NTNU – Trondheim
Norwegian University of
Science and Technology

Identification, Inversion and Implementaion of the Preisach Hysteresis Model in Nanopositioning

Jon Åge Stakvik

Master of Science in Cybernetics and Robotics
Submission date: June 2014
Supervisor: Jan Tommy Gravdahl, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

NTNU
Norges teknisk-naturvitenskapelige universitet

Fakultet for informasjonsteknologi, matematikk og elektroteknikk
Institutt for teknisk kybernetikk

Master of Science Thesis Assignment

Name of the candidate: Jon Åge Stakvik
Subject: Engineering Cybernetics
Title: Hysteresis modeling and identification for nanopositioning

Background

Nanopositioning is positioning of some device with nanometer accuracy. This kind of performance is impossible without feed-forward and feedback control, and as such, control systems design is a key enabling technology for many nano-science applications and in particular nanopositioning. The atomic force microscope (AFM) is one of the foremost tools for imaging, measuring and manipulation at the nanometer scale. In this work, experiments are to be done at the nanopositioning lab at the Department of Engineering Cybernetics and using a Park Systems XE-70 AFM and other positioning stages. In this work, the candidate will further investigate hysteresis models, inversion and identification methods for such models. Hysteresis appear as an unwanted phenomenon in the piezoelectric actuators of the positioning stage of the AFM. This master's thesis is a continuation of Stakvik's 5th year project work[1].

Assignment:

1. Perform a literature review on experiment design; with focus on hysteresis model identification. Use [2] as a starting point.
2. By using experiment design, devise experiments to identify hysteresis models (primarily Preisach-model) in some optimal way, based on design of input signals.
3. Validate (compare model to experiment) identified hysteresis models under various conditions. Try to find the cases where the most mismatch between model and experiment occur, and the best match occur (the extreme cases).
4. Simulate, and then implement an inverse compensator and perform experiments on the nanopositioning lab. Compare the performance when using inverse models based on different identification conditions (i.e. worst and best case).
5. Study the effect of signal loop errors, such as negative phase, on hysteresis identification for the Preisach-model. Compare the model-measurement relationship for various input frequencies.

To be handed in by: 7/7-2014
Co-supervisors: PhD-student Michael Ragazzon, post.doc. Arnfinn Aas Eielsen

Trondheim, 03.02.2014

Jan Tommy Gravdahl
Professor, supervisor

[1] J.Å. Stakvik, Parameter identification for hysteresis models in nanopositioning, 5th year project, ITK, 2013
[2] G.C. Goodwin, Dynamic System Identification: Experiment Design and Data Analysis, Academic Press, 1977

Preface

It always seems impossible until it's done. —Nelson Mandela

This Master's thesis was carried out at the Department of Engineering Cybernetics at the Norwegian University and Technology during the spring semester of 2014.

I would first like to thank my main supervisor, Professor Jan Tommy Gravdahl for regular meetings, support and interesting discussions. The interest you have shown for my work, inspires me to do work even harder. I would also like to thank my co-supervisors Post Doc. Arnfinn Aas Eielsen and PhD. student Michael Remo Palmén Ragazzon for help, directions and guidance throughout the semester. You have all provided a positive working environment, and I hope I will be able to work with you again in the future.

A big thanks to all my friends, both at NTNU and elsewhere. I hope that we will continue to meet in the future, even though the geographical distances are greater.

Thank to my parents, Åse Karin and Jonny, for being who you are. You always try to understand my studies, even when I am not quite sure myself. I would never be the person I am today without your support and love.

Last, but not least, I would like to thank my loving wife Emily. Your love, support and encouragement during our five years at NTNU has been what made this possible. We made it together.

Jon Åge Stakvik

Trondheim, June 17, 2014

Abstract

Atomic force microscopes offer the possibility for imaging a sample topography with accuracy in the nanometer scale. This kind of performance is impossible without actuators enabling a certain accuracy. A frequently used actuator in such nanopositioning applications is the piezoelectric actuator which, although precise, exhibit the unwanted phenomenon of hysteresis. Hysteresis is recognized as the main nonlinearity in the piezoelectric actuator, and introduces severe control issues for accurate positioning.

To compensate for hysteresis an accurate model of the nonlinearity has to be identified. A discrete and a piecewise-continuous model, based on the Preisach operator, is identified with a constraint least squares method. An algorithm for computing the output of the continuous model is proposed and outlined together with a solution for the discrete model.

The general approach for control of hysteresis is to create a right inverse of the identified model. Since the Preisach operator does not have an analytical inverse, this is done numerically by a closest match algorithm. These algorithms are described based on previous literature.

In order to achieve a well designed model for the Preisach operator, the input function has to be thoughtfully designed. It is pointed out that to achieve a persistently excited input signal, the number of reversals has to be sufficient compared to the discretization level of the implemented Preisach operator. Additionally, effects from systematic errors in the signal loop, such as negative phase due to low pass filtering, time delay, creep and actuator dynamics, has to be minimized. It was shown that negative phase in the signal loop can invalidate the identified models.

Both the discrete and continuous model was shown to describe the hysteresis nonlinearity. To reduce the step-size of each discrete value, a higher discretization level can be applied to the discrete model. The inverse models were simulated to verify the inversion algorithms, followed by results from testing the inverse signal. The inverted signal based on the discrete method showed need for interpolation between the discrete levels, while the continuous inverse exhibit higher frequency components that causing vibrations due to resonance dynamics.

An input signal consisting of the same amount of reversals as the discretization level was shown to give a valid identified model. Furthermore, increasing the number of reversals was shown to continue improving the accuracy. The effect of negative phase, from low pass

filtering and time delay, was tested, and changes in the distribution of the Preisach weights were shown. Such effects can cause significant errors in identification of hysteresis models, especially if the reference frequency is subject to changes.

Sammendrag

Atomkraftmikroskop gir muligheten til å ta bilde av topografien til et materiale med nanometers nøyaktighet. Denne typen ytelse er umulig uten posisjoneringverktøy som muliggjør en slik presisjon. En vanlig aktuator brukt i nanoposisjonering er den piezoelektriske aktuatoren, som innehar en ulinearitet kalt hysterese. Hysterese er ansett som den største ulineariteten i piezoelektriske aktuatorer, og introduserer alvorlige kontrollproblemer for nøyaktig posisjonering.

For å kompensere for hysterese er det nødvendig å identifisere en nøyaktig modell. I denne oppgaven vil både en diskret og en kontinuerlig metode, basert på Preisach operatoren, bli identifisert med en begrenset lineær minste kvadraters metode. En algoritme for å beregne utgangen til den kontinuerlige metoden er foreslått sammen med en løsning for den diskre modellen.

Den generelle metoden for å kontrollere hysterese er å regne ut den inverterte hysteresemodellen. Siden Preisach operatoren ikke har noen analytisk invers, gjøres dette numerisk med algoritmer basert på å finne den nærmest mulige inngangen. Disse algoritmene er basert på tidligere litteratur.

For å oppnå en godt designet modell for Preisach operatoren må inngangssignalet være gjennomtenkt. Det er poengtert at for å oppnå et persistent eksitert inngangssignal må inngangen reverseres nok ganger, sammenlignet med diskretiseringsnivået til den implementerte Preisach operatoren. I tillegg må effekten fra systematiske feil i signalsløyfen, for eksempel negativ fase på grunn av lavpassfiltrering, tidsforsinkelse, kryp og aktuatordynamikk, minimeres. Det ble vist at negative fase kan ugyldiggjøre de identifiserte modellene.

Både den diskret og kontinuerlige modellen klarer å beskrive hysterese. For å minke steglengden på de diskre stegne, kan en øke diskretiseringsnivået. De inverse modellene ble simulert for å verifisere inverteringsalgoritmene, fulgt av resultat fra testing av de inverse signalene. Det inverterte signalet basert på den diskre metoden trenger interpolering mellom de diskre verdiene, mens den inverse til den kontinuerlige modellen inneholder relativt høye frekvenskomponenter, som skaper vibrasjoner på grunn av resonansen i dynamikken.

Et inngangssignal bestående av like mange reverseringer som antall diskre nivåer viste seg å gi en god modell. I tillegg, å øke antallet reverseringer ga enda bedre nøyaktighet. Effekten av negativ fase på grunn av lavpassfiltrering og tidsforsinkelse ble testet, og forandringer i fordelingen av vekten til Preisach elementene ble vist. Slike effekter kan forårsake en tydelig

feil i identifiseringen av hysteresemodellen, spesielt hvis frekvensen til referansen endres.

Contents

List of Figures

List of Tables

List of Symbols and Abbreviations

Symbol	Explanation	Unit(s)
f_D	Decreasing function part of Duhem model	-
f_I	Increasing function part of Duhem model	-
n_h	Discretization level of Preisach model	-
n_q	Total number of Preisach elements	-
r	Switch value for Prandtl-Ishlinskii model	-
$\mathcal{R}_{\alpha,\beta}$	Relay output	-

Greek letter	Explanation	Unit(s)
α	Upper limit of a Preisach element	V
α_m	Maximum limit of Preisach model	V
β	Lower limit of a Preisach element	V
β_m	Minimum limit of Preisach model	V
Γ	Function for Preisach output	-
η	Output value of Preisach-type operators	V or μm
η_d	Desired output value	V or μm
μ	Preisach weight function	V or μm
ξ	State of a single Preisach element	-
τ	Time delay	s
ψ	Memory curve of Preisach operator	-

Abbreviation	Explanation
AFM	Atomic Force Microscopy
KP	Krasnosel'skii-Pokrovskii Operator
PE	Percistence Excited
PI	Prandtl-Ishlinskii Operator
PRBN	Pseudo Random Binary Noise

PSI	Positive Strictly Increasing
SPM	Scanning Probe Microscopy
STM	Scanning Tunneling Microscopy

Chapter 1

Introduction

1.1 Motivation

Nanopositioning is positioning of some device with nanometer accuracy. Consequently, it is a vital tool in the field of nanotechnology, which include areas like biology, chemistry, material science and physics. Applications in nanotechnology require imaging in the atomic scale, such as scanning prope microscopy (SPM). This class of microscopes includes, among others, atomic force microscopy (AFM) and scanning tunneling microscopy (STM), both these methods can achieve resolutions in the atomic range.

To achieve an accuracy in the nanometer range, ultra-precise instruments and actuators are needed. Materials obtaining this accuracy are often called smart materials due to their ability to change dimensions based on some external effect. The piezoelectric actuator is such a material, changing dimensions when a voltage is applied to it. Controlling such actuators is a challenging task, due to nonlinearities exhibited from the materials. In a piezoelectric actuator, these nonlinearities are known as creep, hysteresis and, when experiencing high frequency input, vibrations. Hysteresis is recognized as the main nonlinearity in piezoelectric actuators.

By compensating for hysteresis, the performance of piezoelectric actuators in nanopositioing can be drastically improved. For this reason, several models have been used to describe this nonlinearity, perhaps most commonly the Preisach model, and variations of this. A general approach for hysteresis compensation is to first identify the model, then create a right inverse based on the model, removing the effect of hysteresis. Doing this, several challenges arise related to issues during implementation, high frequencies in the inverted signal

and effects due to instrumentation, such as low pass filters and time delay. Hysteresis compensation should address these instrumental issues to improve the performance in practice.

1.2 Objective

In the specialization project, [Stakvik, 2013], conducted during the autumn of 2013, the AFM and nanopositioning problem was described. Additionally, several common actuators in AFM applications were presented, in addition to mathematical models for hysteresis, mainly focusing on the Preisach operator. Furthermore, identification methods for the Preisach operator were studied in detail.A selection of the theoretical results from this work is presented in Chapter 2.

The Master's presents a continuation of the work from the specialization project, focusing on modeling, identification and inversion of the Preisach operator. To achieve this, a literature study was conducted, focusing on both modeling of the Preisach model, and common inversion methods described in the literature. In order to gain the best possible model of the hysteresis, experimental design was studied in the context of hysteresis identification, focusing on the design of the input signal.

Experiments were performed to study the accuracy of the inversion scheme, and how to maximize the information gained from an experiment. Implementational aspects of inversion of the Preisach operator were studied, based on experimental results. Furthermore, effects which influence the identification of the Preisach operator are presented focusing on the negative phase introduced by low pass filters and time delay in the signal loop.

1.3 Outline

Chapter 2 presents the AFM and the nanopositioning problem by introducing control issues as creep, hysteresis and vibrations. Furthermore, it includes a survey of different hysteresis models is described, with focus on the mathematical description of each model. Finally, a short review of identification methods for hysteresis models is given.

Chapter 3 studies a discrete and continuous variation of the Preisach model in detail, with focus on understanding how an input signal affects the Preisach plane for each method. Moreover, an implementation scheme for calculating the output of these methods are

proposed. The method for calculating the continuous output is proposed by the author.

Chapter 4 introduces a general approach for inversion of hysteresis, consisting of generating a right inverse for the hysteresis model. Moreover, inversion algorithms for both the inverse and continuous method are presented in detail.

Chapter 5 provides an overview of general experimental design theory. This chapter also presents a mathematical definition for obtaining a persistence excited input signal for identification of hysteresis models, before the experimental setup is presented and a well defined experiment for hysteresis identification is introduced.

Chapter 6 gives a short overview of different factors that affects the identification of hysteresis. Negative phase of low pass filters, creep, actuator dynamics and time delay are illustrated and some thoughts of how this influences the identification are given.

Chapter 7 presents the results from each of the previous chapters, including both hysteresis models, inversion, experimental design, implementational aspects, and negative phase effects. Each of these points are discussed as the results are presented.

Chapter 8 provides concluding remarks on the results of this thesis, in addition to some proposed fields for further work.

Chapter 2

Background Information

This chapter introduces the nanopositioning problem, the AFM, hysteresis models and identification methods for hysteresis. The material presented here is mostly a modified version of Stakvik [2013], with some exceptions listed below. Among other topics, the Preisach model for hysteresis is described in detail, which is vital for understanding the methods described in Chapter 3

The most of this chapter is based on Stakvik [2013], however Section 2.2.4 and 2.3.3 are new contributions. In addition some material is added in Section 2.2.3 and 2.3.5.

2.1 Atomic Force Microscopy

The AFM was invented by Binnig et al. [1986], and was a revolutionary advance in the field of nanotechnology. From this, it has been possible to measure, image and manipulate samples in the nanometer range. AFM belongs to the larger class of SPM, which uses a probe for scanning the surface of a sample. There are several modes for controlling the probe position, but in general the sample will move under the probe in the xy plane, while the z-direction is controlled with a feedback so that the probe moves up and down. An image is constructed from the measurements of the z-position of the probe.

The first type of SPM, the STM, was constructed and presented by Binnig et al. [1982]. The invention of the STM was the stepping stone for the development of the AFM a few years later, awarding the inventors the Nobel Prize in physics in 1986. STM is based on the idea of quantum tunneling, which is to send a current of electrons from a probe against a sample, where the current will be a function of the distance between the tip of the probe and the

sample. If the current is kept constant, the distance will also be constant, and the samples height profile can be found by monitoring the z-position of the probe tip. However, the STM requires a high vacuum environment in order to yield good image quality, these challenges were discussed by Alexander et al. [1989]. Moreover, the sample needs to be conductive in order for the electrons to pass through the sample. These limitations gave raise to the idea of the AFM, which resolves some of the challenges with the STM.

The AFM was built on the same principles as the STM, however, the methods of measurements differ from the ones used in the STM. Instead of sending electrones against the sample surface, the probe will be in direct contact with the surface. This creates a physical force between the probe, which is placed on the tip of a cantilever, and the surface. The topography of the sample can then be measured from various methods depending on what mode the AFM user has chosen. Alexander et al. [1989] discuss the force estimation method of the AFM, where a laser beam is reflected from the back of the cantilever onto an optical photo diode, see Figure 2.1. When the height of the sample structure increases, the cantilever will bend backwards, as a result the laser beam will be reflected to another point on the photo diode, and the structure is detected.

Devasia et al. [2007] mention that the probe can damage the sample surface if the force becomes large enough, as a consequence, the force is controlled to a constant level by a feedback loop. Thus either the probe or sample will be raised or lowered in order to keep the force constant, depending on the type of microscope. The height structure can then be measured from the z-position of the probe or sample.

2.1.1 Contact Mode AFM

The theory described in this section is mainly based on the work of Abramovitch et al. [2007] and Ragazzon [2013]. In contact mode, also called the constant force mode, the cantilever and sample material are in direct contact. Both attractive and repulsive forces are present between the two objects. Abramovitch et al. [2007] mention several different models for describing the force between the probe and the sample, for instance the Lennard-Jones potential model

$$F(r) = k_1 \left[-\left(\frac{\sigma}{r}\right)^2 + \frac{1}{30}\left(\frac{\sigma}{r}\right)^8 \right] \tag{2.1}$$

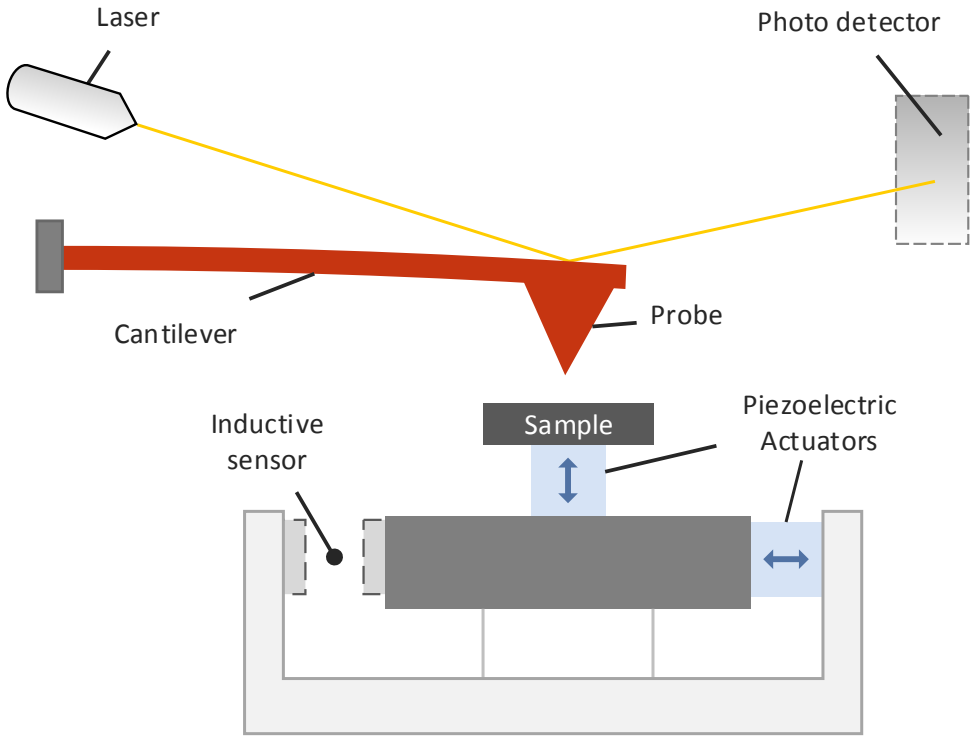


Figure 2.1: A typical AFM device setup, [Ragazzon, 2013].

where σ is a parameter that describes the force interaction, r is the distance between the sample and the cantilever tip, and k_1 is a constant that depends on the geometries of the sample and tip. $F(r)$ is the force relation between the sample and the tip, depending on the variable r. The first term in Equation 2.1 is the attractive force due to the Van der Waals' forces, and the second term is the repulsive force.

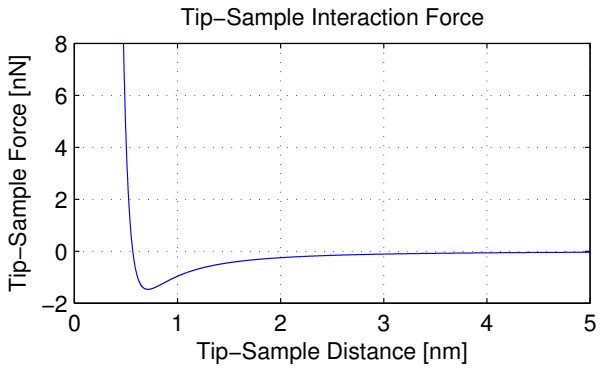


Figure 2.2: Tip-sample interaction force, with $k_1 = 10^{-9}$ and $\sigma = 10^{-9}$, [Ragazzon, 2013].

Theoretically the solution of Equation 2.1 could be found analytically, but often the parameters are unknown or varying. Therefore a frequently used method is to control the force by a feedback loop, based on the measurements of the absolute position, in order to keep the force constant, [Ragazzon, 2013]. I.e. when the force is constant, the distance between

the sample and tip also will be constant. With such a controller the absolute position of the z-axis will represent the surface topography, while a P-controller can be used as the feedback controller. The set point for the force control is typically set to the beginning of the attractive region, that is when the tip-sample force in Figure 2.2 is negative.

The cantilever dynamics are generally complex, and it is non-trivial to model it. The stationary conditions of the cantilever, on the other hand, can be simplified by Hooke's law, where the cantilever is modeled as a spring

$$F_c = -k \cdot d \tag{2.2}$$

where F_c is the force, k the spring constant, and d the deflection from the resting position. The spring constant of the cantilever will influence how the shape of the force curve from Equation 2.1 looks. With a soft spring, the force curve can have two stable points, where the attractive and repulsive forces will be equal. Consequently, in order to have only one stable point, a stiffer spring is recommended for use with constant force mode.

This problem also motivates the dynamic mode methods of measuring the topography of the sample material, which is to use an oscillating cantilever and measuring the changes in the amplitude or frequency to map the surface. These methods will be explained in more detail in the next section.

2.1.2 Dynamic Mode AFM

There are two different dynamic mode methods for measuring the topographical surface structure of a sample material, the amplitude modulation AFM (AM-AFM) and frequency modulation AFM (FM-AFM).

In order to detect the topography of the sample material in the AM-AFM mode, also called non-contact mode AFM, Garcia and Perez [2002] suggest the use of a stiff cantilever which oscillates near its resonance frequency. Thus, if the amplitude of the oscillations are measured, an estimate of the topography can be calculated. As a result of this non-contact mode method, the strain on the cantilever tip and the surface will be greatly reduced. In conclusion, this method is very useful in applications where the sample is contact sensitive. Moreover, this method increases the accuracy of the imaging, therefore, the method is often used in applications where the sample is placed in air or liquid. In contrast, the even more

accurate FM-AFM method is often used in ultra high vacuum (UHV) applications, which makes it possible to reach true atomic resolutions.

Garcia and Perez [2002] also discuss the limitations with the AM-AFM scanning method, when trying to achieve true atomic resolution in UHV. The bandwidth of the scanning routine depends on how accurate one want to scan a sample material. Generally, vacuum applications require high resolution, limiting the bandwidth of the system. Garcia calculates that, if AM-AFM is used in UHV environments, one has to wait 2 seconds for a reliable amplitude measurement. These challenges have motivated the development of the FM-AFM scheme.

While the AM-AFM mode uses only a single feedback circuit, the FM-AFM mode requires an additional circuit for driving the cantilever with constant amplitude, [Ragazzon, 2013]. When the amplitude is kept constant, the changes in frequency can be used as the feedback signal to the z-axis controller. As a result, this type of feedback loop requires very accurate instrumentation in order to detect small changes in the frequency. Therefore, this mode is mostly used in vacuum operating conditions, where these requirements can easier be fulfilled.

2.1.3 Nano Scale Actuators

In order to scan a sample surface in an AFM application, it is necessary to have actuators that are accurate in the nanometer scale. Several different actuators for this purpose exist, Devasia et al. [2007] mention, for instance, piezoelectric and magnetostrictive actuators, while Iyer et al. [2005] also mention shape memory alloys as a possible choice for nano scale actuators. This section will give a short review of some actuators used in nanopositioning applications.

