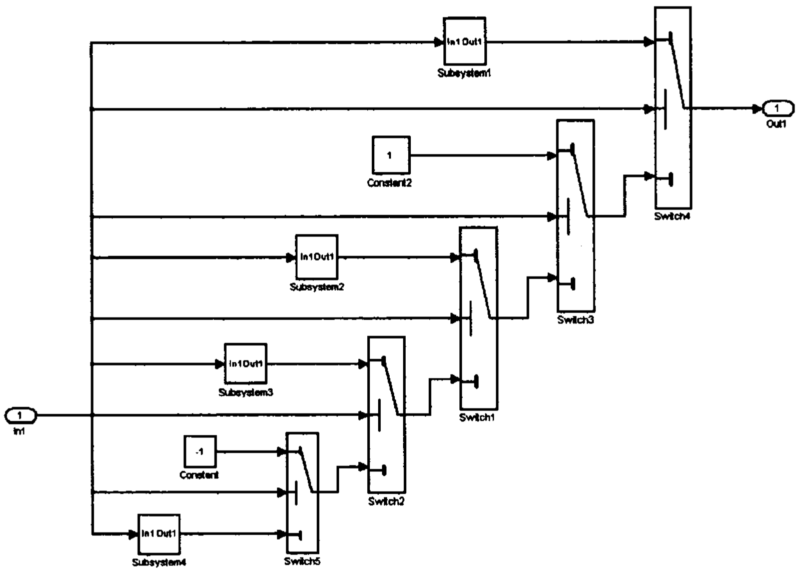


Fig.17A Subsystem of "ascending boundary" in Fig.15A

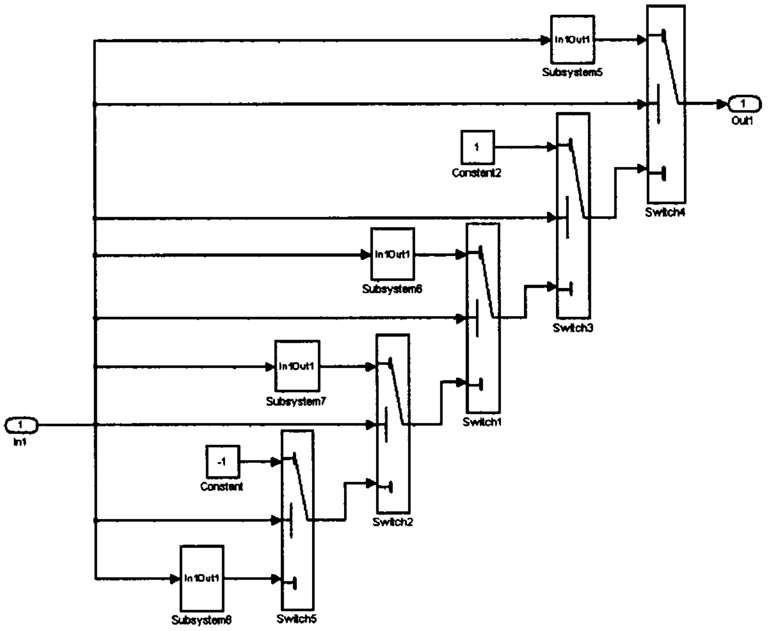


Fig.18A Subsystem of "descending boundary" in Fig.15A

REFERENCES

[1] Song G, V Chaudhry and C Batur 2003 Precision tracking control of shape memory alloy actuators using neural networks and a sliding-mode based robust controller.

[2] Epps J and Chopra I 2001 In-flight tracking of helicopter rotor blades using shape memory alloy actuators Smart Mater Struct. 10 104–11.

[3] Wang H, Li Q and Sun J 2002 Active vibration control of rotor with shape memory alloy (SMA) spring J. Mech. Strength 24 29–31.

[4] Elahinia M H and Ashrafiuon H 2002 Nonlinear control of a shape memory alloy actuated manipulator J. Vib. Acoust., Trans. ASME 124 566–75.

[5] Duerig T W, Melton K N, Stockel D andWayman C M 1990 *Engineering Aspects of Shape Memory Alloys* (London: Butterworth Heinemann).

[6] 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.

[7] G. Tao and P. V. Kokotovi´c. Adaptive control of plants with unknown hysteresis. IEEE Transactions on Automatic Control, 40(2):200–212, 1995.

[8] R. C. Smith. Inverse compensation for hysteresis in magnetostrictive transducers. Technical Report CRSC-TR98-36, CRSC, North Carolina State University, 1998.

[9] W. S. Galinaitis and R. C. Rogers. Control of a hysteretic actuator using inverse hysteresis compensation. In V.V. Varadan, editor, Mathematics and Control in Smart Structures, volume 3323 of SPIE, pages 267–277, 1998.

[10] 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.

[11] C. Natale, F. Velardi, and C. Visone. Modelling and compensation of hysteresis for magnetostrictive actuators. In Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pages 744–749, 2001.

[12] M. A. Krasnosel'skii and A. V. Pokrovskii. Systems with Hysteresis. Springer-Verlag, 1989.

[13] I. D. Mayergoyz. Mathematical Models of Hysteresis. Springer Verlag, 1991.

[14] A. Visintin. Differential Models of Hysteresis. Springer, 1994.

[15] M. Brokate and J. Sprekels. Hysteresis and Phase Transitions. Springer Verlag, New York, 1996.

[16] E. Della Torre. Magnetic Hysteresis. IEEE Press, Piscataway, NJ, 1999.