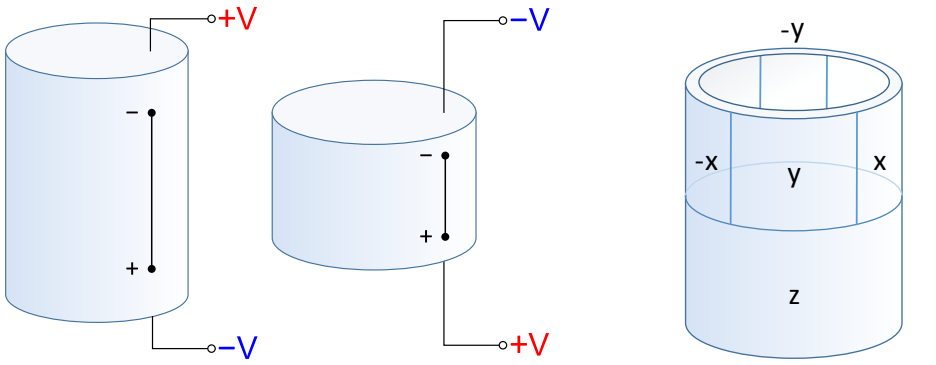
In general, most of the actuators used in nanopositioning exhibit nonlinearities like hysteresis and creep, these phenomenons will be discussed in more detail in Section 2.1.5.

Piezoelectric Actuator

The piezoelectric effect has been known since 1880, when it was discovered by the Currie brothers, Pierre and Jacques, [Ballato, 1996]. They registered that when a certain material was subjected to pressure, it produced a small amount of electricity, which is often named the direct piezoelectric effect. A few years later, the so-called converse piezoelectric effect was discovered, where an electric voltage changes the dimensions of the material, this effect

is illustrated in Figure 2.3a

Devasia et al. [2007] refer to piezoelectric actuators as ubiquitous in nanopositioning devices, such as the SPM and AFM. They have excellent operating bandwidth and can deliver large mechanical forces. The size of these actuators are generally very small, as a result, the operating range is also restricted. Consequently, the use of such actuators are limited to small, but accurate instruments.



(a) The converse piezoelectric effect on a stack actuator shown by applying a voltage V.

(b) Piezo tube actuator.

Figure 2.3: Piezoelectric actuators, [Ragazzon, 2013].

Abramovitch et al. [2007] report that piezoelectric actuators are by far the most common type of actuator in AFM applications, and especially the three degree-of-freedom piezo tube shown in Figure 2.3b. In such piezo tubes, the x and y axis of the piezo tube are actuated by four electrodes placed around the cylinder, while the z-direction is actuated by inner electrodes, [Ragazzon, 2013]. One of the challenges with the piezo tube actuators is that the axes are coupled, as a result, the z-height varies when a voltage is applied to the x and y directions.

To avoid the coupling problem, a piezoelectric stack actuator can be used in stead of the piezo tube actuator. This actuator can only expand and contract in one dimension. Three stack-actuators can then replace a single piezotube actuator, and the xy-axes and z-axis will be decoupled.

The two dominating issues in control of piezoelectric actuators, are the nonlinearities of creep and hysteresis. Hysteresis in piezoelectric actuators appears between applied voltage and induced charge, [Eielsen, 2012]. In addition, a high frequency reference signal will excite vibration modes in the material, which makes control challenging.

Magnetostrictive Actuator

The magnetostrictive actuator is somewhat different from the piezoelectric actuator. An external magnetic field has the same effect on a magnetostrictive material, as the voltage has on a piezoelectric material. The study of Ueno et al. [2003] provides an example of the use of the magnetostrictive actuator in a nanopositioning application.

In this type of actuator the magnetic field is produced from a coil with a current passing through it. The generated magnetic field will then cause the dimensions of the magnetostrictive material to change. By controlling the current through the coil, the actuators position can be controlled precisely.

Devasia et al. [2007] mention that rods of magnetostrictive material has been used as actuators in the industry because of their ability to convert electrical energy into mechanical energy. The most known magnetostrictive material is perhaps the terfenol-D, which possesses a high magnetostriction coefficient.

The power consumption for magnetostrictive materials are greater than for piezoelectric actuators, however, the material also offers a larger displacement range than piezoelectric materials. The same control challenges concerning the nonlinearity of hysteresis are present in magnetostrictive and piezoelectric actuators.

Shape Memory Alloy

The review of Wei et al. [1998] discuss shape memory materials in detail, which include shape-memory alloys, shape memory cheramics and shape memory polymers. These materials may sense thermal, mechanical, magnetic or electric stimulus and exhibit actuation. Therefore, these materials can be used as actuators in nanopositioning applications, [Sadeghzadeh et al., 2012].

Callister and Rethwisch [2011] characterize shape memory alloys as materials with two different phases, that is, they may have two crystal structures and the shape memory effect involves phase transformation between them. The phase of the material is dependent on the temperature, i.e. a high temperature corresponds to one phase, while a low temperature corresponds to the second phase. According to Callister and Rethwisch [2011], a deformed shape memory alloy which experience temperatures below a specific limit will stay in its deformed shape. When the temperature increases, the material will transform back to its original shape. Generally, the upper and lower limits for transformations between the phases are

different, i.e the phase transforms at different temperatures, depending on the current phase of the material. As a result, a hysteresis effect can be observed between in the transformation between the phases.

Sadeghzadeh et al. [2012] mention ferromagnetic shape memory alloys as possible actuators in nanopositioning applications. The shape of this material changes when submitted to a moderate magnetic field, and can experience dimensional changes up to 6%, which is larger than what magnetostrictive and piezoelectric materials experience, [Sadeghzadeh et al., 2012]. When the applied magnetic field is removed, the dimensional change will not be reversed if the temperature is below the lower phase limit described in the previous paragraph.

2.1.4 Sensors in Nanopositioning Devices

If feedback control for nanopositioning devices is applied, the tracking performance is heavily dependent on the sensor noise and bandwidth, [Fleming et al., 2008]. There exists several different sensors suitable for nanopositioning applications, among them inductive, piezoelectric, capacitive, optical decoders, laser interferometers, strain gages and the optical lever method described earlier. Due to the high cost, interferometers are rarely employed in commercial applications, however their positioning accuracy is excellent, as a result they are used in calibration of less expensive sensors [Korpelainen and Lassila, 2007; Dixson et al., 2000]. Piezoelectric sensors are based on the direct piezoelectric effect, where the material produces a charge when a pressure is applied. This section will give a brief explanation of capacitive, thermal and inductive sensors.

Capacitive Sensors

Capacitive position sensors are widely used in the industry because they provide a relatively simple technique to implement a noncontact measurement [Devasia et al., 2007]. The principle of this type of sensor, is that movement of an electrode, a shuttle, relative to a fixed electrode, a stator, causes a capacitive change. From this change in capacitance, a voltage signal can be created from an electric circuit, which is a measure of the displacement of the shuttle. In order to achieve nanometer position resolution, the electric circuit must be able to detect capacitance variations of 0.1 fF (femtofarad). As a result, the most critical component in the capacitive position sensor is the electric circuit design. If properly grounded, this

method offers the greatest resolution and signal to noise ratio, second only to the interferometer, [Fleming et al., 2008].

Thermal Sensors

The principle for the thermal position sensor is that a current flows through a heater, and subsequently increases the temperature in the device, which can be used for position measurements. To sense a displacement of an object, the temperature dependence of the resistance in a silicon crystal is exploited. A fraction of the heat generated by the heater is conducted through the ambient air into the scan table. Displacement of the scan table will then give rise to a change in the cooling mechanics, which results in a temperature change in the heater. This temperature change will then change the electrical resistance in the silicon, and the displacement can be calculated. This is more throughly explained in Devasia et al. [2007].

Inductive Sensors

Inductive sensors are commonly applied in nanopositioning applications, due to their ease of use. Generally, these sensors are built with two copper coils winded around two magnetic cores, [Defoy et al., 2013]. If a coil is placed a given distance away from a conductive target, a change in the distance will result in an induced current which reduces the original field. From this reduction the position and movement of the target can be determined. Fleming [2011] reports that the most recent generation of AFM's use either a capacitive or an inductive sensor to directly measure the vertical topography.

Optical Sensors

Optical sensors are based on the principle of optics, and the various methods ranges from pointwise techniques, such as laser triangulation and laser radar, to wholefield techniques such as projected fringes, [Chien et al., 2006]. In this wide range of sensor methods, the optical lever, proposed by Alexander et al. [1989] and depicted in Figure 2.1, is a commonly applied technique for determining the topography of a sample. Another optical position sensor is the optical encoder, which in later years has seen more use in nanopositioning applications, [Zhang et al., 2004].

2.1.5 Creep and Hysteresis

The most dominant control issues in nanopositioning systems, are the nonlinear dynamics such as creep, hysteresis and vibration. When a more rapid scanning speed is required, these nonlinearities are exited. As a consequence, these effects must be compensated for in order to achieve both accurate and fast scanning, [Croft et al., 2001].

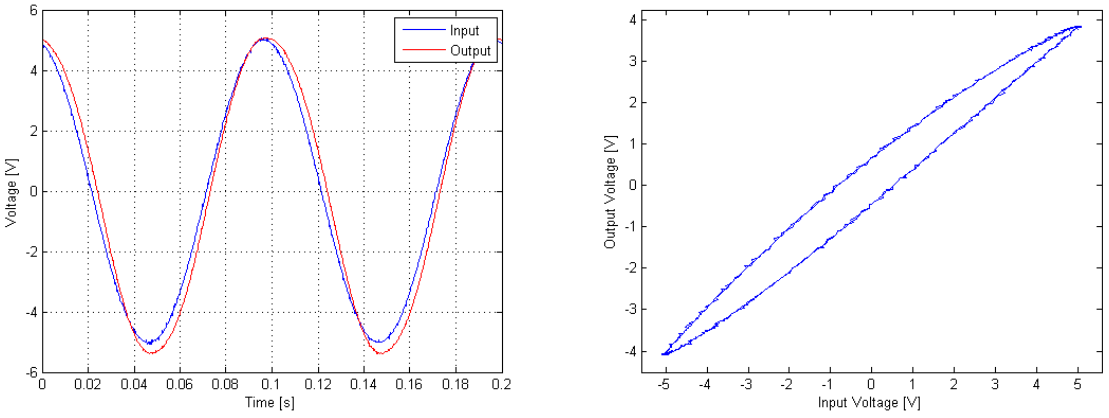
Creep

Ragazzon [2013] mentions that creep in piezoelectric transducers can lead to significant loss of precision in open-loop control. This rate-dependent phenomenon stems from remnant polarization changes after a voltage has been applied to a piezo element. This will then result in a slow creep, often with a time constant in the order of minutes.

This effect can either be described by a model, or be compensated for by a feedback controller.

Hysteresis

Devasia et al. [2007] refer to hysteresis as the main form of nonlinearities in piezoelectric transducers. The word hysteresis originates from a Greek term meaning "to lag behind", and describes a relationship between inputs and outputs of a certain system. The "lagging behind" property is illustrated in in Figure 2.4a, where the output values are delayed compared to the input values.



(a) The output lags behind the input.

(b) Input voltage plotted against output voltage.

Figure 2.4: Classical hysteresis.

A system with scalar inputs and outputs is said to exhibit rate independent hysteresis if

two properties are satisfied, [Iyer and Shirley, 2004]

a) The outputs of the system do not depend on the rate at which the input is applied.

b) The outputs of the system at a certain time depend on the past history of the input function.

The definition of rate-independent hysteresis implies that there exist a rate-dependent hysteresis as well, often referred to as dynamical hysteresis. However, the phenomenon of hysteresis is defined as rate-independent, and dynamic rate-dependent effects should not be included in the modeling of hysteresis. These effects should be described by other models, where the rate-dependent properties can be satisfied. Nevertheless, there are hysteresis models who incorporate such rate-dependent effects and consequently is not only description of hysteresis.

Hysteresis was originally described by James A. Ewing in 1881 in a study of ferromagnetism phenomena, [Iyer and Tan, 2009]. There are various properties that arise in ferromagnetic hysteresis, such as major-loops, minor-loop closure, energy dissipation and monotonicity. As described by Jiles and Atherton [1986], the major-loop of the material is obtained by cycling the input with a progressively increasing value, starting from the demagnetized state. These major-loops are widely used to define the properties of a material exhibiting hysteresis, and will describe for which input values hysteresis is present.

Minor-loop closure, on the other hand, is a property observed for input values inside the major-loop bounds, that is, inside the region the material exhibit hysteresis. The minor loop closure property can be explained by an example, if the input is increased from a value v_1 to a value $v_2 > v_1$ over a time interval $[t_1, t_2]$, and then decreased back to the same input value v_1 in the time interval $[t_2, t_3]$, the output will satisfy $y(t_1) = y(t_3)$. Energy dissipation can be observed as the area in the center of the hysteresis loop, illustrated in Figure 2.4b, and is a result of internal friction, [Iyer and Tan, 2009]. The output of a material that exhibit hysteresis can only increase when the input is increasing, and decreases only when the input is decreasing, for this reason hysteresis is said to be a monotone function.

Hysteresis operators appear, in a general engineering application, connected with systems of differential equations. From this, Iyer and Tan [2009] define a system with hysteresis to be a series, parallel or feedback connection of a hysteresis operator and a dynamical system. The nanopositioning systems discussed in this study will therefore be assumed to be a dynamical system preceded by a hysteresis, as shown in Figure 2.5.

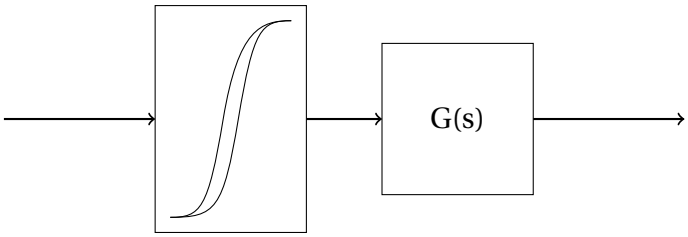


Figure 2.5: System with hysteresis in series with a dynamical system.

Even though hysteresis, in control applications, is an issue, the phenomenon can also be useful for other applications. For instance, the gas adsorption-desorption technique for determining pore size and pore size distribution in porous materials, such as membranes, shows a hysteresis phenomenon, [Mulder, 1997]. By first filling a membrane with gas, starting with a low, but increasing pressure, all pores in the membrane will be filled with gas. Then, by decreasing the pressure again, the gas will flow out of the pores. By plotting the volume of the membrane against the corresponding pressure, a hysteresis effect can be observed. From this effect, the pore shape of the membrane can be determined.

2.2 Hysteresis Models

As mentioned in the previous chapter, piezoelectric actuators are the dominating actuators in AFM applications. Devasia et al. [2007] mention hysteresis as the main form of nonlinearity in piezoelectric transducers. In fact, Abramovitch et al. [2007] note that for long-range positioning these effects can significantly reduce the accuracy. In order to compensate for the nonlinearity it is important to be able to describe the phenomenon with a precise model. When a model is established, there are several methods that can be applied to solve the control problem. Devasia et al. [2007] mention feedback, feedforward, iterative control and sensorless control as the most common approaches.

The models described in this chapter will either be of a physical, or a phenomenological nature. A physical model is based on the laws of physics, however, derivations of such models for the memory nature of hysteresis can be an arduous task, and the obtained models are often too complex in order to be used in practical applications, [Ismail et al., 2009]. Therefore, many models for hysteresis are based on some physical understanding of the system, in addition to some black-box modeling. Such models are often called semi-physical or phenomenological models, since they are partly based on the physical phenomenon.

In this chapter, different models for hysteresis will be described and discussed. The mathematical properties of the various models will be presented and explained. At the end of the chapter, the different hysteresis models will be weighted against each other, in order to decide which model should be applied in the identification in the following chapters. The mathematical models presented in this chapter are primarily based on Macki et al. [1993]; Mayergoyz [2003]; Eielsen [2012]; Visintin [1994] and Al Janaideh et al. [2011].

2.2.1 The Duhem Model

The Duhem model for hysteresis dates back to 1897, in a study by Pierre Duhem, and focuses on the fact that the output can only change its character when the input changes direction. The model is a phenomenological approach for describing the hysteresis phenomena, and uses an integral operator of differential equation to model a relation for hysteresis. The material presented in this section is primarily based on Eielsen [2012] and Macki et al. [1993].

A Duhem model is given by the general differential equation

$$\dot{\eta} = f_I(u,\eta)(\dot{u})^+ + f_D(u,\eta)(\dot{u})^-,\ \eta(0) = \eta_0 \tag{2.3}$$

where η is the output of the model, f_I and f_D corresponds to function valid for an increasing input and decreasing input respectively, while the expressions $(\dot{u})^+$ and $(\dot{u})^-$ are interpreted using the definitions

$$(\dot{u})^+ := \frac{|\dot{u}| + \dot{u}}{2}$$

$$(\dot{u})^- := \frac{|\dot{u}| - \dot{u}}{2}$$

and which satisfy

$$(\dot{u})^+ + (\dot{u})^- = |\dot{u}|.$$

This means that $(\dot{u})^+ = \dot{u}, \dot{u} > 0$, and $(\dot{u})^+ = 0, \dot{u} < 0$. Conversely, $(\dot{u})^- = \dot{u},\ \dot{u} < 0$, and $(\dot{u})^- = 0, \dot{u} > 0$. From this it is seen that the function f_I will give the output when the input is increasing, and f_D gives the output when the input is decreasing. Models on the form of Equation 2.3, are useful as phenomenological models, since the functions and parameters can be tuned in order to fit experimental results.

2.2.2 The Coleman-Hodgdon Model

The Duhem model was investigated by Coleman and Hodgdon in relationship with ferromagnetic hysteresis, but it could also be used to describe ferroelectric hysteresis. The model is described as

$$\dot{\eta} = \beta\dot{u} - \alpha\eta|\dot{u}| + \gamma|\dot{u}|u,\ \eta(0) = \eta_0 \tag{2.4}$$

where u is the input, η the output and α and β are unknown variables. Compared to the Duhem model, one can observe that

$$f_I(u,\eta) = \beta - \alpha\eta + \gamma u \tag{2.5}$$

$$f_D(u,\eta) = \beta + \alpha\eta - \gamma u \tag{2.6}$$

In order for the output to change behavior only when the input changes direction, the parameters must satisfy the following conditions; $\alpha > 0$, $\beta > 0$, $\frac{\gamma}{\alpha} > \beta$, and $\frac{\gamma}{\alpha} \leq 2\beta$, this is, according to Eielsen [2012], corresponding to the laws of thermodynamics. The conditions on the parameters also ensures that the slope of the output $\dot{\eta}$ has the same sign as the slope $\dot{u}$.

The input-output map generated by Equation 2.4, has a symmetric stationary response to periodic inputs witch are monotonically increasing and decreasing between two extrema. The model is therefore best suited to describe hysteresis responses that are dominant symmetric, and for such periodic input signals. Although these types of input signals may yield the best results, the model is defined for a larger class of input signals. The signal u must be bounded, piecewise continuous and connected. This also implies that the time derivative $\dot{u}$ exists and is bounded, i.e., $u \in C^0$. This includes signals such as triangle-waves or low pass filtered steps and square-waves, but not unfiltered steps and square-waves.

For use of the Coleman-Hodgdon model for hysteresis compensation, the article of Du et al. [2009] employ this model together with a sliding mode control approach. In addition, feedforward compensation of hysteresis using the Coleman-Hodgdon model is treated in both Eielsen et al. [2010] and Eielsen et al. [2012]

Connection to Friction Models

The Duhem model described in Equation 2.3, depends on the sign of the input. If the input increases, the function f_I describes the output. Likewise, when the input decreases the

output is given from the function f_D. Differential equations on this form are therefore commonly described as a Duhem model.

Such models can be used in descriptions of physical phenomenas, for instance dynamic friction models. Egeland and Gravdahl [2002] describe several dynamical models for friction in their book *Modeling and Simulation for Automatic Control*, among them the Dahl model and the LuGre model. These models can be described as a Duhem model, as an example, this will be shown for the Dahl friction model, [Egeland and Gravdahl, 2002]

$$\frac{dF}{dt} = \sigma\left(\dot{x} - |\dot{x}|\frac{F}{F_c}\right) \tag{2.7}$$

where $\dot{x}$ is the velocity, and x the position, F_c is the Coulomb force, σ the spring stiffness and F the friction. This can further be written as

$$\frac{dF}{dt} = \sigma\frac{|\dot{x}|}{F_c}\left(F_c \cdot \text{sgn}(\dot{x}) - F\right) \tag{2.8}$$

This can be split into the two functions, f_I and f_D from Equation 2.3, one for $\text{sgn}(\dot{x}) > 0$ and one for $\text{sgn}(\dot{x}) < 0$

$$f_I = \sigma|\dot{x}| - \sigma|\dot{x}|\frac{F}{F_c} \tag{2.9}$$

$$f_D = -\sigma|\dot{x}| - \sigma|\dot{x}|\frac{F}{F_c} \tag{2.10}$$

If the Dahl model in Equation 2.7 is compared to the Colman-Hodgdon model in Equation 2.4, one can easily notice that x in the Dahl model plays the same role as u in the Colman-Hodgdon model. The same derivation as above can be done for the LuGre model as well.

Another model frequently employed to describe friction, is the Bouc-Wen model, which is studied in detail in the survey of Ismail et al. [2009]. In many systems exhibiting hysteresis, the output is lagging behind the input. By contrast, in friction models the friction force will be the output and the displacement the input. Hence, the direction of the hysteresis loop will be opposite of a system where the force is the input and the displacement the output. As a result, friction models, like Bouc-Wen or LuGre, can be used directly as an inverse model for hysteresis, [Ismail et al., 2009].

2.2.3 Preisach-Type Hysteresis Models

One of the perhaps most powerful scalar models for hysteresis is the Preisach model, which was proposed by Ferenc Preisach in 1935, [Eielsen, 2012]. The approach of Preiasch was based on some plausible hypothesis concerning the physical mechanisms of magnetization, [Mayergoyz, 2003]. As a result, the model was first regarded as a physical model for hysteresis. However, in later years a more generalized description has recognized it as a phenomenological model. The use of this model today is not limited to hysteresis based on magnetization, and has seen use for several materials exhibiting hysteresis, for instance the piezoelectric actuator.

The idea introduced by Preisach apply a basic building block called a Preisach hysteron or Preisach operator as a description of a single hysteresis element. By superpositioning these operators one can build complex hysteresis operators that exhibit more subtle properties than the elementary operators, [Iyer and Tan, 2009]. There are several operators presented in the literature, the most common is the delayed relay operator, proposed by Preisach, furthermore the play, stop and Krasnosel'skii-Pokrovskii operator has also been proposed. These operators generate the foundation for different Preisach-type hysteresis models, and will be discussed in detail in the following sections.

Since the various hysteresis models make use of the same model structure as Preisach, a generic term for these operators is known as Preisach-type operators, [Tan and Bennani, 2008; Iyer and Tan, 2009] and Rosenbaum et al. [2010]. On the other hand, since the models are based on operators, they could also be called operator-based hysteresis models. In these models, the hysteretic output is created by integration over a Preisach plane, and the idea will be examined and explained in Section 2.2.3, even though the same idea applies to all Preisach-type operators.

The Preisach Operator

As mentioned in the previous section, the Preisach model was first regarded as a physical model. The reason for this is that Preisach proposed this model in order to represent scalar ferromagnetism. In later years, however, it has also seen a widespread use to model other hysteresis phenomena, [Eielsen, 2012], and the general formulation is therefore no longer directly related to any specific physical phenomena. A discussion about this can be found in Mayergoyz [2003]. The material discussed in this section will mainly be based on the material

described in Mayergoyz [2003]; Eielsen [2012]; Macki et al. [1993] and Iyer and Tan [2009], if not stated otherwise.

The basis of the Preisach model is the delayed relay operator, $\mathcal{R}_{\alpha,\beta}$, which is defined for an arbitrary continuous input, $u(t)$ as

$$y(t) = \mathcal{R}_{\alpha,\beta}[u(t),\xi] = \begin{cases} -1, & \text{for } u(t) > \alpha \\ \xi, & \text{for } \beta < u(t) < \alpha \\ 1, & \text{for } u(t) < \beta \end{cases} \tag{2.11}$$

where $y(t)$ is the output of the relay, ξ is the state of the relay, and can be defined as $\xi = y(t-1)$, and α and β corresponds to the "up" and "down" switching values for the input, respectively, as shown in Figure 2.6.

The output $y(t)$ is dependent on both the input $u(t)$, and the memory of the relay, in the form of the state $\xi \in \{-1, 1\}$. Based on this, the output can be calculated when the state of the relay at $t = 0$ is known.

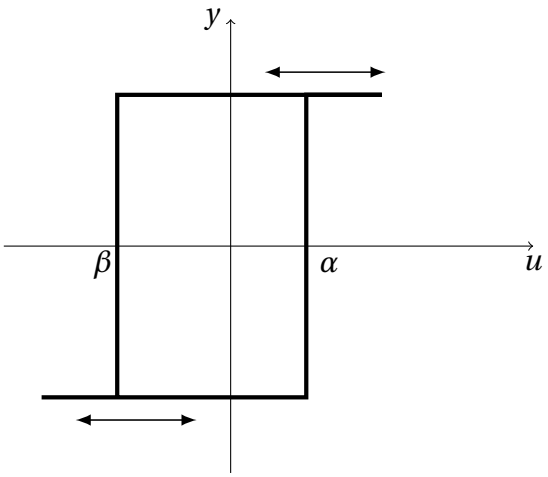


Figure 2.6: Delayed relay operator.

In the sequel it is assumed that $\alpha \geq \beta$, which is quite natural from a physical point of view since the "up" switch correspond to a larger input value than the "down" switch. For instance, as the input is monotonically increased, the output value switches when the input u increases past the value α. Similarly, the output will drop down to -1 when the input has decreased past the β value again.

If an infinite set of parallel delayed relay operators, with the $\alpha \geq \beta$ assumption satisfied, the resulting Preisach hysteresis model characterizes the model output $\eta(t)$ of a system ex-

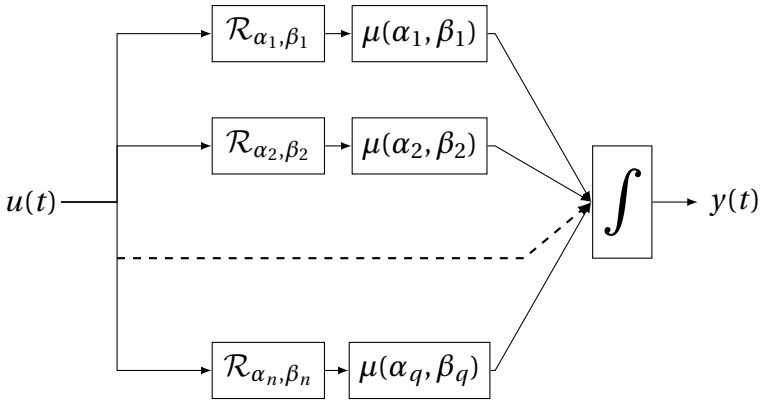


Figure 2.7: Superposition idea of Preisach model.

hibiting hysteresis as the superposition weighted relays

$$\eta(t) = \iint_{\alpha\geq\beta} \mu(\alpha,\beta)\mathcal{R}_{\alpha,\beta}[u(t),\xi]d\alpha d\beta \tag{2.12}$$

where $\mathcal{R}_{\alpha,\beta}$ is the relay output, $\mu(\alpha,\beta)$ is a weight function, called the Preisach density function, with an unique element corresponding to each delayed relay operator. This weigh function must generally be identified from experimental data. The superposition idea is illustrated in Figure 2.7. Each relay element represent hysteresis nonlinearities with local memories, with the state $\xi_i \in \{-1,1\}$ as noted earlier, where i is the ith relay. Physically, this can be interpreted as the switching behavior of individual dipoles within a material. The weight function $\mu(\alpha,\beta)$ is an integrable function defined for the half-plane

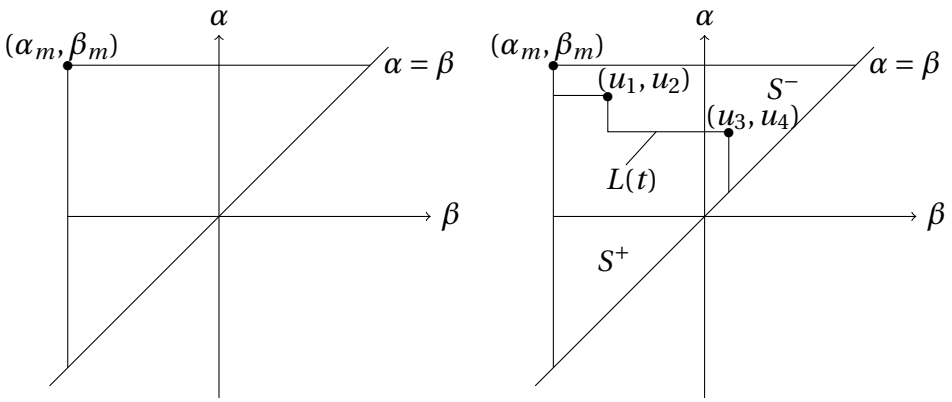
$$S = \{(\alpha,\beta) : \alpha \geq \beta, \alpha \leq \alpha_m, \beta \leq \beta_m\} \tag{2.13}$$

The set S is the Preisach plane, while $\mu(\alpha,\beta)$ is the Preisach density function. For every unique point (α,β), there exists an unique relay $\mathcal{R}_{\alpha,\beta}$. The α_m corresponds to the highest α described in the model, while β_m refers to the lowest β value, i.e. $\alpha < \alpha_m$ and $\beta > \beta_m$ for all α and β.

This Preisach plane is illustrated in Figure 2.8a. By assuming an input $u(0) < \beta_m$, where β_m is the lowest β value, then all the delayed relays will have an output equal to -1. In other words, all the $\mathcal{R}_{\alpha,\beta}$ operators are put in the "down" position.

The subdivision of the Preisach plane can be described with a series of input maximums and minimums. Firstly, by assuming a monotonically increased input from $t_0 = 0$ to t_1, at some maximum value u_1, the delayed relay operators $\mathcal{R}_{\alpha,\beta}$ with switching values α, less than u_1 are being turned into the "up" position. The output of these operators will then become

equal to +1. Secondly, the input monotonically decreases until the time reaches $t = t_2$ with some minimum value u_2. As the input is being decreased, all relays with "down" switching values β above the current input value u_2 will be turned back to the "down" position. If this procedure is repeated with a maximum $u_3 < u_1$ and a minimum $u_4 > u_2$, the resulting Preisach plane can be shown in Figure 2.8b.



(a) Preisach plane.

(b) Subdivision of Preisach plane.

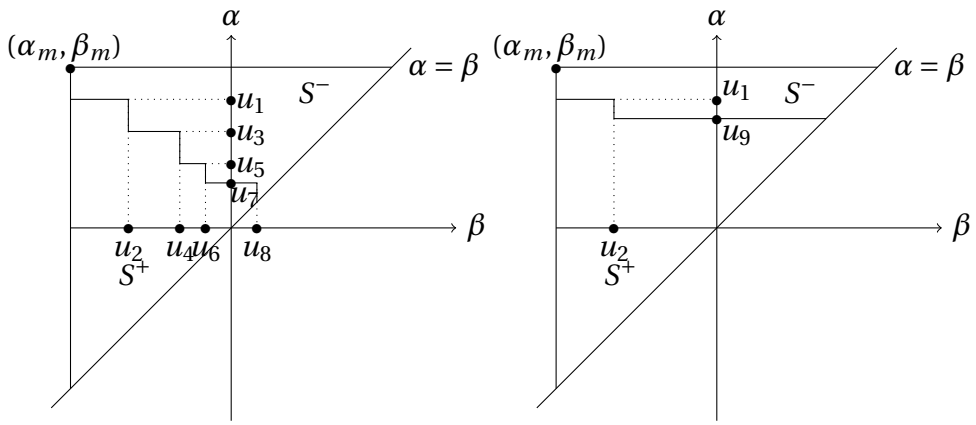
Figure 2.8: Preisach plane.

As a result of the given input, the Preisach plane S will geometrically be subdivided into two regions; $S^+(t)$ and $S^-(t)$, where each of these sets corresponds to the relays in the "up" and "down" position respectively as

$$S^+(t) = \{(\alpha,\beta) \in S : \mathcal{R}_{\alpha,\beta} u(t) = 1\} \tag{2.14}$$

$$S^-(t) = \{(\alpha,\beta) \in S : \mathcal{R}_{\alpha,\beta} u(t) = -1\} \tag{2.15}$$

Geometrically, this leads to a subdivision of the half-plane S into two sets: $S^+(t)$ which consists of (α,β) pairs where the corresponding relay operators $\mathcal{R}_{\alpha,\beta}$ are in the "up" position, and $S^-(t)$ consists of points (α,β) where the relay operators are in the "down" position.



(a) The descending staircase of the Preisach state.

(b) The wiping out property of the Preisach operator.

Figure 2.9: Properties of the Preisach plane.

The boundary $L(t)$ between $S^+(t)$ and $S^-(t)$ is called the Preisach state, and can be seen as a descending "staircase" in Figures 2.8b and 2.9a. This state corresponds to which relay operators that are turned "on" and "off", and it describes the past local minimums and maximums of the input. However, as described in Mayergoyz [2003], the model does not accumulate all past extremum values of the input. Some of them are wiped out by subsequent input variations. This wiping out property is best understood with an example. Assume an input sequence of decreasing local maximas: $\{u_1, u_3, u_5, u_7\}$ and an increasing sequence $\{u_2, u_4, u_6, u_8\}$ of local input minima. A figure describing the Preisach plane with this inputs can be seen in Figure 2.9a. Now, the input $u(t)$ is monotonically increased until it reaches some maximum value u_9 that is above u_3. This monotonic increase of the input results in a boundary $L(t)$ that moves upwards until the maximum at u_9 is reached. From Figure 2.9b it can be seen that all vertices with α switching values below u_9 have now been wiped out. The wiping out property also occurs for monotonically decreasing inputs.

Another property of the Preisach model is congruency, which means that all minor hysteresis loops corresponding to back and forth variations of inputs between the same two consecutive extremum values are congruent. That is, that two hysteresis loops with the same input values will be similar. In addition, the Preisach model is rate-independent, which means that the output only depends on the levels of the input, and not on the speed of the input variations.

Inversion of the Preisach Operator In order to use the Preisach model as hysteresis compensation, it is necessary to calculate the inverse of the system. Tan and Bennani [2008]

point out that the inversion problem for Preisach operator generally do not have any analytical solutions, and therefore has to be calculated with an iterative method. The inversion of Preisach-type operators are discussed in detail in Iyer et al. [2005]. In Brokate and Sprekels [1996], a detailed discussion of existence of the inverse of the Preisach operator is given.

Hysteresis compensation of smart materials using the Preisach operator is common in the literature. A selection of studies using or discussing the Preisach operator in hysteresis compensation is Tan and Baras [2004]; Iyer and Tan [2009]; Iyer et al. [2005] and Hughes and Wen [1997]. These studies also discuss and implement methods for identification of the Preisach density function. Moreover, the inversion problem for the Preisach operator will be discussed in detail in Chapter 4, where a inversion algorithm based on a closest match procedure is shown.

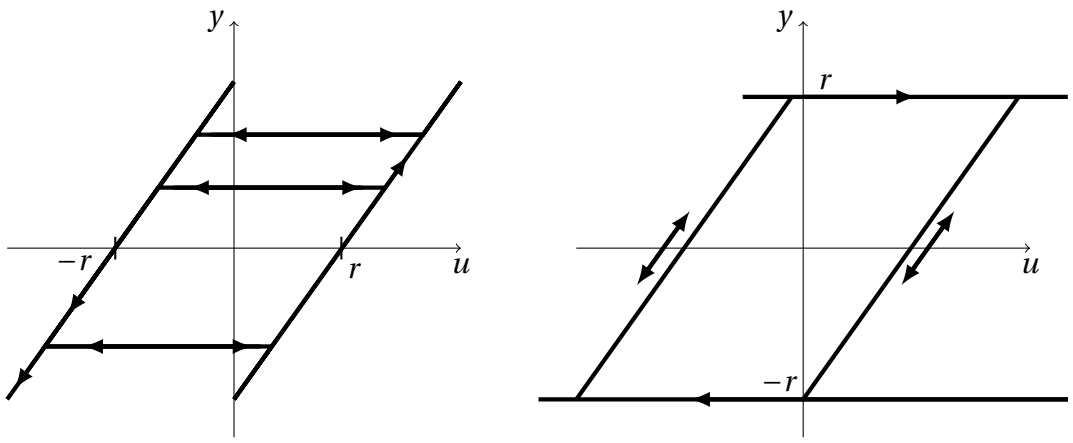
The Prandtl-Ishlinskii Operator

This section will present the Prandtl-Ishlinskii (PI) operator, which makes use of the play and stop operators as basic building blocks. The model was first derived by Prandtl in 1928, before Ishlinskii independently described the same model sixteen years later. It was mainly proposed as a model for plasticity-elasticity in physics, however, the model has also been used in the description of other kinds of hysteresis phenomena. The material presented in this section will primarily be based on the articles of Al Janaideh et al. [2011]; Macki et al. [1993], and the book by Brokate and Sprekels [1996].

The PI hysteresis model is also a phenomenological based model, since it does not directly describe a physical phenomenon. Like in the Preisach model, this model is described as a superposition of elementary operators. In contrast to the Preisach operator, the PI operator are the play and stop operators, which is parameterized by a single threshold variable, unlike the (α, β) pair in the Preisach model.

The play operator is shown in Figure 2.10a, and is characterized by the input $u(t)$, output $y(t)$ and the threshold value r. Furthermore, the stop operator is given in Figure 2.10b and is parameterized with the same coordinates as the play operator. Since the positive and negative threshold is described by the same value r, this model can only be used to describe symmetric hysteresis shapes. However, the work of Al Janaideh et al. [2011] present an modified version of the PI operator, which can describe asymmetric hysteresis as well.

The output of the play operator given a continuous input function $u(t)$ can be analytically



(a) Play hysteresis operator. (b) Stop hysteresis operator.

Figure 2.10: Prandtl-Ishlinskii operator.

expressed as

$$y(t) = F_r[u(t), \xi] = \begin{cases} \max(u(t) - r, \xi), & \text{for } u(t) > u(t-1) \\ \min(u(t) + r, \xi), & \text{for } u(t) < u(t-1) \\ \xi, & \text{for } u(t) = u(t-1) \end{cases} \tag{2.16}$$

where $y(t)$ is the output of the play operator, and F_r is the output function corresponding to $\mathcal{R}_{\alpha,\beta}$, in the Preisach model. ξ is the state of the operator, defined as $\xi = y(t-1)$,

With the same notation, the output of the stop operator is given by

$$y(t) = E_r[u(t), \xi] = \begin{cases} \min(r, u(t) - u(t-1) + \xi), & \text{for } u(t) > u(t-1) \\ \max(-r, u(t) - u(t-1) - \xi), & \text{for } u(t) < u(t-1) \\ \xi, & \text{for } u(t) = u(t-1) \end{cases} \tag{2.17}$$

where $y(t)$ is the output of the stop operator, E_r is the output function corresponding to $\mathcal{R}_{\alpha,\beta}$, in the Preisach model and ξ is still the state of the relay.

In order to model hysteresis, the PI model utilizes the output from the play and stop operator $y(t)$ and a density function $p_r(r)$, where the threshold r is applied in the PI model as $0 = r_0 < r_1 < \cdots < r_n = R$. The relationship of the output $\eta(t)$ and a continuous input $u(t)$ was described by Brokate and Sprekels [1996]

$$\eta(t) = \int_0^R p_r(r) F_r[u(t), \xi] dr \tag{2.18}$$

where $p_r(r)$ is an integrable positive density function corresponding to the Preisach density function $\mu(\alpha,\beta)$ in Equation 2.11, and $F_r[u(t),\xi]$ can be replaced with $E_r[u(t),\xi]$ in order to obtain the output for the stop operator. As a result, $p_r(r)$ serves as a weighting for the play or stop operators. Due to the unknown nature of this density function, it has to be identified from experimental data in a similar way as $\mu(\alpha,\beta)$.

From Figure 2.10 it can be seen that the PI operator are symmetric. Thus, the weight function in Equation 2.18 can be expressed in the same notation as the Preisach weight function, if $\alpha_i = r_i$, $\beta_i = -r_i$ and $\alpha_i = -\beta_i$, where $i = [0,n]$. Then Equation 2.18 can be rewritten to

$$\eta(t) = \iint_{\alpha\geq\beta} \mu(\alpha,\beta) F_{\alpha,\beta}[u(t),\xi]\, d\alpha\, d\beta \tag{2.19}$$

Where the play operator $F[u(t),\xi]$ can be replaced with the stop operator $E[u(t),\xi]$.

Inversion of the PI Operator Even though the PI model, unlike the Preisach model, is analytically invertible and therefore can conveniently be implemented in a feedforward routine, Bashash and Jalili [2007] note that the model lacks accuracy. For instance, the PI model assumes a symmetric hysteresis structure, which in general is not the case. As a result, this model can not be applied for predicting asymmetric input-output hysteresis, which is observed in both shape memory alloy and magnetostrictive actuators. Moreover, the unbounded nature of the play operator does not permit the PI model applications for saturation properties, which is again observed in, for instance, shape memory alloy actuators.

For information about hysteresis compensation using the PI operator the articles of Bashash and Jalili [2007] and Al Janaideh et al. [2011] can be used as a starting point.

The Krasnosel'skii-Pokrovskii Operator

Another Preisach-type operator is the Krasnosel'skii-Pokrovskii (KP) operator. This model uses the same pair of thresholds (α,β) as the Preisach operator, however a KP operator, based on the ridge function, $\delta(u(t))$, is used instead of the delayed relay operator. The ridge function is defined as, [Riccardi et al., 2012]

$$\delta(u) = \begin{cases} -1, & \text{for } u < 0 \\ -1 + \frac{2u}{a}, & \text{for } 0 \leq u \leq a \\ 1, & \text{for } u > a \end{cases} \tag{2.20}$$

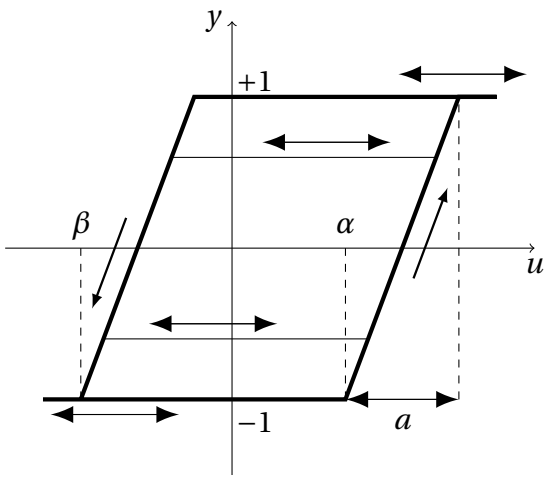


Figure 2.11: The KP operator.

where a is the distance shown in Figure 2.11, and u is a scalar value.

Given the state ξ, with initial value $\xi = \xi_0$, and an input function $u(t)$ the output, $y(t)$ of this elementary operator is given as, [Riccardi et al., 2012]

$$y(t) = KP_{\alpha,\beta}[u(t),\xi] = \begin{cases} \max\{\xi, \delta(u(t)-\alpha)\}, & \text{for } du(t)/dt \geq 0 \\ \min\{\xi, \delta(u(t)-\beta)\}, & \text{for } du(t)/dt < 0 \end{cases} \tag{2.21}$$

where $\delta(u(t))$ is the ridge function and the memory term ξ is the state of the operator, and can be defined as $\xi = y(t-1)$.

As in the Preisach operator and the PI operator, the output of the KP operator, $\eta(t)$, given a continuous input $u(t)$ can be written

$$\eta(t) = \iint_{\alpha\geq\beta} \mu(\alpha,\beta) KP_{\alpha,\beta}[u,\xi](t)\, d\alpha\, d\beta \tag{2.22}$$

where $\mu(\alpha,\beta)$ corresponds to the same type of density function as in the Preisach operator, with the same unknown nature as well and $KP_{\alpha,\beta}$ is the output of each element.

Inversion of KP Operator In contrast to the PI operator the KP operator does not have any analytical inverse, except in some very special situations, [Tan and Bennani, 2008]. Tan and Bennani [2008] propose a recursive solution to the inversion problem and implemented this on a field programmable gate array (FPGA). Riccardi et al. [2012] use the inversion algorithm presented by Tan and Bennani [2008] in a position system for magnetic shape memory alloys (MSMAs). Galinaitis and Rogers [1998] focus on the problem of determining a model for

hysteresis and inversion of this model using the KP operator.

Summary of Priesach-Like Operators

This section presents a table which summarizes the different Preisach-type operators presented above, in a similar notation. The general output of these operator based hysteresis models were presented in Equation 2.12

$$\eta(t) = \iint_{\alpha \geq \beta} \mu(\alpha,\beta)\mathcal{R}_{\alpha,\beta}[u(t),\xi]d\alpha d\beta \tag{2.23}$$

where $\mathcal{R}_{\alpha,\beta}$ is the operator output to be substituted with different operators, $F_{\alpha,\beta}[u(t),\xi]$, $E_{\alpha,\beta}[u(t),\xi]$ or $KP_{\alpha,\beta}[u(t),\xi]$, and $\mu(\alpha,\beta)$ is the weight function on the Preisach plane. The various operators presented, are summarized in Table 2.1, where the state ξ is defined as

$$\xi = y(t-1) \tag{2.24}$$

and ξ_0 is known.

Table 2.1: Overview of different operators for hysteresis.

Hysteretic Operator	Operator Output	Weight Function
Preisach Operator	$\mathcal{R}_{\alpha,\beta}[u(t),\xi] = \begin{cases} -1, & \text{for } u(t) > \alpha \\ \xi, & \text{for } \beta < u(t) < \alpha \\ 1, & \text{for } u(t) < \beta \end{cases}$	$\mu(\alpha,\beta)$, where $\alpha \geq \beta$
Play Operator	$F_{\alpha,\beta}[u(t),\xi] = \begin{cases} \max(u(t)+\beta,\xi), & \text{for } u(t) > u(t-1) \\ \min(u(t)+\alpha,\xi), & \text{for } u(t) < u(t-1) \\ \xi, & \text{for } u(t) = u(t-1) \end{cases}$	$\mu(\alpha,\beta)$, where $\alpha \geq \beta$, $\alpha_i = r_i$ and $\beta_i = -r_i$
Stop Operator	$E_{\alpha,\beta}[u(t),\xi] = \begin{cases} \min(\alpha, u(t)-u(t-1)+\xi), & \text{for } u(t) > u(t-1) \\ \max(\beta, u(t)-u(t-1)-\xi), & \text{for } u(t) < u(t-1) \\ \xi, & \text{for } u(t) = u(t-1) \end{cases}$	$\mu(\alpha,\beta)$, where $\alpha \geq \beta$, $\alpha_i = r_i$ and $\beta_i = -r_i$
KP Operator	$KP_{\alpha,\beta}[u(t),\xi] = \begin{cases} \max(\xi, \delta(u(t)-\alpha)), & \text{for } du(t)/dt \geq 0 \\ \min(\xi, \delta(u(t)-\beta)), & \text{for } du(t)/dt < 0 \end{cases}$	$\mu(\alpha,\beta)$, where $\alpha \geq \beta$

2.2.4 Preisach Operator in (r,s) Coordinates

The (β,α) coordinates used in the previous section are not always the most convenient description of the Preisach operator, instead the (r,s) coordinates can be used, where r and s is described as

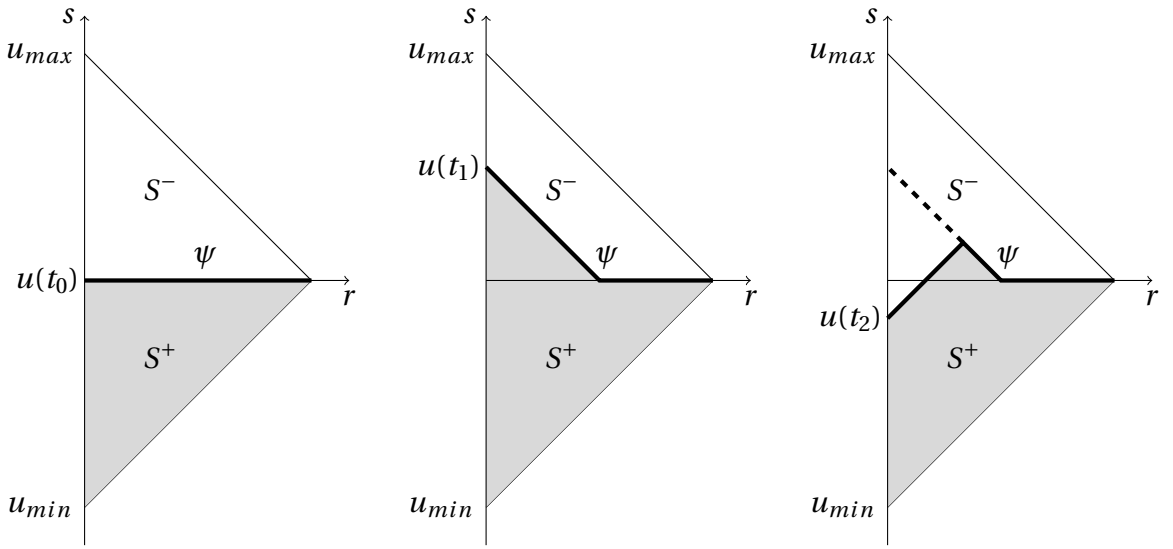


Figure 2.12: The memory curve of the Preisach model when using (r,s) coordinates. The figure is motivated from Gorbet et al. [1999].

$$r = \frac{\alpha-\beta}{2} \tag{2.25}$$

$$s = \frac{\alpha+\beta}{2} \tag{2.26}$$

where the output of the Preisach operator can be expressed in terms of (r,s) as [Tan, 2002]

$$y(t) = \Gamma[u,\psi](t) = \int_0^\infty \int_{-\infty}^\infty w(r,s)\gamma_{s-r,s+r}[u,\zeta_{\psi_0}(s-r,s+r)](t)\,ds\,dr \tag{2.27}$$

where $y(t)$ is the output, Γ is the output function based on the input $u(t)$ and the memory curve $\psi(t)$ and $w(r,s)$ is the output corresponding to the Preisach density operator $\mu(\alpha,\beta)$, described in (r,s) coordinates. The memory curve ψ has the same information as the states ξ, however it is described as the past minimums and maximums instead of the value of each state. Tan [2002] also states that the memory curve $\psi(t)$ is a graph of a function of r, and that the previous input, $u(t)$, can be found from the memory curve from $\psi(t)$. Further, Equation 2.27 can be rewritten as

$$y(t) = \Gamma[u,\psi](t) = \nu_0 - 2\int_0^\infty \int_{\psi[t](r)}^\infty w(r,s)\,ds\,dr \tag{2.28}$$

where ν_0 is the output corresponding to the positive saturation.