[17] Cruz-Hern'andez J M and Hayward V 1998 An approach to reduction of hysteresis in smart materials *IEEE Int. Conf. Robotics Automation* 2 15 10–15.

[18] Webb G V and Lagoudas D C 1998 Hysteresis modeling of SMA actuators for control applications *Smart Mater. Struct.* 94 32–48.

[19] Song G and Quinn D 2000 Robust tracking control of a shape memory alloy wire actuator *6th ASME Biennial Symp. On Active Control of Vibration and Noise (Orlando, FL, 2000).*

[20] Hughes D andWen J 1994 Preisach modeling and compensation for smart material hysteresis *SPIE Active Mater. Smart Struct.* 2427 50–64.

[21] Ge P and JouanehM 1996 Tracking control of a piezoceramic actuator *IEEE Trans. Control Syst. Technol.* 4 209–16

[22] M. Goldfarb and N. Celanovic, "Behavioral implications of piezoelectric stack actuators for control of micromanipulation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1996, pp. 226–231.

[23] J. M. Cruz-Hernández and V. Hayward, "On the linear compensation of hysteresis," in *36th IEEE Conf. Decision Contr.*, vol. 1, 1997, pp.1956–1957.

[24] H. T. Banks and A. J. Kurdila, "Hysteretic control influence operators representing smart material actuators: Identification and approximation," in *Proc. IEEE Conf. Decision Contr.*, vol. 4, 1996, pp. 3711–3716.

[25] J.-J. E. Slotine and W. Li, Applied Nonlinear Control, Englewood Cliff, NJ: Prentice-Hall,1991.

[26] T. E. Pare and J. P. How, "Robust stability and performance analysis of systems with hysteresis nonlinearities," in Proc. Amer. Control Conf., 1998, pp. 1904–1908.

[27] A. Isidori, Nonlinear Control Systems: An Introduction, 2nd ed. Berlin, Germany: Springer-Verlag, 1989.

[28] P. A. Ioannou and J. Sun, Robust adaptive control, Prentice hall PTR, 1996.

[29] G. Tao and F. L. Lewis (Eds), Adaptive Control of Nonsmooth Dynamic Systems, Springer, 2001.

[30] N. J. Ahmad and F. Khorrami, "Adaptive control of systems with backlash hysteresis at the input," in Proc. Amer. Control Conf., 1999, pp. 3018–3023.

[31] C. Hatipoglu and Ü. Özgüner, "Robust control of systems involving nonsmooth nonlinearities using modified sliding manifolds," in Proc. Amer. Control Conf., 1998, pp. 2133–2137.

[32] J. M. Nealis and R. C. Smith, "Nonlinear Adaptive Parameter Estimation Algorithms for Hysteresis Models of Magnetostrictive Actuators," Proceedings of the SPIE, Smart Structures and Materials 2002, Vol. 4693, pp. 25-36, 2003.

[33] C. Y. Su, Y. Stepanenko, J. Svoboda, and T. P. Leung, "Robust adaptive control of a class of nonlinear systems with unknown backlash-like hysteresis," IEEE Trans. Automat. Contr., vol. 45, pp. 2427–2432, Dec. 2000.

[34] X. Tan, R. Venkataraman and P.S. Krishnaprasad, "Control of hysteresis: Theory and experimental results," Smart Structures and Materials 2001, Modelling, Signal Processing and Control in Smart Structures, SPIE Vol. 4326, pp. 101-112, 2001.

[35] R. B. Gorbet, "Control of hysteresis systems with Preisach representations," Ph. D Thesis, Waterloo, Ontatio, Canada, 1997.

[36] Song G, V Chaudhry and C Batur 2003 A Neural Network Inverse Model for a Shape Memory Alloy Wire Actuator.

[37] A. Visintin (ed.): Models of hysteresis. Proceedings of a meeting held in Trento in 1991. Longman, Harlow 1993

[38] Stoner, E.C., E.P.Wohlfarth. 1948. "A mechanism of magnetic hysteresis in heterogeneous alloys," Philos. Trans. Royal Soc. London. 240:599-642.

[39] Preisach, F. 1935. "Uber die magnetische nachwirkung," Z. Phys., 94:277-302.

[40] H. Bank, A. Kurdila, and G.Webb, "Identification of hysteretic control influence operators representing smart actuators: Formulation," Tech.Rep.CRSC-TR96-14,Center for Research in Scienctific Computation, North Carolina State University, Raleigh, N.C., 1996.

[41] H. Bank, A. Kurdila, and G.Webb, "Identification of hysteretic control influence operators representing smart actuators: Convergent approxiamtions," Tech.Rep.CRSC-TR96-14,Center for Research in Scienctific Computation, North Carolina State University, Raleigh, N.C., 1997.

[42] J.A. Nelder and R.Mead. A simplex method for function minimization. Computer Journal, 7:308-313,1964.

[43] Desoer, C.A. and M. Vidyasagar, *Feedback Systems: Input-Output Properties*, Academic Press Inc., New York, 1975.

[44] Royden, H.L., Real Analysis, The Macmillan Company, New York, 1963.

[45] Rudin, W., Real and Complex Analysis, 2nd Edition, McGraw-Hill, New York, 1974.

[46] Popov, V. M., Hyperstability of Control Systems, Springer-Verlag, New York, 1973.

[47] Luenberger, D.G. Optimization by Vector Space Methods, John Wiley & Sons, Inc., New York, 1969.

[48] Rao, S.S., Optimization: Theory and Application, Wiley Eastern LtD, 1984.