The memory curve of the Preisach model for (r,s) coordinates is shown in Figure 2.12. The boundary of the triangle in the figure corresponds to the left and right boundaries in Figure 2.8a, which depends on the maximum and minimum values on the input. The memory curve has a slope of ± 1 depending on an increasing or decreasing input. For instance, assume that the input $u(t_0) = 0$ is the initial condition of the system, illustrated in Figure 2.12a, where the thick line represents the memory curve ψ, while the gray area is the resulting S^+ region corresponding to the positive relays. Now assume that the input $u(t)$ is increased monotonically until the input reaches $u(t_1)$. The resulting memory curve is shown in Figure 2.12b. Assume that the input is monotonically decreased from $u(t_1)$ to $u(t_2)$, then the area of S^+ decreases as well. The dashed line in the last figure shows how the memory curve changed from the previous input.

The Preisach plane in (r,s) coordinates is explained in more detail in the work of Gorbet et al. [1999], where, for instance, the wiping out property of the Preisach model for (r,s) coordinates is illustrated.

Since the slope of the memory curve in Figure 2.12 corresponds to the delayed relay operator in Figure 2.6, other Preisach-type operators, as the play-, stop- and KP-operator, can give a memory curve with different behavior than the delayed relay operator does. The (α,β) will for the most part be used in this work, however, if more appropriate, the output of the Preisach operator will be described with (r,s) coordinates as $w(r,s)$.

2.2.5 Jiles Athernon Model

In contrast to the hysteresis models given in the previous sections, the Jiles-Athernon Model is a physically motivated model, based on magnetic hysteresis. The model attempts to describe the relationship between the magnetization M, within a material, and an applied magnetic field strength H, [Rosenbaum et al., 2010]. The model was originally introduces in the paper of Jiles and Atherton [1986], with the intention of describing hysteresis in magnetic material. Rosenbaum et al. [2010] presented this model as

$$\frac{dM}{dH} = \frac{(1-c)(M_{an}-M_{irr})}{k\cdot \operatorname{sgn}\left(\frac{dH}{dt}\right) - \alpha(M_{an}-M_{irr})} + c\frac{dM_{an}}{dH} \tag{2.29}$$

where M_{an} is given by

$$M_{an} = M_S\left(\coth\left(\frac{H+\alpha M}{a}\right) - \frac{a}{H+\alpha M}\right) \tag{2.30}$$

and $\coth(\cdot)$ is the hyperbolic cotangent function. The parameters a, α, c, k and the saturation magnetization M_S are determined from measured hysteresis characteristics, [Sadowski et al., 2002]. Rosenbaum et al. [2010] note that the discrete form of Equation 2.29 and 2.30 can be computed numerically for each (H, M) pair within a hysteresis loop.

2.2.6 Maxwell Resistive Capacitance Model

Another phenomenological hysteresis model is the Maxwell resistive capacitance model. This model was proposed by Goldfarb and Celanovic [1997], and is motivated by static friction models. Goldfarb and Celanovic [1997] argue that in most physical systems hysteretic behavior is the result of energy storage, which is fundamentally coupled to rate independent dissipation. Furthermore, they model the energy storage as an ideal linear spring and the rate independent dissipation as a pure Coulomb friction element. Such a model of an *elasto-slide* element, which consists of a massless block that is subjected to Coulomb friction, can then be described by Goldfarb and Celanovic [1997], which the rest of the section will be based on as well

$$F = \begin{cases} k(x - x_b), & \text{if } |k(x - x_b)| < f \\ f\ \text{sgn}(\dot{x}) \text{ and } x_b = x - \frac{f}{k}\ \text{sgn}(\dot{x}), & \text{else} \end{cases} \tag{2.31}$$

where x is the input displacement, $\dot{x}$ the change in displacement, x_b the initial displacement, F is the output force, k is the stiffness of the spring, and $f = \mu N$ is the breakaway friction force of a block. For a detailed description of friction modeling the reader is advised to consult Egeland and Gravdahl [2002] which treat the topic in detail.

When a number of these elasto-slide elements are placed in parallel, each subjected to an incrementally larger normal force (i.e. each has an incrementally larger breakaway force), the model becomes a piecewise linear approximation of rate independent hysteresis. When the limit of the numbers of elasto-slide elements goes to infinity, Goldfarb and Celanovic [1997] refer to this model as the generalized Maxwell slip.

If the model is expressed in the electrical domain, the Maxwell slip can be expressed as follows

$$V_i = \begin{cases} \frac{q-q_{b_i}}{C_i}, & \text{if } \left|\frac{q-q_{b_i}}{C_i}\right| < v_i \\ v_i \operatorname{sgn}(\dot{q}) \text{ and } q_{b_i} = q - C_i v_i \operatorname{sgn}(\dot{q}), & \text{else} \end{cases} \tag{2.32}$$

$$v_{mrc} = \sum_{i=1}^{n} V_i \tag{2.33}$$

where v_{mrc} is the hysteresis voltage drop on the total charge q, $\dot{q}$, the change is charge, and all other variables are the electrical analogs of those given for the single elasto-slide element in Equation 2.31.

Goldfarb and Celanovic [1997] also note that the Maxwell resistive capacitance model is not domain specific. Therefore it can represent any rate-independent hysteretic relationship between a generalized force and generalized displacement. Consequently, in addition to force and displacement, the generalized Maxwell model can represent rate independent hysteresis between voltage and charge, temperature and entropy and magnetomotive force and magnetic flux. For instance, Juhász et al. [2011] apply the Maxwell model for hysteresis compensation on piezoelectric stack actuators.

2.3 Identification Methods

In the previous chapter, several models for describing the hysteresis nonlinearity were discussed and explained. They can all be applied as a description of the nonlinearity in nanopositioning devices, however, some models are more used than others. The Preisach operator is one of the most common operators for the hysteretic nonlinearity, as a result the Preisach model will be employed in the identification procedure. In order to use this model as a description of a hysteretic system it is necessary to identify the hysteresis dynamics in the specific actuator of use. The key parameter in such an identification is the Preisach denisty function, which weight each point in the Preisach plane, [Azzerboni et al., 2004].

Mayergoyz [2003] presented a method for identifying the Preisach density function directly from measuring the first-order reversal curves, which is a complete hysteresis loop. However, this method involves a differentiation of the output signal, which is corrupted by noise. For this reason Iyer and Shirley [2004] argue that this method has limited applicability in practice, and other methods must be applied in order to estimate the density function,

this was also shown with experimental data in the paper of Henze and Rucker [2002].

Iyer and Tan [2009] mention several schemes for identification of the Preisach density function, among them both parametric and nonparametric methods. Nonparametric methods approximate the Preisach density function as a piecewise constant function on each cell in the Preisach plane. Parametric methods, on the other hand, use a model structure based on few parameters in order to describe the Preisach density function. Such models can have the form of a two dimensional probability distribution, which approximates the weights in the Preisach plane. As proposed in Bertotti [1998] these probability distributions can be the factorized-Lorentzian, Gaussian-Gaussian or Lognormal-Gaussian distributions. The motivation for this description is that the Preisach density function can have a shape that can be described by one of these probability distributions. These distribution functions will be described in Section 2.3.2.

As nonparametric methods Iyer and Tan [2009] propose to use a constrained linear least squares method. This method describes each point in the Preisach plane with a specific value, in order to achieve the weight for each unique relay. The first-order reversal method is also known as a nonparametric method, [Iyer and Tan, 2009].

In addition to the mentioned methods, which will be discussed in the following sections, there has been proposed other methods for identification of the Preisach density function. The most known of these is perhaps the neural network procedure. This method has been used in several studies, among them Adly and Abd-El-Hafiz [1998]; Natale et al. [2001] and Zhao and Tan [2006]. The last sections of the chapter will introduce the methods used for discretization and implementation of the Preisach plane, in addition to how the data for identification was gathered.

2.3.1 Constrained Least Squares

In order to use the constrained least squares method, the Preisach plane has to be discretized and the weight of each point in the plane has to be decided. By using a constrained least squares method to optimize the difference between the measured output and the estimated output the Preisach density function can be found. The result is a density function that minimizes, [Iyer and Tan, 2009]

$$\min_{\mu_{\alpha,\beta}} J = \frac{1}{2}\int_0^T \left(\hat{\eta}(t) - \iint_{\alpha\geq\beta} \mu(\beta,\alpha) R_{\alpha,\beta}[u(t),\xi]\right)^2 d\alpha d\beta \tag{2.34}$$

subject to the condition $\mu(\beta,\alpha) \geq 0$, where $\hat{\eta}(t)$ is the measured output signal, $u(t)$ the measured input signal, and the known initial memory $\xi_0 = \xi(0)$.

Iyer and Shirley [2004] discuss and use this method on experimental data. They also argue that this is the best suited approach in order to obtain the most correct approximation of the actual density function, in a least squared sense. Their argument is that this approach does not make any assumptions on how the actual density function appears, and therefore the model can capture a wide range of density function shapes. The constrained least squares method can also applied in on-line applications, this is more studied in Section 2.3.3.

2.3.2 Parametric Methods

In parametric methods for description of the Preisach density function, one assumes a specific shape of the density function. By doing this, it is possible to describe the density function with a few parameters instead of finding an unique value for each point in a discretized Preisach plane.

As mentioned earlier in this section, Bertotti [1998] refers to the factorized-Lorentzian, Gaussian-Gaussian and Lognormal-Gaussian distributions as approximations of the Preisach density function. These methods, or variations of them, have been applied as identification procedures in for instance Henze and Rucker [2002] and Azzerboni et al. [2004] and are mentioned as a possibility by Iyer and Shirley [2004] and Iyer and Tan [2009]. Even though these methods are based on probability density functions, they are not connected to any random variables. The motivation for using them, is that they can describe the shape of the Preisach density function analytically with a continuous weight.

With experimental input and output data, the identification procedure can be done by creating the distribution, when a set of initial parameters are given. By sending the measured input through the operator given by Equation 2.12, an estimated output can be obtained. Minimizing the difference between this output and the measured output by updating the parameters, will give the best parameter fit.

The models for the weight functions, will use the notation introduced in the previous chapter, i.e. $\mu(\alpha,\beta)$. In the rest of this section the different approximations of the Preisach density function will be presented, mainly based on the work by Bertotti [1998] and Henze and Rucker [2002].

Factorized-Lorentzian Distribution

If the Preisach density function is assumed as a Lorentzian distribution, the density function takes the form

$$\mu(\alpha,\beta) = \frac{2}{\pi\sigma_c^2 H_0^2\left(\frac{\pi}{2}+\arctan\left(\frac{1}{\sigma_c}\right)\right)}\frac{1}{\left[1+\left(\frac{\alpha-H_0}{\sigma_c H_0}\right)^2\right]\left[1+\left(\frac{\beta+H_0}{\sigma_c H_0}\right)^2\right]} \tag{2.35}$$

where α and β is the up and down switching values, σ_c is the standard deviation and H_0 is the position of the maximum.

The resulting identification is then reduced to finding the unknown parameters σ_c and H_0. The standard deviation σ_c refer to how fast the probability decreases from the maximum point. A small value represents a rapid decrease, while a large value represents a slow decrease. An example of this distribution is illustrated in Figure 2.13.

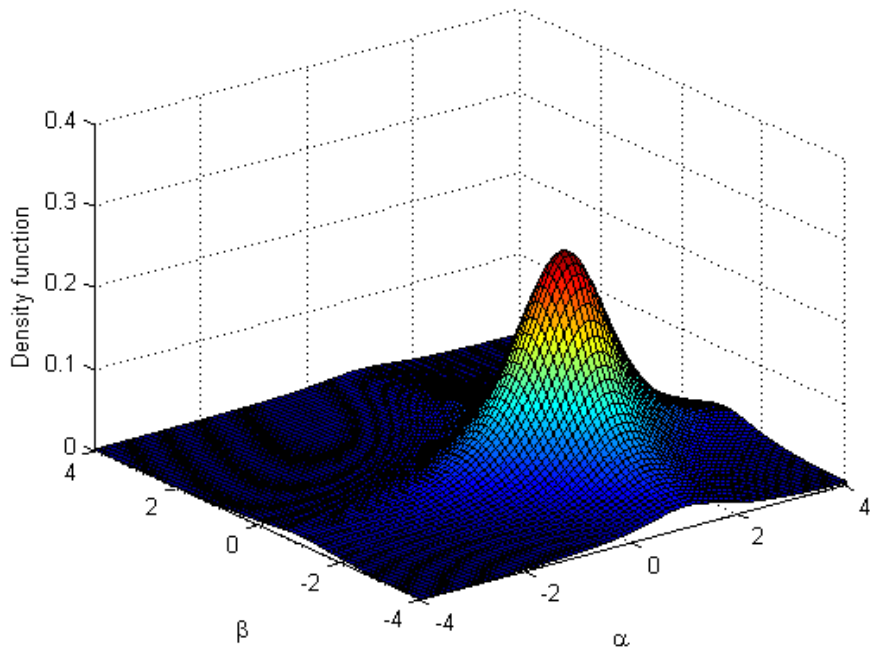


Figure 2.13: Plot of the factorized Lorentzian with $\sigma_c = 1$ and $H_0 = 1$.

Gaussian-Gaussian Distribution

The Gaussian-Gaussian distribution uses two Gaussian probability distributions to describe the approximation of the Preisach density function. Due to this, the distribution has two different standard deviations for each diagonal

$$\mu(\alpha,\beta) = \frac{1}{2\pi\sigma_c\sigma_u H_0^2}\frac{2}{\text{erf}(\sigma_c\sqrt{2})+1}\exp\left[-\frac{\left(\frac{\alpha-\beta}{2}-H_0\right)^2}{2\sigma_c^2 H_0^2}\right]\exp\left[-\frac{\left(\frac{\alpha+\beta}{2}\right)^2}{2\sigma_u^2 H_0^2}\right] \tag{2.36}$$

where σ_c and σ_u is the standard deviations on each diagonal, H_0 is the position of the maximum and the erf function is the error function defined as

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x \exp(-u^2)\,du$$

In contrast to the factorized-Lorentzian, in this distribution there are three parameters that needs to be identified, σ_c, σ_u and H_0. Due to this, the identification procedure will be somewhat more demanding, but can perhaps give a more correct estimate of the real Preisach density function. A plot of the Gaussian-Gaussian distribution is given in Figure 2.14.

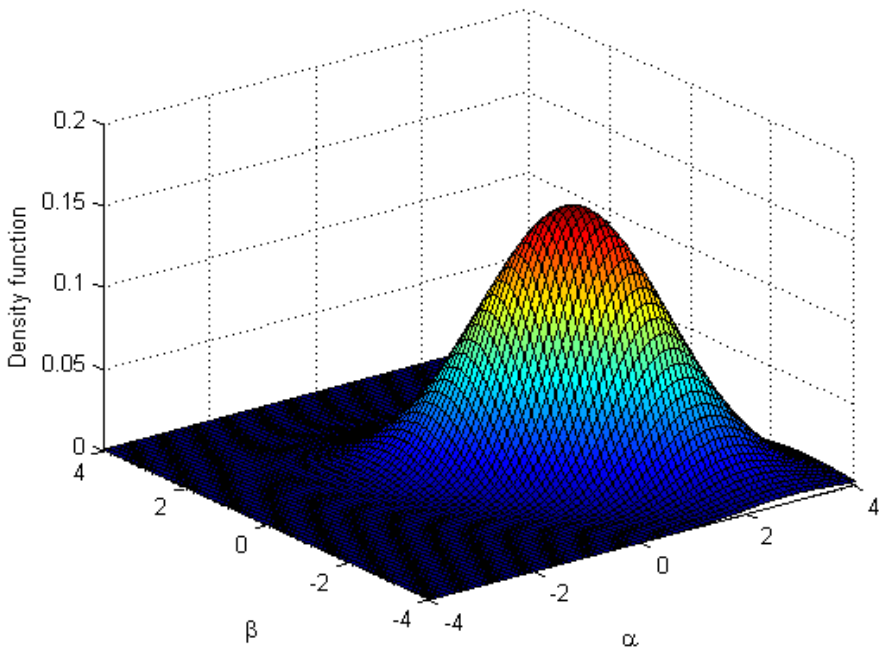


Figure 2.14: Plot of the Gaussian-Gaussian distribution with $\sigma_c = 1$, $\sigma_u = 1$ and $H_0 = 1$.

Lognormal-Gaussian Distribution

Just as in the Gaussian-Gaussian distribution, the Lognormal-Gaussian distribution is described by two different standard deviations

$$\mu(\alpha,\beta) = \frac{1}{2\pi\sigma_c\sigma_u H_0}\frac{2}{\alpha-\beta}\exp\left[-\frac{\ln\left(\frac{\alpha-\beta}{2H_0}\right)}{2\sigma_c^2}\right]\exp\left[-\frac{(\alpha+\beta)^2}{8\sigma_u^2 H_0^2}\right] \tag{2.37}$$

where σ_c and σ_u refer to the two standard deviations, while H_0 is connected to the position of the maximum.

In difference to the factorized-Lorentzian and Gaussian-Gaussian distributions, the distribution of Lognormal-Gaussian has a division term containing $(\alpha-\beta)$. This term experiences a singularity on the diagonal $\alpha=\beta$, and therefore prevents this from being calculated. By assuming the values on the diagonal equal to zero, one obtain the plot given in Figure 2.15.

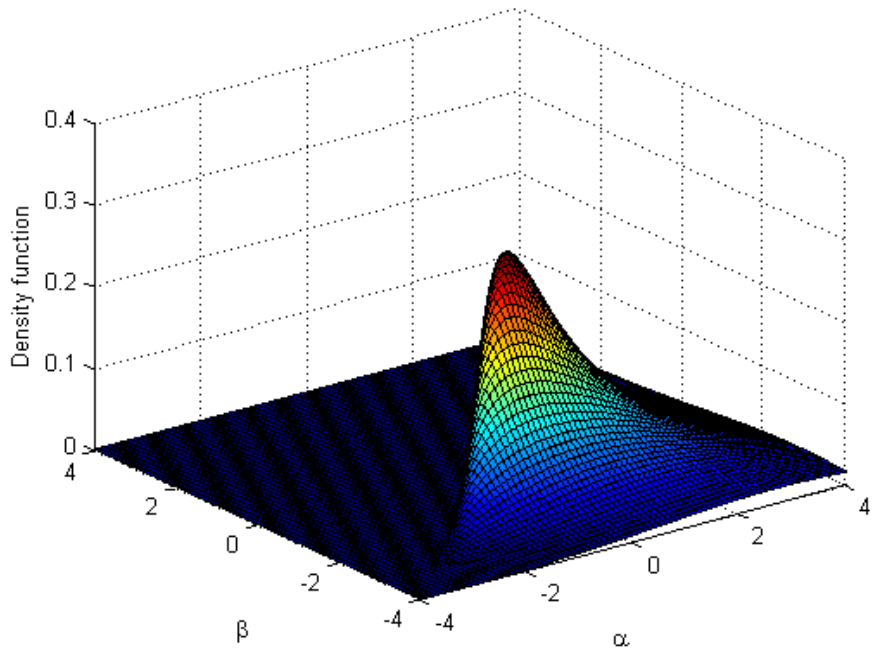


Figure 2.15: Plot of the Lognormal-Gaussian distribution with $\sigma_c = 1$, $\sigma_u = 1$ and $H_0 = 1$.

Surface Identification

If the distribution of the weight function in the real hysteretic element is known, an alternative method for estimating the parameters in the distributions given in the previous sections

could be used. By comparing the surface of the measured distribution with the surface of the proposed density distributions, the parameters could be estimated. However, the real distribution of the weight function is in general unknown, therefore, in this work, the distribution gained from the constraint least squares method will be assumed as an estimate of the real distribution.

2.3.3 On-line Identification

In many applications, for instance AFM, an on-line identification scheme for hysteresis is desirable. There are several methods used for this in the literature, and this section will mention some of the most common schemes.

The popular least square method, introduced for off-line applications in Section 2.3.1, can also been used in adaptive and on-line applications where a recursive form of Equation 2.34 is implemented. This can be the classical least squares method described in Equation 2.34, or a modified version based on a forgetting factor Eielsen et al. [2011]. An example of the least squares method used for estimation of the Coleman-Hodgdon equation, described in Section 2.2.2, in an on-line application can be seen in Eielsen et al. [2012].

Tan and Baras [2005] compare two different on-line identification methods for hysteresis. The first is an output based identification based on the well known gradient algorithm

$$\hat{\mu}[n+1] = \mu[n] - \gamma \frac{(\hat{y}[n] - y[n])W[n]}{W[n]^T W[n]} \tag{2.38}$$

where $\hat{y}$ is the estimated output, n is the time step, k is the relays in the Preisach plane, $0 < \gamma < 2$ is the adaption constant, W is the state of the relay, corresponding to a vector of all the xi states, and $\hat{\mu}$ is the estimated Preisach weight. The second algorithm is an identification based on using the time difference

$$z[n] \triangleq= y[n] - y[n-1] = (W[n] - W[n-1])^T \mu^* \tag{2.39}$$

where $z[n]$ is the difference, y and W is the same as for the gradient algorithm and μ^* is the real distribution of the Preisach weight. After calculations Tan obtained another identification scheme based on the gradient algorithm

$$\hat{\mu}[n+1] = \begin{cases} \hat{\mu} - \gamma \frac{(z[n]-z[n])V[n]}{V[n]^T V[n]}, & \text{if } V[n] \neq 0 \\ \hat{\mu}, & \text{if } V[n] = 0 \end{cases} \tag{2.40}$$

where $V[n] = W[n] - W[n-1]$ consists of the difference of each relay value between two time steps and γ still is the adaption constant described above.

The difference based method is shown to give a faster estimate in a noise-free environment, however if noise is present, the output based method is more accurate. Other adaptive identification schemes are described in detail in Ioannou and Sun [2003], where robustness of these methods are also studied.

2.3.4 Discretization of the Preisach Plane

In order to implement the Preisach model numerically, Equation 2.12 can be solved by discretization of the Preisach plane. This section will present a procedure for discretization of the Preisach plane based on the method described in Eielsen [2012].

By partitioning the Preisach plane S, illustrated in Equation 2.13, into subregions, as shown in Figure 2.16, the Preisach plane will be discretized. Within each sub-region, the relays are assumed to switch simultaneously depending on the applied input $u(\cdot)$, and the weighting values $\mu_{\alpha,\beta}$ are the same within that region. From this, Equation 2.12, takes the discrete form

$$\eta(t) = \sum_{i=1}^{n_q} \mu_i A_i \mathcal{R}_i[u,\xi](t) + \eta_0 \tag{2.41}$$

where n_q represent the total number of sub-regions, μ_i is the weighting value for the relay $\mathcal{R}_i$ in the ith sub-region of area A_i, and η_0 is the bulk contribution to the output from sub-regions outside of the Preisach plane. If the Preisach plane has been partitioned into n_h levels uniformly along the β and α axes, the total number of sub-regions is $n_q = (n_h(n_h+1))/2$. Then the sum of relays that needs to be implemented is n_q.

Chapter 3 will discuss the discretization of the Preisach model in detail. Algorithms for output calculation are shown, and the implementation procedure is described. In addition, Mayergoyz [2003] discusses several aspects of the discretization.

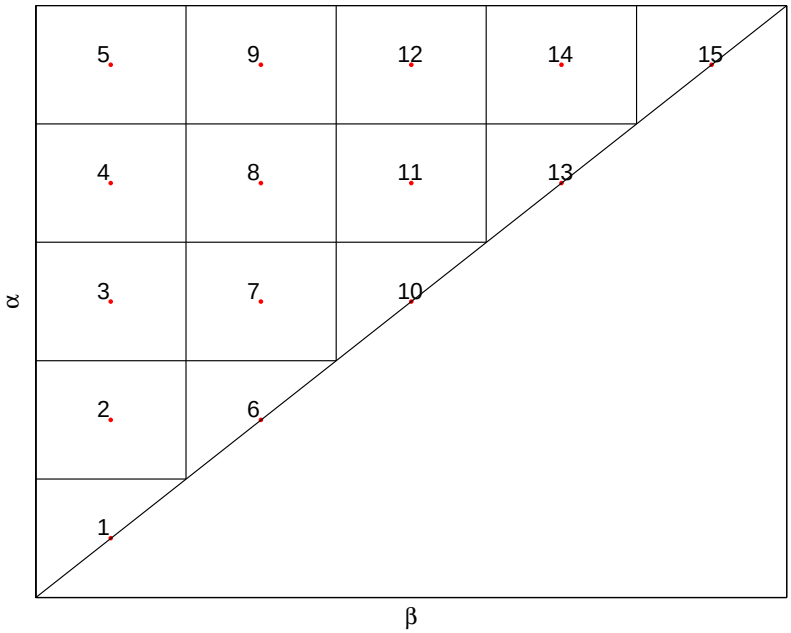


Figure 2.16: Preisach plane partitioning for $n_h = 5$ levels.

2.3.5 Implementation

When implementing the constrained least squares method, the Preisach model in Equation 2.34 have to be discretized as described in Section 2.3.4. The identification process explained below is based on the description given by Eielsen [2012].

By assuming that all possible (α, β) pairs are scaled to the maximum and minimum input values, u_M and u_m, the weight μ_i, in Equation 2.41, can be found by sampling the input $u(t)$ and output $\eta(t)$ of a system exhibiting hysteresis, in order to obtain a primitive hysteresis loop.

To fit the system with constrained least squares the time-series was used to create an over-determined system of linear equations on the form

$$\begin{bmatrix} \eta(t_1) \\ \eta(t_2) \\ \vdots \\ \eta(t_{n_t}) \end{bmatrix} = \begin{bmatrix} A_1\mathcal{R}_1u(t_1) & A_2\mathcal{R}_2u(t_1) & \cdots & A_q\mathcal{R}_qu(t_1) \\ A_1\mathcal{R}_1u(t_2) & A_2\mathcal{R}_2u(t_2) & \cdots & A_q\mathcal{R}_qu(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ A_1\mathcal{R}_1u(t_{n_t}) & A_2\mathcal{R}_2u(t_{n_t}) & \cdots & A_q\mathcal{R}_qu(t_{n_t}) \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_{n_q} \end{bmatrix} + \eta_0 \tag{2.42}$$

where A is the area, $\mathcal{R}$ the relay output. Equation 2.42 can be written on a more compact form

$$c_\eta = A_u b_\mu \tag{2.43}$$

where c_η is $n_t \times 1$, A_u is $n_t \times n_q$, b_μ is $n_q \times 1$, and $c_\eta = \boldsymbol{\eta}(t) - \eta_0$, where η_0 can be identified by subtracting this value from all measured outputs collected in the vector $\boldsymbol{\eta}(t)$. The fit for the constraint least squares method $b_\mu = [\mu_1, \mu_2 \cdots, \mu_{n_q}]^T$ was then found by using the MATLAB function *lsqnonneg* which has the constraint $b_\mu > 0$ for all elements.

Summary of Identification Methods

The identification methods mentioned are summarized in Figure 2.17. The LSQ block refers to the constrained least squares method, while curve and surface fit are the MATLAB function *lsqnonlin* applied for output data and surface data respectively.

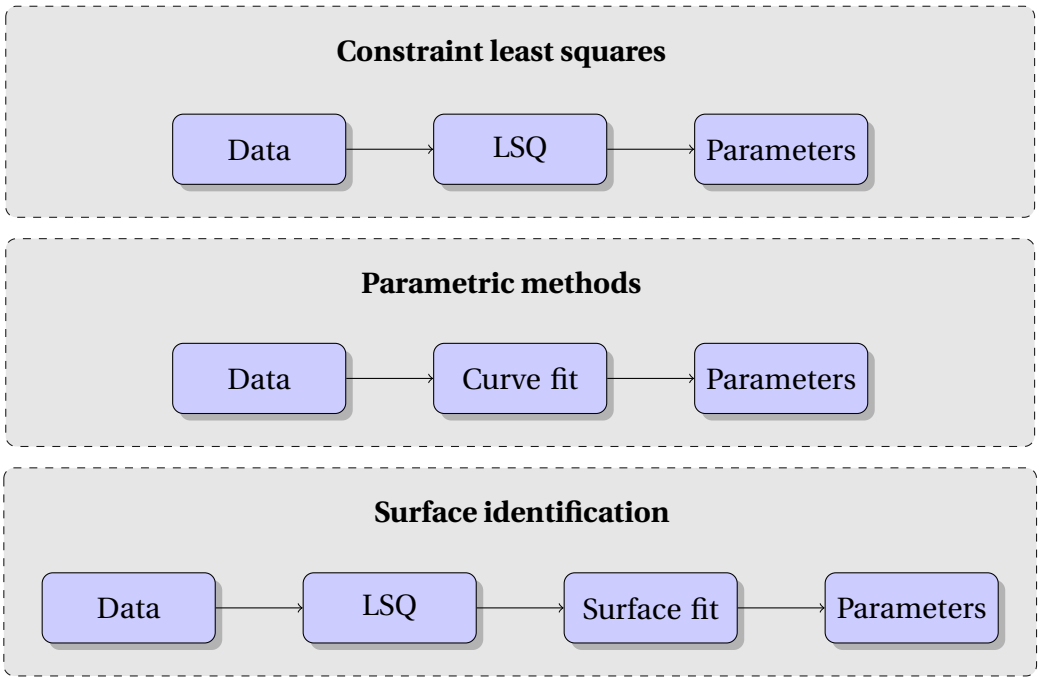


Figure 2.17: Applied identification procedures.

Chapter 3

Discrete and Continuous Preisach Model

In Section 2.2.3 the various Preisach-type hysterons were presented mathematically. In this chapter two different interpretations of the Preisach operator, i.e. the delayed relay, will be explained, namely the discrete method, with the Preisach density function $\mu(\alpha,\beta)$ consisting of point masses in the center of each cell, and a method based on a piecewise-constant density function. The last method will is referred to as the continuous method in this thesis, even though the output is only piecewise-continuous, this will be discussed in more detail in Section 3.5.1.

As mentioned in Section 2.3.4, the Preisach plane has to be discretized in order to implement the models. This applies to both the discrete and continuous methods, presented above. The output, $\eta(t)$ then becomes, as outlined in Equation 2.41

$$\eta(t) = \sum_{i=1}^{n_q} \mu_i A_i \mathcal{R}_i[u(t),\xi] + \eta_0 \tag{3.1}$$

where η_0 is the constant bulk contribution, μ_i is the Preisach density function, with one value for each cell, A_i is the area of the ith cell, and $\mathcal{R}_i[u(t),\xi]$ is the value of the relay in the ith cell. It is the computation of the Preisach relay, $\mathcal{R}$, which distinguishes the discrete and continuous methods. This will be discussed in more detail in the following section.

3.1 Comparison of the Discrete and Continuous Models

As noted above, the difference between the discrete and continuous method is based on the calculation of the $\mathcal{R}_i[u(t),\psi]$ value. In the discrete case each cell behaves like the delayed relay operator illustrated in Figure 2.6, where α and β is given as the coordinate for at the top

of the cell and the coordinate to the left of the cell respectively. This implies that in order to change a Preisach element from +1 to −1 the input needs to become lower than the β value corresponding to the left border of the cell.

In contrast to the discrete case, the continuous method requires that the signed area of each cell containing the Preisach memory curve, ψ is calculated. ψ describes all the states ξ as a curve of past minimums and maximums corresponding to the memory curve corners in the Preisach plane. Each of these cells is then separated into two regions, S^+ and S^- where they represent the area on each side of the memory curve. The signed area $w_A[i](t)$ of the ith cell in the Preisach distribution is defined as [Iyer and Tan, 2009]

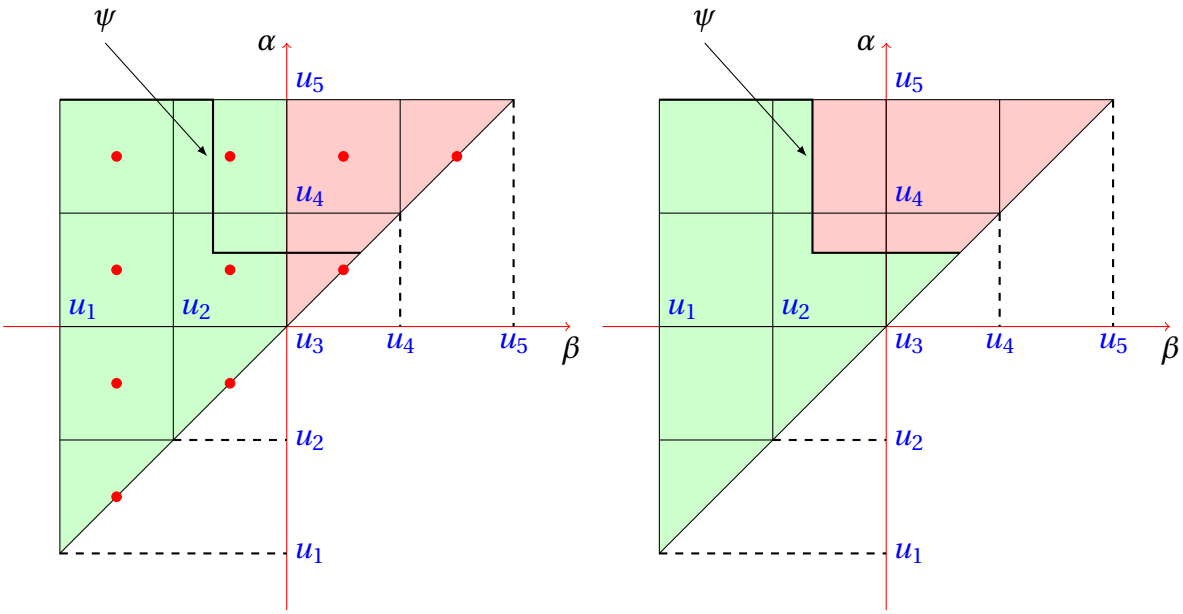
$$w_A[i](t) = \text{area of } S^+ - \text{area of } S^- \tag{3.2}$$

Both the discrete and continuous partition of the Preisach plane is illustrated in Figure 3.1. In the example in the figure, input has first been increased to the level u_5, then decreased to just above u_2, before it is increased again to almost u_4. The same input is depicted for both the discrete and continuous case, and the difference between the partitions for the discrete and continuous case can easily be recognized. In the discrete case the entire cell has the same output value until the input either raise above the α limit or decrease below the β limit, depending on which state the hysteron is in. By contrast, the continuous case changes the signed area inside the hysterons when the input changes. This means that the resolution of this model is not limited by the partitioning of the Preisach plane, since the output changes on just a small variation in the input value.

As a summary of the difference of the two methods the calculation of the $\mathcal{R}_i[u(t),\xi]$ can be described as an equation

$$\mathcal{R}_i[u(t),\xi] = \begin{cases} y[i](t) & \text{For discrete method} \\ w_A[i](t) & \text{For continuous method} \end{cases} \tag{3.3}$$

where the signed area $w_A[i](t)$ is given in Equation 3.2 and $y[i](t)$ is given as the output of



(a) Discrete partition of the Preisach plane. (b) Continuous partition of the Preisach plane.

Figure 3.1: The discrete and continuous partition of the Preisach plane compared. In Figure (a) the identified Preisach density function has all it's weight in the center of the cell (illustrated with the red dots), as a result the relay either has a value of -1 or $+1$ (exemplified with the red and green colors respectively). In Figure (b) the value of each relay is computed as the signed area of the cell. The Preisach density is now distributed uniformly in the cell. The red color contributes the the negative area, while the green color contributes to the positive.

the delayed relay operator of the ith cell, defined as

$$y[i](t) = \begin{cases} -1, & \text{for } u(t) > \alpha_i \\ \xi_i, & \text{for } \beta_i < u(t) < \alpha_i \\ 1, & \text{for } u(t) < \beta_i \end{cases} \tag{3.4}$$

which is the discretized version of Equation 2.11, where α_i and β_i refer to the corresponding (α, β) coordinates of cell i. and ξ_i is the state of the ith cell.

3.2 Construction and Implementation of the Preisach Plane

The output of the Preisach model, both discrete and continuous, depend on four variables as shown in Equation 3.1; the Preisach density function μ_i, the area of each cell A_i, the relay value $\mathcal{R}_i[u(t), \xi]$ and the bulk contribution η_0. All these factors is needed in order to calculate the output. This section will explain the general approach to construct the Preisach plane, and show how the area of each hysteron is calculated. The next sections will discuss how the

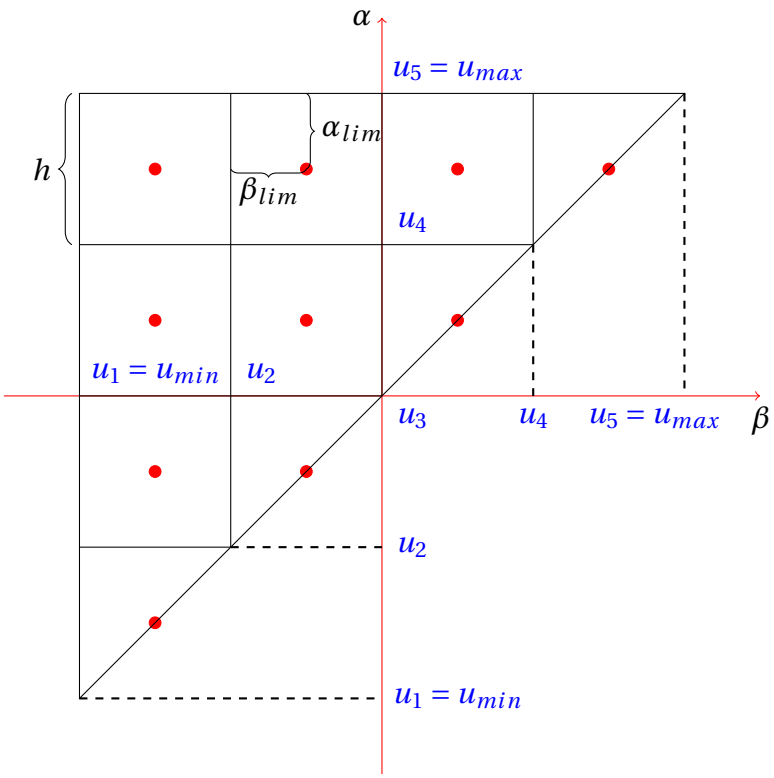


Figure 3.2: How the Preisach plane is built up in the implementation for $n_h = 4$. h is the height/width of each cell, while α_{lim} and β_{lim} refer to the coordinates for switching the output value of the cell on or off, and u_{min} and u_{max} is the input range.

hysteron value $\mathcal{R}_i[u(t), \xi]$ can be computed for the discrete and continuous case, before the implementation of the identification of μ_i and η_0 is discussed.

First, the range and partitioning of the Preisach plane has to be defined. That is the range of the input signal, and how many cells will be used in the construction of the Preisach plane, which is the discretization level n_h in Section 2.3.4. Both these factors can be defined by the user, however, some considerations should be taken into account. The range of the input, $[u_{min}, u_{max}]$, should correspond to the practical input range of the hysteresis operator [Tan and Baras, 2005], which means that inputs outside of this range should not be applied to the identified model, the consequence of this is shown in Chapter 7. In addition Tan and Baras [2005] discuss the discretization level for the constrained least-squares algorithm described in Section 2.3.4. Their results indicate that a consequence of a larger discretization level is a more time-consuming identification procedure, which becomes an important factor if the identification should be run on-line in real-time applications.

When the discretization level and input range is defined, the Preisach plane can be constructed. By assuming a synchronized input range around zero, that is $u_{min} = -u_{max}$, the Preisach plane can be constructed by partitioning the input range in n_h levels, which will re-

sult in $n_q = (n_h(n_h+1))/2$ sub-regions. Then, the coordinate of each center can be computed by a double for loop starting in the bottom left corner of Figure 2.16 and calculate the x and y coordinates of each center. The corresponding α limit can then be found by adding half the distance between the cells to the y coordinate, while subtraction of the same distance from the x coordinate will give the β limit. Figure 3.2 illustrates the construction of the Preisach plane with a discretization level of $n_h = 4$.

When the Preisach plane is constructed, the area of each cell can be calculated. Since the width, h, of each cell easily can be found, the area of cell i can be computed by the following equation

$$A_i = \begin{cases} h^2 & \text{For a square cell} \\ (h^2)/2 & \text{For a triangular cell} \end{cases} \tag{3.5}$$

where the triangular cells are found at the diagonal.

3.3 Output of a Discrete Preisach Model

In this section the implementation of the calculation of the hysteron value $\mathcal{R}_i[u(t), \xi]$ will be discussed for the discrete case, where each hysteron will either have the state $+1$ or -1, depending on both the input size and the last state of the hysteron. As a consequence, the states of all the hysterons have to be remembered from one time step to the next.

The calculation of the next hysteron output can then be conducted with a very simple algorithm

Algorithm 1 Output calculation of delayed relay.

```
1:  if ξ = −1 then
2:      if u(t) > α + (h/2) then
3:          y_i(t) = 1
4:      else
5:          y_i(t) = −1
6:      end if
7:  end if
8:  if ξ = 1 then
9:      if u(t) < β − (h/2) then
10:         y_i(t) = −1
11:     else
12:         y_i(t) = 1
13:     end if
14: end if
```

where h is the height and width of each Preisach hysteron, as illustrated in Figure 3.2, $u(t)$ is the input at time t, $y_i(t)$ is the output of the ith hysteron at time t and ξ is the state of the relay, containing only -1 and 1 values.

This algorithm is then run for each hysteron for each time step t in order to achieve a complete description of how the input affects the system. When this is done, all the hysteron outputs $\mathcal{R}_i[u(t),\xi]$, can be put together in a matrix, where the outputs from each time step is a vector.

The algorithm described above is very time-consuming, since it requires that all hysterons have to be recalculated at each time step. This can be avoided by using a method based on remembering the memory curve, which will be described in the next section for the continuous case. With some modification this method can easily be used for the discrete case as well, since the continuous case calculates the output of the hysterons differently for the memory curve cells.

3.4 Output of a Continuous Preisach Model

The implementation of $\mathcal{R}_i[u(t),\xi]$ in the continuous case is more complicated than in the discrete case. This is because of the need to identify which hysterons the memory curve ψ passes through, and in addition find the signed area of these cells. This method is very briefly mentioned in Tan and Baras [2005], however no algorithm for the calculation is given. A suggestion of an algorithm for the continuous case will therefore be given here.

In this section, the algorithm will be presented, before the ideas of the algorithm are discussed in some more detail. The algorithm assumes that the Preisach plane is constructed as described in Section 3.2, with a discretized grid, and the area of the diagonal elements are half of the square elements.

The proposed procedure is shown in Algorithm 2, where $\boldsymbol{A}$ is a matrix consisting of column vector for each time step containing all areas, while $\boldsymbol{y}$ and $\boldsymbol{\mu}$ are column vectors containing all outputs and Preisach weights respectively. n_q is the number of Preisach elements given by the discretization level $n_q = (n_h(n_h+1))/2$.

The initialization step consists of defining the Preisach plane, while the discretization level and input range, i.e. α_m and β_m, is already given by the density function, however the initial memory curve has to be defined. This can be done, for instance, by defining all

Algorithm 2 Output of piecewise uniform density function

Initialize
for $t = 1$ to *length(t)* **do**
    %*Calculate memory curve based on new input:*
    if increasing input **then**
        New maximum, can delete maximums and minimums
    end if
    if Decreasing input **then**
        New minimum, can delete minimums and maximums
    end if
    %*Calculate ares of elements:*
    for $i = 1$ to $i = q$ **do**
        if memory curve element **then**
            $A_i(t) \in [-1, 1]$
        end if
        if element above memory curve **then**
            $A_i(t) = -1$
        end if
        if element below memory curve **then**
            $A_i(t) = 1$
        end if
    end for
end for
$\boldsymbol{\eta} = \boldsymbol{A}^T \cdot \boldsymbol{\mu}$

in hysterons off or on, corresponding to a memory curve starting at the left or top of the Preisach plane. The next two steps needs to be run for each time step, since a new input will alter the memory curve, which in turn affects the calculated area. For each time step a new input is applied to the model, and the memory curve needs to be updated. If the input is increasing, then it will present itself as a new maximum in the memory curve, however if the input value is larger than any previous maximums, then they will be deleted and the new input replaces it. Moreover, if the input increases past a previous maximum, the minimum value corresponding to the same corner in the memory curve will be deleted as well. The decreasing input case can be treated analogously. From the minimum and maximum vector, the coordinates of each corner of the memory curve can be found as a descending staircase, illustrated in Figure 3.1, where the coordinates refer to the (α, β) values in the Preisach plane.

When the coordinates of the memory curve corners are found, the process of finding the area of each Preisach element can start. The corners of the memory curve is not necessarily in neighboring cells, consequently it is required to find all coordinates which the memory curve passes through. When all the coordinates of the memory curve hysterons are found,

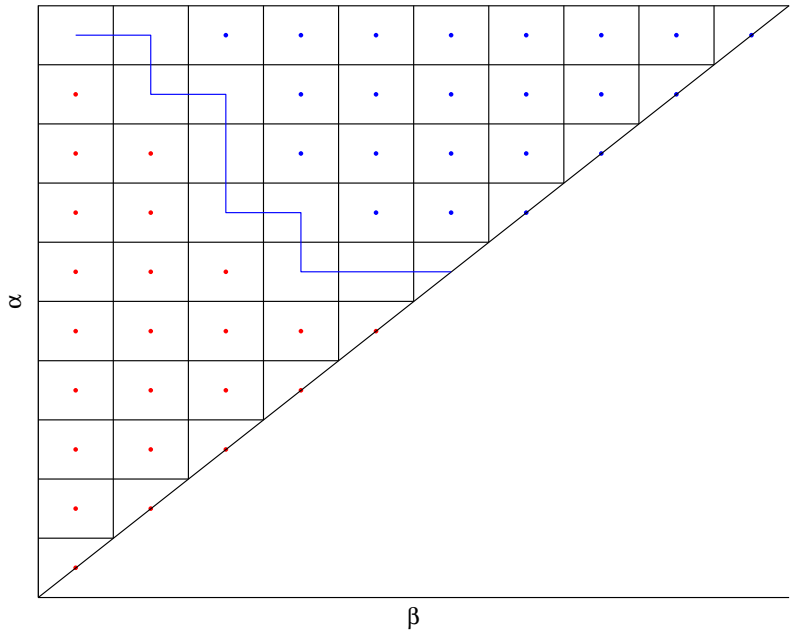


Figure 3.3: Illustration of the partitioning of the Preisach plane. The blue line represent the memory curve, while the red and blue dots corresponds to the relays which is positive and negative respectively.

it is possible to determine which hysterons that is $+1$ and -1, corresponding to hysterons on each side of the memory curve. The Preisach plane at this point is shown in Figure 3.3, where the blue line represents the memory curve, and the red and blue dots are the positive and negative hysterons respectively. The calculation of these values can be done much faster than in the algorithm described in the discrete case, since it is known that all hysterons to the left and below the memory curve are positive. In the same way, the hysterons right and above the memory curve are negative.

Finally, the signed area of the hysterons where the memory curve passes through can be calculated. In each cell, calculate the area either above or below the memory curve, this is simple if the memory curve does not have any corners in the cell. However, when there is corners in the cell, then the calculation of the area has to be done with a more sophisticated method. A suggestion for a general approach is to do $N+1$ calculations in each cell, where N is the number of corners in the cell. For instance, when $N = 2$, as illustrated in Figure 3.4, the calculation of the area can be done in three steps, where the different shades of gray refer to each calculation, as shown in the following equation

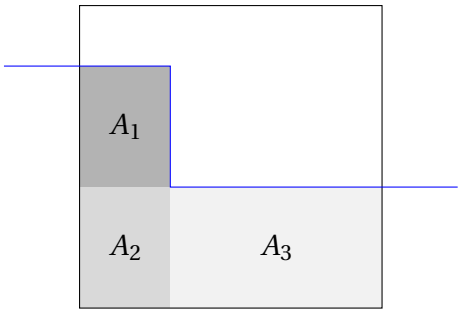


Figure 3.4: Example of how the calculation of S^+ can be done in one hysteron. Since the memory curve has two corners in this cell, the calculation can be done in three steps.

$$A_{pos} = A_1 + A_2 + A_3 \tag{3.6}$$

where A_{pos} is the positive are of the element, and A_1, A_2 and A_3 refer to the areas in Figure 3.3. In order to calculate the negative part, subtract the positive area from the total area of the Preisach cell. This approach will also apply to cells where $N = 0$.

Now, if the signed areas for the memory curve cells is normalized, it is possible to calculate the areas of all the Preisach elements in one step. This means that, for example, the hysterons with value +1 is multiplied with the corresponding area of that cell. The area of all square cells is equal, while the areas of the triangular cells is half the size of the square cells.

The procedure described in this section can also be used for the discrete case, however the area calculation is not needed. Instead, Algorithm 1 can be used for the memory curve hysterons only, since all other hysteron values are known.

The algorithm described in this section, is not necessarily optimal in any sense. However the potential for reducing the run-time of such an algorithm is great compared to the procedure described for the discrete case, where each hysteron had to be recalculated for each time-step.

3.5 Identification of Both Methods

When both the relay value $\mathcal{R}_i[u(t),\xi]$ and area A_i is found, it is possible to identify the Preisach density function μ_i and the bulk contribution η_0 of the system. The procedure for this was discussed in general in Section 2.3.5, however, some points will be repeated here for the sake of completeness. In order to do any identification, it is assumed that data of a hysteresis measurement is available, that is, measurement of the input and output.

The density function μ_i can be found by using the MATLAB function *lsqnonneg*, which requires the weights of the Preisach density function to be positive, that is $\mu_i > 0, \forall i$. Furthermore, the MATLAB function *lsqnonneg* can be used together with *lsqnonlin* in order to find the bulk contribution. In order for the function *lsqnonlin* to work, it needs to be supplied with a function for which it tries to minimize the sum of squares of the output by changing one or more parameters. The output of this function will be the difference between the model and measurement output, therefore this function also need to identify μ_i in order to calculate the output of the model (same equation as Equation 3.1)

$$\eta(t) = \sum_{i=1}^{n_q} \mu_i A_i \mathcal{R}_i[u(t), \xi] + \eta_0 \tag{3.7}$$

where η_0 is the current estimate of the bulk contribution, and μ_i is the best fit when η_0 is applied, derived by the *lsqnonneg* function as described in Section 2.3.5.

When both the Preisach density function $\mu(\alpha, \beta)$ and the bulk contribution η_0 is found, the final model output can be calculated. The identified model depends on the input and output data applied. This is discussed in more detail in Chapter 5.

3.5.1 Piecewise Continuity of the Continuous Model

As mentioned earlier, the continuous model is only piecewise continuous. This is because the Preisach density function $\mu(\alpha, \beta)$ is constant only inside one hysteron. Two neighboring hysterons can have different weights, as a consequence, the slope of the model output can vary when a new hysteron is excited by the input, which again makes a discontinuous point in the output.

A small part of the output from the continuous model is illustrated in Figure 3.5. An example of a discontinuity point can be seen when $t \approx 0.0397$ s, where the blue graph suddenly changes its slope. As mentioned earlier, the Preisach density function consists of both square and triangular hysteron cells. When the input increases in a square cell, the output will change linearly, since the signed area increases linearly as well. In the triangular cells, on the other hand, the signed area increases more rapidly as the input increases, since the shape is triangular, this is shown in Figure 3.6. The effect of this can be seen in Figure 3.5 as a quadratic change of the fitted output. As the input decreases, the negative part of the area grows quicker, and therefore the output will also decrease faster. If all the cells were

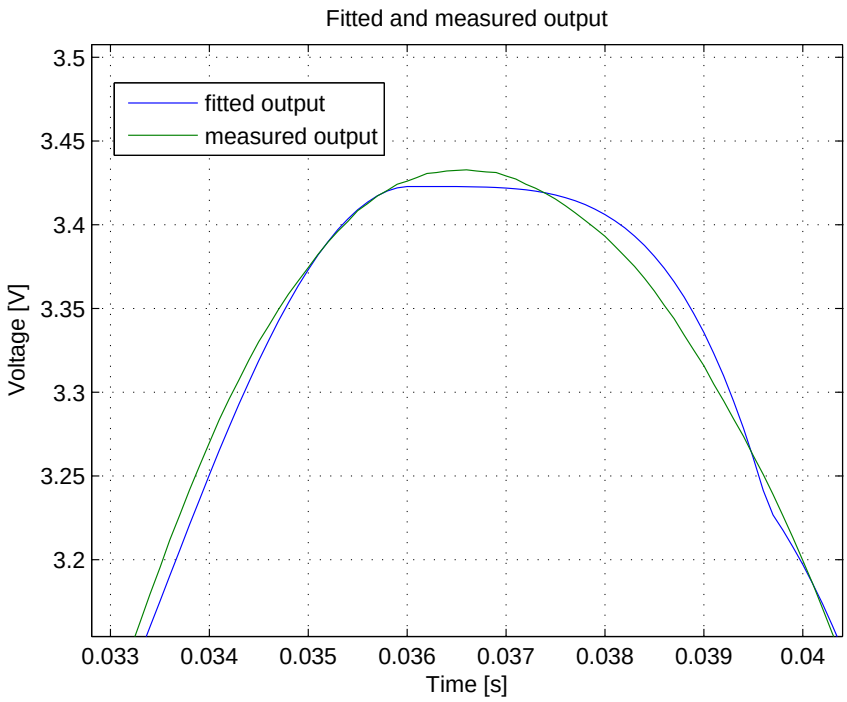


Figure 3.5: The output of a continuous Preisach model compared to the measured output for discretization level $n_h = 20$, zoomed in to the first maximum. The blue graph represents the output of the continuous model, while the green graph is the measured output used in the identification of the model.

Figure 3.6: The change of signed area in square and triangular hysteron cells with an increasing input. While the change in the square cells is linear, the signed area of the triangular cells increases more rapidly.

square, the output would be linear between each time step, and all the output would have been piecewise linear.

Chapter 4

Inversion of the Preisach Model

In order to compensate for hysteresis effect, the models identified in the previous chapter, have to be inverted. In so-called smart materials where hysteresis is common, the system is often modeled as a hysteresis preceding a linear system, as can be seen in Figure 4.1. For instance, in piezoelectric actuators the applied voltage is converted to an electric field inducing polarization, which again induces stress in the piezoelectric material through electro mechanical coupling [Iyer and Tan, 2009]. Hysteresis appear in the conversion between the applied voltage and induced charge [Newcomb and Flinn, 1982].

The study of Tao and Kokotovic [1995] gives an example of a general control structure for systems exhibiting hysteresis, where the inversion compensation is combined with feedback control. Since the system has a preceding hysteretic element, an approximate right-inverse can be constructed to eliminate the hysteresis in the system, illustrated in Figure 4.2, where η_d is the reference output and Γ the hysteresis operator. Then a feedback controller $C(s)$ can be designed in order to control the linear dynamics in $G(s)$ [Iyer and Tan, 2009]. This control method is outlined in Figure 4.3.

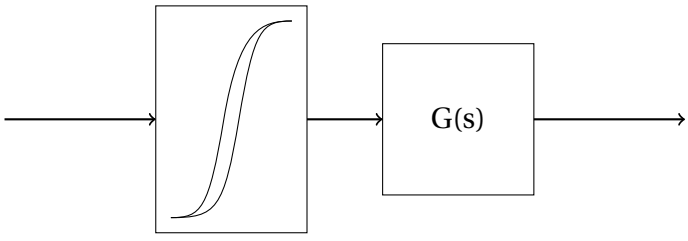


Figure 4.1: System with hysteresis in series with a dynamical system. This is the same figure as Figure 2.5.

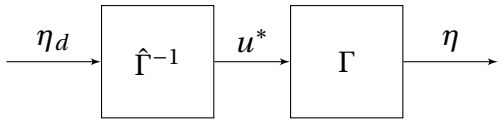


Figure 4.2: Inversion routine for the system. η_d represents the desired output from the system. u^* is the output from the approximate inverse $\hat{\Gamma}^{-1}$ of the Preisach density function Γ corresponding to the input η_d, while η is the output from Γ. The figure is motivated from Iyer and Tan [2009].

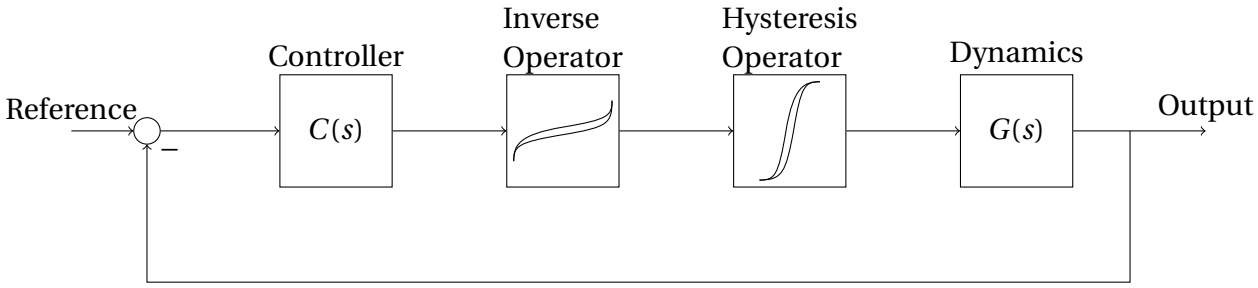


Figure 4.3: The general method for controlling systems with hysteresis. The feedback controller $C(s)$ is designed to handle the linear dynamics $G(s)$ and the inversion error [Iyer and Tan, 2009].

The general approach described above can be used for compensation of several hysteresis operators, for instance the common Preisach-type operators, described in Section 2.2.3. It is worth recalling that only the PI operator can be analytically inverted, while a numerical solution has to be found for both the Preisach- and KP operator. This difference will affect the inversion algorithms for the various operators. In this work, only the Preisach operator is inverted, however the general approach from Figure 4.3 can be used for the other operators as well.

Most of the work on inversion of the Preisach model attempt to construct a right-inverse for the Preisach operator [Tao and Kokotovic, 1995; Tan et al., 2001; Tan and Baras, 2004; Iyer et al., 2005; Iyer and Tan, 2009]. Iyer and Tan [2009] describe the right inverse problem as: given a desired output $\eta_d(t)$, $t \in [0, T]$, and an initial memory curve ψ_0 for a Preisach operator Γ, the inverse operator $\hat{\Gamma}^{-1}$, generates u^* as the input to Γ, such that $\eta(t) = \Gamma[u^*, \psi_0](t) \approx \eta_d(t)$.

4.1 Inversion Algorithms

As described above, one of the most common methods for hysteresis compensation is to construct a right-inverse of the Preisach function. This has been performed with several

methods in the past, depending on, for instance, the type of Preisach operator used. As mentioned in the previous section, the inverse of the PI operator can be found analytically, consequently such an inversion scheme is the most common method for this operator. However, the inversion has also been done numerically, for instance in Tan and Bennani [2008], where the inversion is implemented on a FPGA. The article of Al Janaideh et al. [2011] mentions several studies where inversion algorithms for the PI operator have been used, and the interested reader is advices to investigate this article.

For the Preisach- and KP operator, the inverse have to be calculated numerically, which corresponds to an approximate solution, since no analytically solution exist. One of the first inversion schemes proposed is a table lookup procedure based on calculating the Everett functions [Mayergoyz, 2003]. Models identified with the Everett functions are subject to noise in the measurements, due to two differentiations of the measured signal, during the identification. Consequently, this method is computational expensive and subject to noisy disturbances.

Another proposed method is to identify the inverse Preisach model directly, i.e. switch the input and output of the Preisach model [Natale et al., 2001]. This proposed method result in the so-called pseudo-compensator for hysteresis, due to the fact that the inverse of the Preisach operator is not a Preisach operator [Natale et al., 2001; Tan, 2002]. However, in such a method, the inverse can be found directly from the output of the model, which reduces the complexity compared to the numerical inverse studied below. This method has been used for compensation of hysteresis in both Natale et al. [2001] and Croft et al. [2001].

Iyer and his colleagues have proposed two numerical solutions for the Preisach model, based on the piecewise monotonicity property of the Preisach density function, $\mu(\alpha, \beta)$. This property is also the basis of the continuity and existence properties discussed in the following section. The first method they mention is based on a fixed point iterative scheme, where the convergence of the algorithm relies on the piecewise monotonicity and Lipschitz continuity properties. The details of this model is presented in Iyer et al. [2005] and Iyer and Tan [2009].

The other algorithm presented is the so-called closest match algorithm. In Section 4.3 this algorithm is described in detail for both the discrete and continuous model. The output of this inversion scheme is designed to always be as close as possible to the reference signal. This algorithm is also presented in Iyer et al. [2005] and Iyer and Tan [2009].

4.2 Existence and Continuity of the Inverse Preisach Operator

This section will study some of the conditions that have to be satisfied in order to have both existence and continuity of the inverse Preisach operator. Both these properties will apply to the closest match algorithm. The material in this section is mostly based on Iyer et al. [2005], with some contributions from Visintin [1994] and Brokate and Sprekels [1996].

Iyer et al. [2005] deals with the approximate inversion of the Preisach operator, where they always require that the input $u(t)$ is a Hölder continuous function, which means that the following condition is satisfied. Let $\nu \in (0,1]$ and $T > 0$, then $C^{0,\nu}[0,T]$ is the space of Hölder continuous functions on $[0,T]$, i.e., $\forall\, u \in C^{0,\nu}[0,T]$, $\forall\, t_1, t_2 \in [0,T]$ such that

$$\sup_{0\le t_1,t_2\le T} \frac{|u(t_2)-u(t_1)|}{|t_2-t_1|^{\nu}} \le C_0 \tag{4.1}$$

for some constant C_0. This space is a generalization of the space of Lipschitz continuous functions, with $\nu = 1$ being the Lipschitz condition [Brokate and Sprekels, 1996].

By assuming a nonnegative, integrable Preisach density function, $\mu(\alpha,\beta)$ with compact support S, that is $\mu(\alpha,\beta) \ge 0$ and $\in L^1(S)$, in addition to a continuous input $u(t) \in C[0,T]$. Let the input signal have values only in the interval $I = [u_{min}, u_{max}]$, that is $u(t) \in I, \forall\, t \in [0,T]$. Then define a function $\chi(\cdot)$ on $[0, u_{max} - u_{min}]$

$$\begin{aligned} \chi(x) \triangleq \inf\{&|\Gamma[u,\psi](T) - \Gamma[u,\psi](0)| : \\ &\psi \in \Psi_0, u \in C_m[0,T], |u(T)-u(0)| = x\} \end{aligned} \tag{4.2}$$

where inf is the infimum, i.e. an upper limit.

Iyer et al. [2005] argue that $\chi(\cdot)$ is continuous and monotonically increasing under their assumptions on $\mu(\alpha,\beta)$. Furthermore, it is easy to check that if $\chi(x) > 0$ for all $x > 0$, then the Preisach operator Γ is piecewise strictly increasing (PSI). The PSI condition is defined as

$$(\Gamma[u(t)](T) - \Gamma[u(t)](0))(u(T) - u(0)) \ge 0 \tag{4.3}$$

for a monotone input $u(t) \in C[0,T]$.

Now let J be the smallest interval that contains the output values of Γ when the full range of the input is used, i.e. $u \in I$. Iyer et al. [2005] refer to the proofs in Visintin [1994] and

Brokate and Sprekels [1996] when it is mentioned that if $\chi(x) > 0, \forall x > 0$, then $\Gamma[\cdot,\psi] : C_I[0,T] \rightarrow C_J[0,T]$ is invertible and the inverse operator is also continuous.

Furthermore, Iyer et al. [2005] present two theorems which together proves that the inverse of the Preisach operator is weak-star continuous, which means that it is a continuous signal with respect to the weak-star topology. Detailed information about weak-star topology can be found in Conway [2012], in addition DeVito [1978] discusses the differences between various topologies. These theorems will now be presented in the notation introduced for the (r,s) coordinates in Chapter 2.

Theorem 4.1. *Let $\Gamma[\cdot,\psi]$ be a Preisach operator with domain $I = [u_{min}, u_{max}]$, where $\psi \in \Psi_0$. Assume that the density function $w(r,s)$ has compact support; is integrable; is nonnegative; and $w(r,s) \geq Cr^{\xi}$ for almost every (r,s) in $R_\epsilon = [0,\epsilon] \times [u_{min} - \epsilon, u_{max} + \epsilon]$ where $C > 0$, $\xi \geq 0$, and $\epsilon > 0$. Then*

- $\Gamma[\cdot,\psi] : C_I[0,T] \rightarrow C_J[0,T]$ *is Lipschitz continuous;*
- $\Gamma[\cdot,\psi] : C_I^{0,\nu}[0,T] \rightarrow C_J^{0,\nu}[0,T]$ *is weak-star continuous, where $0 < \nu \leq 1$;*
- $\Gamma[\cdot,\psi] : C_I[0,T] \rightarrow C_J[0,T]$ *is invertible, and its inverse can be extended to a continuous operator $\Gamma^{-1}[\cdot,\psi] : C_J[0,T] \rightarrow C_I[0,T]$.*

Moreover, they show, under the same conditions as in the previous theorem, the existence and continuity of the inverse Preisach operator acting between the spaces of Hölder continuous functions

Theorem 4.2. *Assume that the Preisach density function $w(r,s)$ has compact support, $w \geq 0$, and $w(r,s) \geq Cr^{\xi}$ for almost every $(r,s) \in R_\epsilon = [0,\epsilon] \times [u_{min} - \epsilon, u_{max} + \epsilon]$, where $C > 0$, $\xi \geq 0$, $\epsilon > 0$. Then for any $\psi \in \Psi$, $\Gamma^{-1}[\cdot,\psi]$ is weak-star continuous from $C_J^{0,\nu_2}[0,T]$ to $C_I^{o,\nu_1}[0,T]$, where $\nu_2 \in (0,1]$ and $\nu_1 = (\nu_2/\xi + 2)$.*

Iyer and Tan [2009] summarize these results as follows; if the Preisach density function is greater than zero on a thin strip along the line $\alpha = \beta$ (the diagonal of the Preisach plane), then the Preisach operator $\Gamma[\cdot,\psi] : C_I[0,T] \rightarrow C_J[0,T]$ is invertible with a continuous inverse.

Moreover, Iyer and Tan [2009] outline that with the same conditions on the weights on the diagonal of the Preisach density function, it was shown in Theorem 4.1 and 4.2 that the Preisach operator is a one-to-one mapping from the space of Hölder continuous functions

$C^{0,\lambda}[0,T]$ into $C^{0,\eta}[0,T]$, where $\lambda < \eta$. Then, since $C^{0,\eta}[0,T]$ is a proper subset of $C^{0,\lambda}[0,T]$ and consists of more regular functions than $C^{0,\lambda}[0,T]$, the inverse Preisach operator cannot possess the same smoothing property. Due to this, the inverse of the Preisach operator is not a Preisach operator itself, and therefore the solution of the inverse has to be found numerically, in contrast to the PI operator where the inverse itself is a PI operator [Esbrook et al., 2013; Brokate and Sprekels, 1996].

4.3 Closest Match Algorithm

The closest match algorithm was first published by Tan et al. [2001], and has been modified in a few subsequent studies, Tan [2002] and Iyer et al. [2005], before it appeared in its most recent form in Iyer and Tan [2009]. The idea behind this algorithm is to find the input which give the closest match between the desired output and the model output. The algorithm is described for both discrete and continuous input signals, even though these algorithms look very different, they are both based on the same principle of achieving a closest match output.

4.3.1 Discrete Closest Match Algorithm

The discrete closest match algorithm to be presented here is based on the work of Tan et al. [2001] and Tan [2002]. As described in the previous chapter, the discretization of the Preisach plane means that the input can only take values in a finite set $U \triangleq \{u_l, 1 \leq l \leq L+1\}$ with each $u_l, 1 \leq l \leq L+1$, representing an input level [Tan et al., 2001]. This means that $u_1 = u_{min}$ and $u_{L+1} = u_{max}$, where u_{min} and u_{max} represent the minimum and maximum input value respectively. The distance between each level is described by Δu

$$\Delta u = \frac{u_{max} - u_{min}}{L} \tag{4.4}$$

The inversion problem is formulated as follows: given an initial memory curve $\psi(n)$ at time T_n (from which the initial input $u(n)$ and output $\eta(n)$ can be computed) and a desired output η_d, find $u^* \in U$, such that

$$|\Gamma[u^*, \psi](n) - \eta_d| = \min_{u \in U} |\Gamma[u, \psi](n) - \eta_d| \tag{4.5}$$

where u^* is the optimal input corresponding to the next input $u(n+1)$ and n is the discrete

time step.

The algorithm must also return the resulting memory curve ψ created from the inputs $u(t)$ used in the inversion, since this memory curve describes the current state of the Preisach plane.

The idea of the algorithm is described by Tan et al. [2001] as follows. Consider the case where the current output $\eta(n)$ is less then the next desired output $\eta_d(n+1)$ (the case $\eta(n) > \eta_d(n+1)$ is treated in exactly the same way with some obvious modification). By increasing the input iteratively with one level until, at iteration k, the input $u^{(k)}$ reaches u_{max}, or the output $\eta^{(k)}$ corresponding to $u^{(k)}$ exceeds $\eta_d(n+1)$. For the first case, the optimal output will be u_{max}; for the second case there are two candidates for the optimal input u^*, $u^{(k-1)}$ or $u^{(k)}$. The optimal input is then found by choosing the input which corresponds to the least output error. The memory curve of the model must be backed up whenever the input is increased, to retrieve the correct memory curve at all times.

The algorithm from Tan et al. [2001] is described in detail in Algorithm 3. This algorithm will find the optimal input in at most L iterations, and each iteration can be computed very fast, since only a small amount of hysterons have changed state, and only these states needs to be updated.

The algorithm only finds the next optimal input, however Tan [2002] describes a method based on dynamic programming for finding the optimal inputs for a sequence of desired outputs. He states that finding each optimal input, one at a time, is a sub-optimal solution. This method will not be described here, and the interested reader is advised to study the work of Tan for this approach.

Interpolation of the Discrete Method

The inversion algorithm presented above will result in a discrete input signal, not suited for direct application on for instance piezoelectric actuators. To avoid this, a linear interpolation method between the computed values is proposed by Tan et al. [2001]. This method ensures a continuity of C^0, instead of having a discontinuous point for each step in the input function.

This is further studied in Chapter 7 with three different methods for achieving the query points for a linear interpolation. A query point refers to a point used as an start and/or end point for a linear segment in the interpolation. Given a set of query points, the linear inter-

Algorithm 3 Closest Match Algorithm - Discrete method

Initialization
Set $k = 0$
if $\eta^{(0)} = \eta_d(n+1)$ **then**
    $u^* = u^{(0)}$; $\psi(n+1) = \psi^{(0)}$; **Exit Algorithm**
end if
while *Not found the closest match* **do**
    if $\eta^{(0)} < \eta_d(n+1)$ **then**
        if $u^{(k)} \geq u_{max}$ **then**
            $u^* = u_{max}$; $\psi(n+1) = \psi^{(k)}$; **Exit Algorithm**
        else
            $u^{(k+1)} = u^{(k)} + \Delta u$; $k = k+1$; $\bar{\psi} = \psi^{(k)}$ %*Backing up the memory curve*
        end if
        Evaluate $\eta^{(k)} = \Gamma(u^{(k)}, \psi^{(k-1)})$ % *At the same time update memory curve to* $\psi^{(k)}$
        if $\eta^{(k)} = \eta_d(n+1)$ **then**
            $u^* = u^{(k)}$; $\psi(n+1) = \psi^{(k)}$; **Exit Algorithm**
        end if
        if $\eta^{(k)} < \eta_d(n+1)$ **then**
            Return to start of while loop
        else
            if $|\eta^{(k)} - \eta_d(n+1)| \leq |\eta^{(k-1)} - \eta_d(n+1)|$ **then**
                $u^* = u^{(k)}$; $\psi(n+1) = \psi^{(k)}$; **Exit Algorithm**
            else
                $u^* = u^{(k-1)}$; $\psi(n+1) = \bar{\psi}$; **Exit Algorithm**
            end if
        end if
    else if $\eta^{(0)} > \eta_d(n+1)$ **then**
        if $u^{(k)} \leq u_{min}$ **then**
            $u^* = u_{min}$; $\psi(n+1) = \psi^{(k)}$; **Exit Algorithm**
        else
            $u^{(k+1)} = u^{(k)} - \Delta u$; $k = k+1$; $\bar{\psi} = \psi^{(k)}$ %*Backing up the memory curve*
        end if
        Evaluate $\eta^{(k)} = \Gamma(u^{(k)}, \psi^{(k-1)})$ % *At the same time update memory curve to* $\psi^{(k)}$
        if $\eta^{(k)} = \eta_d(n+1)$ **then**
            $u^* = u^{(k)}$; $\psi(n+1) = \psi^{(k)}$; **Exit Algorithm**
        end if
        if $\eta^{(k)} > \eta_d(n+1)$ **then**
            Return to start of while loop
        else
            if $|\eta^{(k)} - \eta_d(n+1)| \leq |\eta^{(k-1)} - \eta_d(n+1)|$ **then**
                $u^* = u^{(k)}$; $\psi(n+1) = \psi^{(k)}$; **Exit Algorithm**
            else
                $u^* = u^{(k-1)}$; $\psi(n+1) = \bar{\psi}$; **Exit Algorithm**
            end if
        end if
    end if
end while

polation will consist of linear lines between each of these points.

4.3.2 Continuous Closest Match Algorithm

The continuous closest match algorithm was first described in Tan [2002], but has been slightly modified since then. This section is based directly on the article of Iyer and Tan [2009], which presents the modified version of the algorithm.

This algorithm assumes, in contrast to the discrete case, that the weights of the density function $\mu(\alpha,\beta)$ is uniformly distributed in each discrete cell, but the value can vary from cell to cell. In the discrete case the weights were assumed to be a point mass in the center of each cell. With the piecewise monotonicity assumption it possible to invert the model with arbitrarily accuracy.

The inversion problem is formulated as almost equal to the discrete case, but will be repeated here for completeness: Given a memory curve ψ_0, find u^* (the next optimal output $\eta(n+1)$, for a discrete time n), such that the hysteresis output is equal or close to the desired output $\eta_d(n+1)$ for the next time instant, that is, $\eta_d(n+1) = \Gamma[u,\psi(n)](n+1)$.

The Preisach operator is assumed to be piecewise monotone, and the input and output of the Preisach operator corresponding to the memory curve $\psi(n)$ are $u(n)$ and $\eta(n)$ respectively, as in the discrete case. Furthermore, suppose that $\eta_d(n+1) > \eta(n)$ (the case $\eta_d(n+1) < \eta(n)$ can be treated analogously with a few modifications).

With the assumptions described above, the basic idea of the algorithm can be explained as follows. When the change in input Δ_u is small enough, the output is related to Δ_u through the quadratic function

$$\Delta_\eta = a_2\Delta_u^2 + a_1\Delta_u \tag{4.6}$$

where the coefficients a_1 and a_2 depend on the current memory curve. The algorithm uses the same idea as in the discrete case, where the input is changed iteratively, but now the change is the smallest among three possibilities; d_0, d_1 or d_2. d_1 represents the difference between the input value in the current iteration, and the next node with a greater value in the input partition, d_2 is the smallest value of Δ_u that eliminates the next corner of the memory curve, and d_0 is the solution Δ_η to Equation 4.6 with Δ_u given by the difference between $u_d(n+1)$ and the output value in the current iteration.

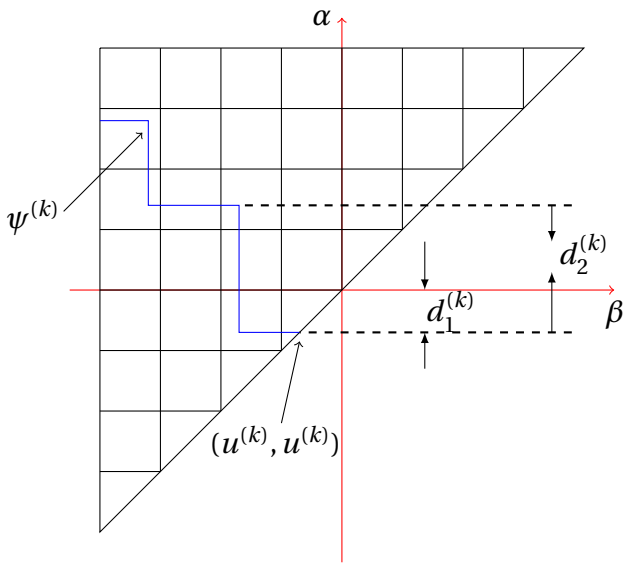


Figure 4.4: An illustration of the discretization grid of the Preisach model with the parameters $d_1^{(k)}$ and $d_2^{(k)}$ used in the continuous inversion algorithm. The level of the discretization is chosen to $L = 8$. d_1 represents the distance between the input value on the current iteration and the next input level on the discretization grid at iteration k. d_2 represents the change of input required in order to remove the next corner in the memory curve. The illustration is based on a figure from Iyer and Tan [2009].

The closest match algorithm for the continuous case is presented in Algorithm 4. A proposed method for finding $a_1^{(k)}$, $a_2^{(k)} \geq 0$ is to define, for instance $a_1^{(k)} = 1$, and solve for $a_2^{(k)}$. Check if the resulting $a_2^{(k)}$ is greater than zero, if this is not true, decrease $a_1^{(k)}$ and check again. Do this until both $a_1^{(k)}$ and $a_2^{(k)}$ are positive.

A illustration for calculation of the $d_1^{(k)}$ and $d_2^{(k)}$ values can be seen in Figure 4.4. As mentioned in the algorithm, $d_1^{(k)}$ is the distance from the end of the current memory curve to the next discrete input level, while $d_2^{(k)}$ is the minimum value which will eliminate the next corner in the memory curve. Note that the sign before $d_1^{(k)}$ and $d_2^{(k)}$ is opposite in the two cases in Algorithm 4 corresponds to an increasing and decreasing input. The choice of the d value such that $0 < d < \min\{d_1^{(k)}, d_2^{(k)}\}$, is studied in Chapter 7.

Algorithm 4 Closest Match Algorithm - Continuous method

Initialization $u^{(0)} = u_0$, $\eta^{(0)} = \eta_0$, $\psi^{(0)} = \psi(n)$, $k = 0$;
while *Not found closest match* **do**
  if $\eta^{(0)} < \eta_d(n+1)$ **then**
    Calculate:
    $d^{(k)} = \min\{d_0^{(k)}, d_1^{(k)}, d_2^{(k)}\}$
    $u^{(k+1)} = v^{(k)} + d^{(k)}$
    $\eta^{(k+1)} = \Gamma[v^{(k+1)}, \psi^{(k)}]$
    %*Where* $d_0^{(k)}$, $d_1^{(k)}$, $d_2^{(k)}$ *are determined in the following way*
    Let $d_1^{(k)} > 0$, be such that $v^{(k)} + d_1^{(k)}$ equals the next discrete input level.
    Let $d_2^{(k)} > 0$ be the minimum value such that applying $v^{(k)} + d_2^{(k)}$ eliminates the next corner of $\psi^{(k)}$.
    % *Compute* $a_1^{(k)}$, $a_2^{(k)} \geq 0$ *satisfying, for* $0 < d < \min\{d_1^{(k)}, d_2^{(k)}\}$
    $\Gamma[v^{(k)} + d, \psi^{(k)}] - \Gamma[v^{(k)}, \psi^{(k)}] = a_2^{(k)} d^2 + a_1^{(k)} d$
    % *Solve for* $d_0^{(k)}$
    $\eta_d(n+1) - \Gamma[v^{(k)}, \psi^{(k)}] = a_2^{(k)} (d_0^{(k)})^2 + a_1^{(k)} d_0^{(k)}$
  else if $y^{(0)} > y_d(n+1)$ **then**
    Calculate:
    $d^{(k)} = \min\{d_0^{(k)}, d_1^{(k)}, d_2^{(k)}\}$
    $u^{(k+1)} = v^{(k)} + d^{(k)}$
    $\eta^{(k+1)} = \Gamma[v^{(k+1)}, \psi^{(k)}]$
    %*Where* $d_0^{(k)}$, $d_1^{(k)}$, $d_2^{(k)}$ *are determined in the following way*
    Let $d_1^{(k)} > 0$, be such that $v^{(k)} - d_1^{(k)}$ equals the next discrete input level.
    Let $d_2^{(k)} > 0$ be the minimum value such that applying $v^{(k)} - d_2^{(k)}$ eliminates the next corner of $\psi^{(k)}$.
    % *Compute* $a_1^{(k)}$, $a_2^{(k)} \geq 0$ *satisfying, for* $0 < d < \min\{d_1^{(k)}, d_2^{(k)}\}$
    $\Gamma[v^{(k)} + d, \psi^{(k)}] - \Gamma[v^{(k)}, \psi^{(k)}] = a_2^{(k)} d^2 + a_1^{(k)} d$
    % *Solve for* $d_0^{(k)}$
    $\Gamma[v^{(k)}, \psi^{(k)}] - \eta_d(n+1) = -a_2^{(k)} (d_0^{(k)})^2 - a_1^{(k)} d_0^{(k)}$
  end if
  if $d^{(k)} = d_0^{(k)}$ **then**
    $u = v^{(k)} + d$
    Exit Algorithm
  else
    $k = k + 1$ % *Continue the iterations*
  end if
end while

Chapter 5

Experimental Design for Hysteresis Identification

Experimental design is the study of maximizing the amount of information from an experiment. There are several factors that affect the information provided by the experiment. Goodwin and Payne [1977] refer to a dynamical system as an example, where experimental design includes choice of input and measurement ports, test signals, sampling instants, and presampling filters. In this chapter, some of the theory behind experimental design will be presented, including a more detailed description of the motivation of experimental design, an example of a general experiment and what factors are relevant in designing experiments for hysteresis identification. The last point will receive the most focus, as it lays the foundation for the design of the experiments conducted in this thesis.

5.1 Motivation

As mentioned above, the aim of experimental design is to maximize the information of from experiment. This is of course an important effect to consider in laboratory work, since a well designed experiment can reduce the time consumption of an experiment, however, it can be even more useful in industrial applications where economy is a vital factor. Spall [2010] exemplifies this point when commenting on the inefficiency of just changing one factor at a time. He criticizes the inventor and entrepreneur James Dyson for his claim that only one factor can be changed at a time in a prototype, since, when making several changes at a time, there is no easy way to know what factor that improved the object or which did

not. Spall quickly contradicts this statement, and at the same time introduces the concept of factorial design, which is a method of changing several factors, but still be able to find which factors made the system improve the most. He also states: *Furthermore, this learning can be done more efficiently and with the additional benefit of acquiring information about input-variable interactions that cannot be revealed in the one-at-a-time changes advocated by Dyson.* Where the learning he refers to is the idea of factorial design.

In addition to making an experiment less time consuming and not as economically expensive, a well designed experiment will also give more insight of the system. This, in turn, will give the experimenters more information of how the system can be improved in the future.

As mentioned in Atkinson et al. [2007], an well designed experiment will also make it possible to describe the system with a good model. On the other hand, such an experiment can also test the adequacy of a predefined model. By maximizing the information given in an experiment, the resulting model will be more precise than it would have been if the information from the experiment was not maximized. A challenge in many experiments is that the measurements often includes unknown noise. Atkinson et al. [2007] argue that statistical methods are essential for both the design and analysis of an experiment. This coincides with the factorial design methodology described by Spall [2010].

5.2 A General Experiment

There are several variables or factors that affect the amount of information from an experiment. Atkinson et al. [2007] sort these factors into three groups; explanatory variables, concomitant variables and disturbances or errors. A figure of a general experiment, with the three groups of inputs is presented in Figure 5.1.

Before an experiment can be conducted, the system has to be studied in order to identify which factors influence the results. Furthermore, all the variables should be sorted together in the correct group. Errors cannot be controlled and can be challenging to identify, however an effort should be done in order to find errors that influence the output significantly, since the results can be interpreted with respect to these errors. Atkinson et al. [2007] use a medical experiment in order to exemplify the explanatory and concomitant variables. For instance, when a new drug is tested, a test group is chosen in order to study the effects of the drug. The

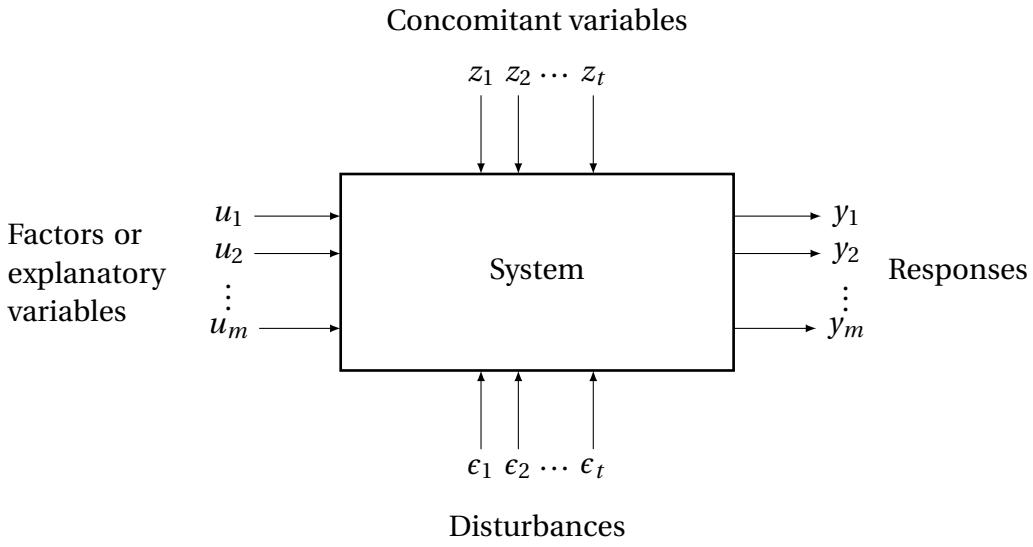


Figure 5.1: Schematic representation of an experiment on a system [Atkinson et al., 2007].

explanatory variables can be controlled directly, for instance the concentration of the drug, or the test frequency of the group. The concomitant variables, on the other hand, can only be controlled indirectly, for example through the choice of test group. These variables can, in the medical example, for instance be the weight or age of the peoples participating in the experiment.

The explanatory variables described by Atkinson et al. [2007], can be split into two different groups, namely qualitative and quantitative factors, as described by Spall [2010]. This classification can also be used for the concomitant variables, however the use is maybe more common in the explanatory variable case. According to Spall [2010], quantitative factors often vary continuously, while qualitative factors, on the other hand, are often assigned a discrete scale. Consequently, quantitative factors are generally associated with a numerical scale, while qualitative factors are associated with a labeling scheme or, the absence or presence of some characteristics. In the medical example from Atkinson et al. [2007], a quantitative factor can be the amount of drug the participants get, in contrast, a qualitative factor, on the other hand, can for instance describe if a specific participant is a part of a placebo test group or not.

Theoretically, the values of the input factors can be chosen without any restrictions, however, a physical system often has constraints which needs to be taken into account. These constraints can, for instance be a result of the equipment used in the experiment, or restrictions on the system itself. Going back to the medical experiment, a restriction can be the maximum amount of medicine a participant should get. In a mechanical system such re-

strictions can, for instance, be the maximum force used on the system. Such constraints will be discussed in more detail in the next section.

5.3 Factors for a Well Designed Experiment

As mentioned in the introduction to this chapter, Goodwin and Payne [1977] refer to experimental design as the choice of input and measurement ports, test signals, sampling instants, and presampling filters. Each of these factors can have a significant impact on the information provided by an experiment. In this thesis, the design of the input signal will be given the most attention, however, the influence from the other factors are mentioned for the general case, in addition to some references to their influence on identification of hysteresis. Moreover, in Chapter 7, the effect of low pass filters are studied in context of hysteresis identification.

5.3.1 A Measure of Information

In order to compare different experimental factors used in various experiments, it is necessary to have a measure that gives some indication of how informative a specific experiment is. Goodwin and Payne [1977] refer to this as a measure of the "goodness" of an experiment, where a logical approach is to use a measure related to the accuracy of the parameter estimates which will be obtained from the data collected. In other words, from the experiment one obtains a number of parameters, which can be compared with a statistical method.

In identification and estimation, there is an unknown vector of model parameters, $\boldsymbol{\theta}$. If $\hat{\boldsymbol{\theta}}_n$ is the estimate of $\boldsymbol{\theta}$ after n independent measurements, and $\hat{\boldsymbol{\theta}}_n$ is assumed to be an unbiased estimator, then the estimator is said to achieve the Cramer-Rao lower bound [Spall, 2010]

$$\mathrm{Cov}(\hat{\boldsymbol{\theta}}_n) \geq \boldsymbol{F}_n(\boldsymbol{\theta}_{true})^{-1} \tag{5.1}$$

which means that the covariance of the estimated model parameters is larger or equal to the inverse of the Fisher information matrix, given as [Spall, 2010]

$$\boldsymbol{F}_n(\boldsymbol{\theta}) \equiv E\left(\frac{\partial \log p(Y|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \frac{\partial \log p(Y|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T}\right) = -E\left(\frac{\partial^2 \log p(Y|\boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T}\right) \tag{5.2}$$

where $p(Y|\boldsymbol{\theta})$ is the statistical mass function of some data and is assumed known. The proof of the Cramer-Rao lower bound, for an unbiased estimator, can be found in Goodwin and Payne [1977].

In addition, if an estimator is efficient, which means that the parameter covariance matrix is equal to the inverse of the Fisher information matrix, i.e. [Goodwin and Payne, 1977]

$$\text{Cov}(\hat{\boldsymbol{\theta}}_n) = \boldsymbol{F}_n(\boldsymbol{\theta}_{true})^{-1} \tag{5.3}$$

the variance can be found directly from the Fisher information matrix. This means, that in order to minimize the variance of the covariance, which will increase the amount of information, the Fisher information matrix has to be maximized. This will result in a lower valued covariance matrix, and therefore give a more well designed experiment.

In Section 5.4 the design of the input signal is described in detail, with focus on the criteria of a persistence excited (PE) input signal. In addition, Goodwin and Payne [1977] mention other designs that affects the information from an experiment, for instance, they explore the possibility of having nonuniform sampling intervals. Based on how the model of the system is, there can be obtained more information by sampling with two or more frequencies.

5.3.2 Constraints

As mentioned in the previous section, experiments have to be conducted within the restrictions of the system. These constraints cannot be broken, even if this, in theory, would give more information about the system. Goodwin and Payne [1977] mention several typical constraints that are met in practice:

- Amplitude constraints on input, outputs or internal variables.
- Power constraints on input, outputs, or internal variables.
- Total time available for the experiment.
- Total number of samples that can be taken or analyzed.
- Maximum sampling rate.
- Availability of transducers and filters.
- Availability of hardware and software, for analysis.

All or none of these constraints can be important, depending on the type of experiment. Since it can be difficult to determine which constraints that have to be taken into account, one approach is to use a subset of the constraints while checking for violation of the other constrains. Often the most critical constraints can be monitored, while constraints which are not that vital are checked occasionally. Critical constraints can be the maximum voltage used on the experimental setup, while a less critical constraint is, for instance, the number of samples taken in an experiment, since this probably does not destroy any equipment.

5.4 Design of Input Signal

The aim of the experiments performed in this thesis is to study what types of input signals which gives the most information about the system. For on-line identification this subject is frequently discussed in the context of a PE input signal. Such a signal ensures that the estimated parameters converges to the real parameters of the system [Ioannou and Sun, 2003].

In this section the PE property will be discussed, both in general and for systems exhibiting hysteresis. Even though the identification algorithms used in this thesis is on a closed form, i.e. they are off-line, the estimated parameters should be closer to the real values when the input signal satisfies the PE property. This is because the data from the experiment should include more information.

5.4.1 Persistence of Excitation

As remarked in the introduction to this section, a PE input signal will ensure that an estimated set of parameters converge to the real values in an on-line identification. In order to achieve this, the input has to give information about the system, and as Ioannou and Sun [2003] mention, a constant input $u = 0$ will not excite the system at all, and therefore the output does not carry any information. Consequently, the input has to change, in order for the output to give information.

Ioannou and Sun [2003] define the PE property as

$$\alpha_1 I \leq \frac{1}{T_0}\int_t^{t+T_0} \phi(\tau)\phi^T(\tau)d\tau \leq \alpha_0 I, \quad \forall\, t \leq 0 \tag{5.4}$$

where ϕ is PE in $\mathcal{R}^n$ with level of excitation α_0 if there exists constants for α_1, $T_0 > 0$. ϕ refer to the vector of inputs.

Ioannou and Sun [2003] explain that: *Although the matrix $\phi(\tau)\phi(\tau)^T$ is singular for each τ, (4.3.39)* (that is Equation 5.4) *requires that ϕ varies in such a way with time that the integral of the matrix $\phi(\tau)\phi(\tau)^T$ is uniformly positive definite over any time interval $[t, t+T_0]$.*

5.4.2 Persistence of Excitation for Systems with Hysteresis

In linear systems, the input is required to have a certain amount of information in order to satisfy the PE property. By contrast, hysteresis is a rate independent nonlinearity, which means that the behavior of the system will not be altered by changing the frequency of the input. Consequently, using various frequencies in the identification would not, by itself, give any more information about behavior of the system.

As a result of this rate independent property, the input has to be designed based on some other criteria to satisfy the PE property, namely the magnitude of the input. This difference is also summarized in the quote from Iyer and Shirley [2004]

> *Note that, as the Preisach operator is rate-independent, one does not need to test the prediction of the Preisach operator for varying input frequencies - only the input magnitude needs to be varied.*

The reason for this magnitude variation is that each Preisach element in the discretization grid has to be excited to see the effect each element has on the output. If a Preisach model of $n_h = 4$ is excited with an input bounded inside $u \in [u_2, u_4]$, then the only parts of the Preisach plane identified would have been the blue area in Figure 5.2. Hence, all the Preisach weights μ_i representing the white elements in the figure would not be found, and assumed zero. As a result, the input needs to excite all the elements in the Preisach plane in order to get a complete description of all the elements.

However, in order to achieve a PE input for identification of the Preisach model, the input has to satisfy some other conditions, besides exciting the entire Preisach plane. Tan and Baras [2005] discuss and analyze the PE property for a system exhibiting hysteresis, and their results were summarized in the following two theorems

Theorem 5.1. (Necessary condition for PE): *If $V[n]$ is PE, then there exists $N > 0$, such that for any n_0, for any $i \in \{1, 2, ..., L\}$, $u[n]$ achieves a local maximum at u_{i+1} or a local minimum at u_i during the time period $n_0, n_0 + N - 1]$.*

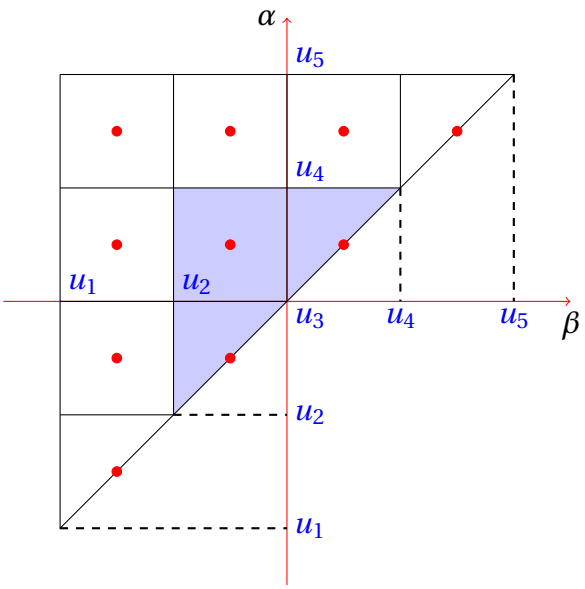


Figure 5.2: The Preisach plane with discretization level $n_h = 4$. The blue area represents which Preisach elements that are excited if the input only varies inside $u \in [u_2, u_4]$.

where $V[n] \equiv W[n] - W[n-1]$, and W is a vector of the states ξ of each Preisach element.

Theorem 5.2. (Sufficient condition for PE): *If there exists $N > 0$, such that for any n_0, one can find $\{u'[n]\}_{n=n_0}^{n_0+N-1} \in \{\underline{u[n]}\}_{n=n_0}^{n_0+N-1}$ satisfying the following: There exist time indices $n_0 \le n_a \le n_1^- < n_1^+ < n_2^- < n_2^+ < \cdots < n_i^- < n_i^+ < \cdots \le n_b \le n_0+N-1$ or $n_0 \le n_a \le n_1^+ < n_1^- < n_2^+ < n_2^- < \cdots < n_i^+ < n_i^- < \cdots \le n_b \le n_0+N-1$, such that $u'[n_i^+]$ is a local maximum and $u'[n_i^-]$ is a local minimum of $\{u'[n]\}_{n=n_a}^{n_b}$ for each i, these local maxima and minima include all input levels $u_i, 1 \le i \le L+1$, and either*

a) $\{u'[n_i^+]\}$ is nonincreasing, $u'[n_i^+] \ge u'[n]$ for $n_i^+ < n \le n_b$, $u'[n_i^+]$ differs from $u'[n_{i+1}^+]$ by no more than Δ_u, and $u'[n_i^-]$ is nondecreasing, $u'[n_i^-] \le u'[n]$ for $n_i^- < n \le n_b$, $u'[n_i^-]$ differs from $u'[n_{i+1}^-]$ by no more than Δ_u; or

b) $u'[n_i^+]$ is nondecreasing, $u'[n_i^+] \ge u'[n]$ for $n_i^+ < n \ge n_a$, $u'[n_i^+]$ differs from $u'[n_{i+1}^+]$ by no more than Δ_u, and $u'[n_i^-]$ is nonincreasing, $u'[n_i^-] \le u'[n]$ for $n_i^- > n \ge n_b$, $u'[n_i^-]$ differs from $u'[n_{i+1}^-]$ by no more than Δ_u

In other words, the theorems above describe that the input has to have enough reversals, i.e. the input need to change direction. This is logical, since the hysteresis effect arise when the input changes direction, where the output seems to lag behind. Furthermore, from the results of Tan and Baras [2005], the input satisfies the PE condition if, at each input level, i.e. $u_1 \cdots u_n$, where the discretization level is $n_h = n$, the input has a reversal. Hence, all Preisach

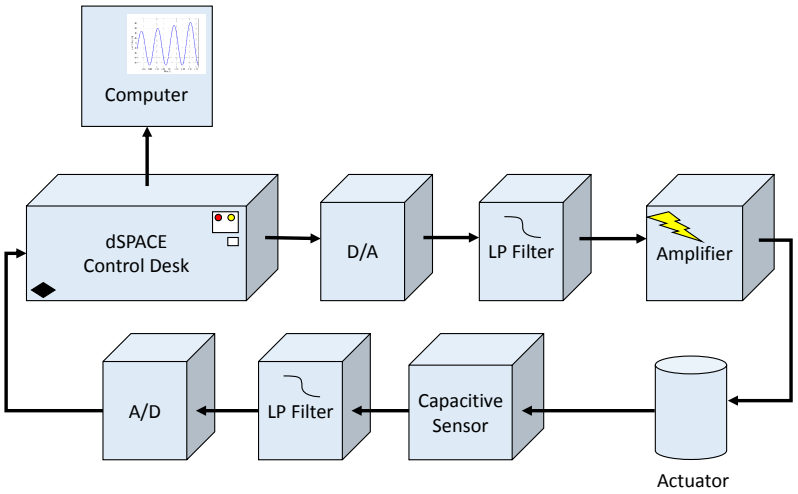


Figure 5.3: Experimental Setup.

elements is excited with an input reversal, which gives information about the hysteresis behavior at each subregion of the Preisach plane.

The theorems above were derived for the discrete model, however, with a few modifications the results can be extended to the continuous case. The interested reader is advised to consult Tan and Baras [2005] for more information.

5.5 Experimental Setup

The experimental setup used in this thesis is described in the work of Eielsen [2012], and the following description is from his work, with a few modifications of the equipment

The experimental set-up consists of a dSPACE DS1103 hardware-in-the-loop (HIL) system, an ADE 6810 capacitive gauge, an ADE 6501 capacitive probe from ADE Technologies, a Piezodrive PDL200 voltage amplifier, two SIM-965 programmable filters from Stanford Research Systems, and the custom-made long-range serial-kinematic nanopositioner shown in Figure 5.4. The nanopositioner is fitted with a Noliac SCMAP07-H10 actuator. The capacitive probe has a bandwidth of 100 kHz, and the voltage amplifier with the capacitive load of the actuator, has a bandwidth in excess of 100 kHz. The capacitive measurement has a sensitivity of $1/5$ V/μm and the voltage amplifier has a gain of 20 V/V. With the DS1103 board, a sampling frequency of $f_s = 100$ and $f_s = 10$ kHz is used for all the experiments. An illustration of this experimental setup is shown in Figure 5.3.

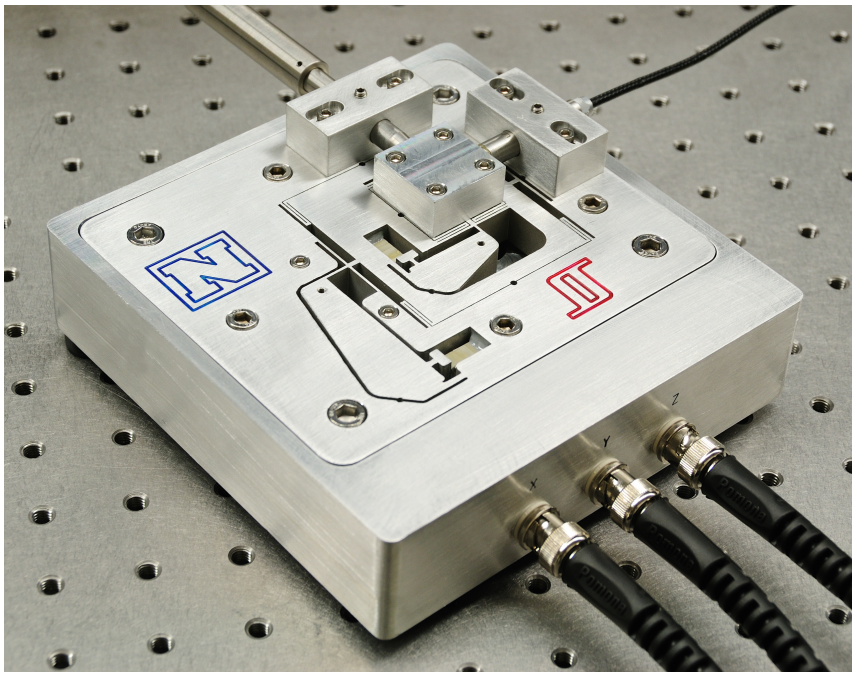


Figure 5.4: Custom made nanopositioning device, [Eielsen, 2012].

The two filters were designed as low pass filters with cutoff frequency at 25 kHz, except when performing experiments of the effect of low pass filtering on identification, then the cutoff frequency of the last low pass filter was set to 500 Hz. When time delay was introduced in the measurement, a *transport delay* block was added in simulink, delaying the signal with a defined time delay τ.

5.6 A Well Designed Hysteresis Experiment

A well designed experiment is, as discussed earlier, an experiment where the information is maximized, with respect to the physical limitations of the system. In a general experiment, the "goodness" of an experiment could be measured with the Fisher information matrix, however, for a system exhibiting hysteresis, the PE property of the input signal are a more relevant measure. This is due to the fact that in experiments from systems with hysteresis, there is only one input, while the Fisher information matrix is often used in order to distinguish the effects of several changes at a time [Spall, 2010], where a low covariance responds to a high Fisher information matrix, and the measure of "goodness" would have been the largest possible information matrix.

Consequently, the design of input signals for the identification gives the most information about the system. Therefore the PE class of signals discussed in Tan and Baras [2005]

were used. In addition, several other signals, described in Section 5.6.2 were used in order to study the identification results when the input is not PE. Before these input signals are shown, the physical constraints of the system will be discussed, since the input signals must not violate these constraints. Moreover, Leang and Fleming [2009] argue that the input signal significantly influences the overall performance of high speed nanopostioning applications.

5.6.1 Constraints

The constraints of the system are mainly due to the piezoelectric actuator, but the voltage amplifier introduces some limitations to the system as well. The constraints in this section are grouped as frequency and voltage constraints.

Frequency

Even though, as mentioned in Section 5.4, the amount of frequencies does not give any additional information about the hysteresis, the system has limitations based on the frequencies, due to creep and vibrations. While the creep effect is significant for low frequency input signals, the effect becomes less as the input frequency increases. This creep effect is illustrated in Devasia et al. [2007]. Vibrations, on the other hand, gives an upper limit for the input frequency. Croft et al. [2001] argues that, for a piezoelectric actuator, the vibration effect restricts the input frequency to be 10-100 times lower than the first resonance frequency of the system.

As a result of creep and vibrations, the input should not have frequencies that are too high or low. From Figure 5.5, it can be seen that the resonance frequency of the piezoelectric actuator is about 800 Hz, which gives, based on Croft et al. [2001], that the input frequency should at least be lower than 80 Hz. The resonance frequency is due to the actuator dynamics of the system.

These effects can be compensated for with models for creep and vibrations, however, this has not been done in this thesis. The reader is advised to study Croft et al. [2001] for a detailed explanation of models for creep and vibration, and compensation routines for them.

Voltage Constraints

The voltage amplifier cannot produce a higher voltage than 200 V, which means that if a voltage larger than 10 V is sent in to this amplifier, the output will be limited to 200 V, since

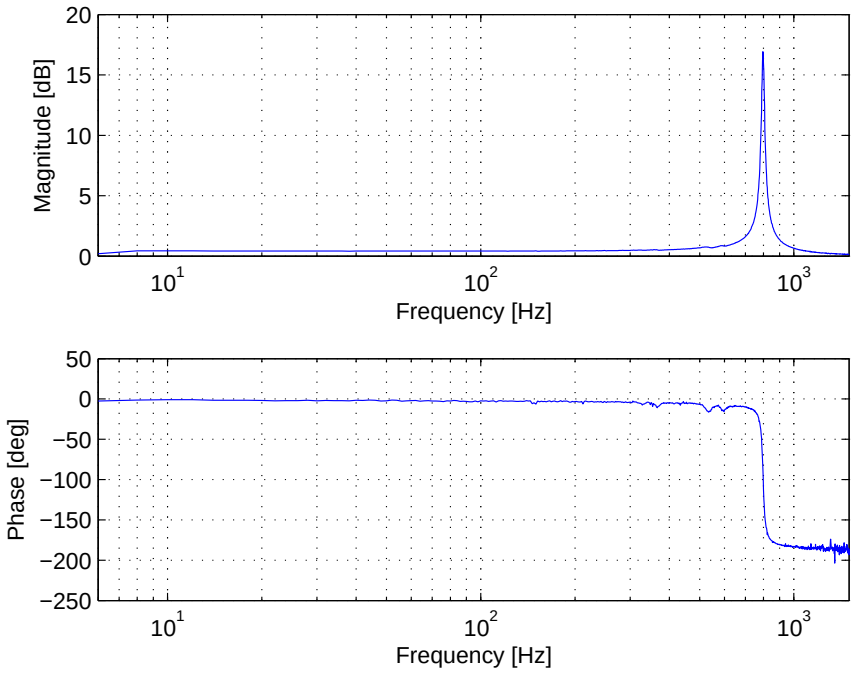


Figure 5.5: Measured frequency response of the piezoelectric actuator.

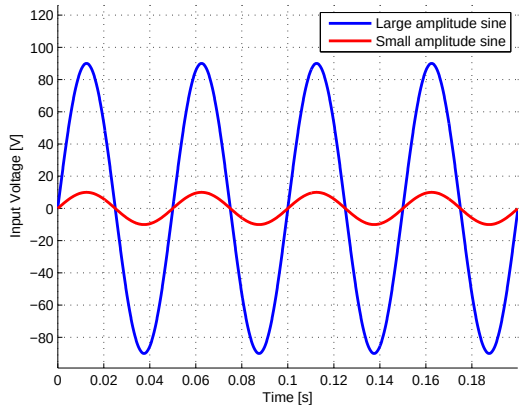
the input voltage is amplified with 20 V/V. If this happens, the input and output will not correspond, therefore the results will be invalid.

In addition, there is a lower limit on the input voltage as well. If the input is sufficiently negative, the piezoelectric actuator can be depolarized, which in practice means that the actuator is broken. However, this does not need to be considered in this experimental setup, since the voltage amplifier has a lower limit of -10 V on the output.

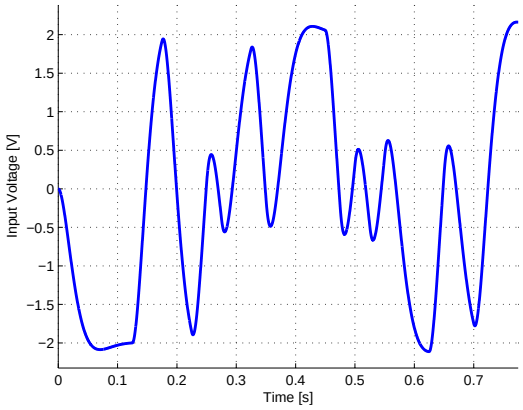
5.6.2 Input Signals for Identification

Several input signals have been used in the identification of the hysteresis effect in this thesis. These signals are shown in Figure 5.6, where Figure 5.6a shows two sine waves with amplitudes 10 and 90, while Figure 5.6b is a Pseudo random binary noise (PRBN) signal. The last two signals, in Figure 5.6c and 5.6d refer to signals with at least one reversal at each Preisach level, however the first signal is built up with a sine wave, while the second is a triangular wave.

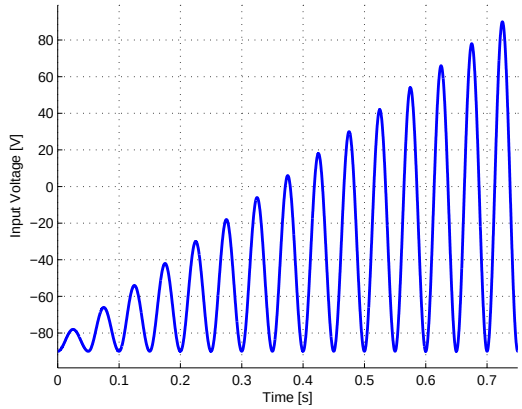
The small sine wave in Figure 5.6a give an example of what happens if only a small part of the Preisach plane is excited, while the large-amplitude sine wave in Figure 5.6a will give an indication of how the output responds when the entire Preisach plane is excited, but with



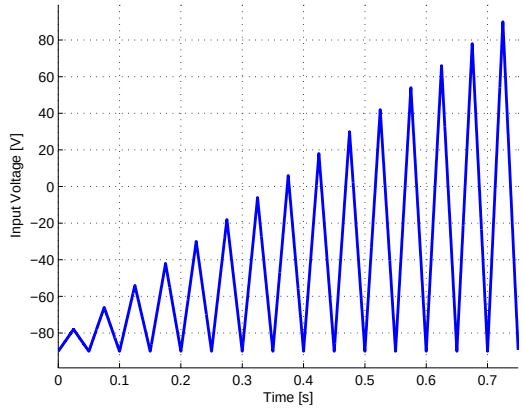
(a) Small amplitude sine wave.



(b) Big amplitude sine wave.



(c) Input described as PE in Tan and Baras [2005].

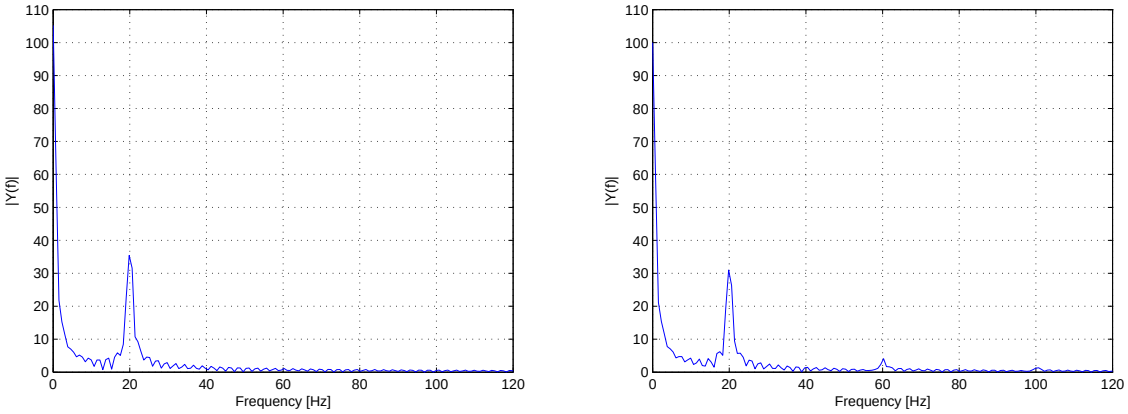


(d) A filtered pseudo random binary noise signal.

Figure 5.6: Input signals for identification.

only reversals at the same amplitude. The two PE input signals, seen in Figure 5.6c and Figure 5.6d represents two similar signals, however, the sine wave does only excite one frequency in the system, while the triangular wave will experience several maximums in the frequency response. This can be observed in Figure 5.7, where the frequency response of both the sine and triangular signals are shown.

The PRBN signal in Figure 5.6b is a signal in between the single amplitude sine signal and the reversing input function, where several reversals are conducted, but not in the same systematical manner as in Figure 5.6c and 5.6d.



(a) Frequency response of Figure 5.6c.

(b) Frequency response of Figure 5.6d.

Figure 5.7: The frequency response of an increasing sine signal and an increasing triangular signal used in the identification. The reason for the large amplitude at zero frequency, is that the signal has a DC component when it starts at -90 V. If studied closely, the frequency response of the triangular wave have maximums at 60 Hz and 100 Hz, due to the fundamental at 20 Hz.

Chapter 6

Systematic Errors in the Signal Loop

The Preisach model only describes hysteresis, which is defined as rate independent, and does not take into account frequency dependent effects, such as time delay, low pass filtering, creep and actuator dynamics. However, due to the instrumentation of the signal loop, these effects can influence the measurements of the system. Consequently, the identified models based on data where these effects are present, will be colored by, for instance, low pass filtering and time delay. As a result, the Preisach weight function will try to include these effects in the model, however, due to the frequency dependence of these effects, the model cannot encompass various frequencies of the desired input.

In this chapter, factors that lowers the phase of the system will be discussed, these effects mainly include low pass filtering, time delay and creep. Moreover, the origin of these effects in the experimental setup will be discussed with a few examples. In addition, the actuator has a resonance frequency at about 800Hz and a negative phase around the same frequency, as shown in the Chapter 5, which may influence the measurements of the system, and will be discussed briefly in this chapter.

6.1 Introduction

When performing experiments on nanopositioning actuators, such as a piezoelectric actuator, there are several instruments used in processing of the signal. The experimental setup was described in detail in Section 5.5, in this section, however, a more general approach will be given for the setup. The instruments in the signal loop will be discussed, with respect to error sources that influencing hysteresis identification.

In a general experiment with a piezoelectric actuator, based on the work done in this thesis, the signal is designed and implemented on a real-time computer, in this case the dSPACE Control Desk. Further, the signal is sent through a digital to analog converter, a low pass filter and a voltage amplifer before a voltage is applied to the piezoelectric actuator. The displacement of the actuator is measured with a capacitive sensor, which provides a voltage output corresponding to the displacement. This voltage signal is passed through another low pass filter, before being converted to a digital signal for the dSPACE Control Desk. The first low pass filter is often named a reconstruction filter, the second filter, on the other hand, is called an anti aliasing filter [Eielsen, 2012].

Each of the instruments mentioned affect the measurement of the hysteresis. In the ideal case, neither the input or output signals will be affected by the instrumentation, and in this case the measurement should only include the response of the system. Unfortunately, this is not the case, for instance, a low pass filter will introduce a negative phase in the system, the actuator dynamics can be excited, and time delays occurs in sensors, due to converting energy from one form to another. Each of these errors will be introduced in the following sections.

6.2 Effect of Negative Phase

As described in Balchen et al. [2004], an effect of negative phase in a system means that the output is phase shifted compared to the input. This phase shift is problematic for identification of hysteresis, since it will give a hysteresis-like effect in the input-output relationship, illustrated in Figure 6.2, where the input-output relationship of the input $u(t)$

$$u = 10\sin(2\pi f t) \tag{6.1}$$

with $f = 0.1$ as the chosen frequency of the sine-wave, is passed through a low pass filter $G(s)$

$$G(s) = \frac{1}{s+1}. \tag{6.2}$$

The resulting input-output relationship between the input $u(t)$ and the filtered output $y(t) = G(s)u(t)$, can be seen in Figure 6.2, where the output is clearly phase-shifted compared to the input. This phase shift, due to negative phase, is also shown in the bode diagram of

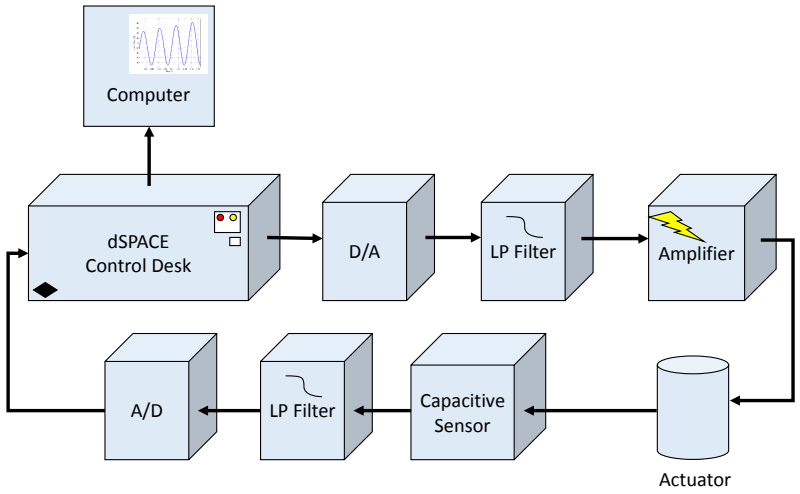


Figure 6.1: Experimental Setup. The same figure as Figure 5.3 in the previous chapter, however it is included once more for completion.

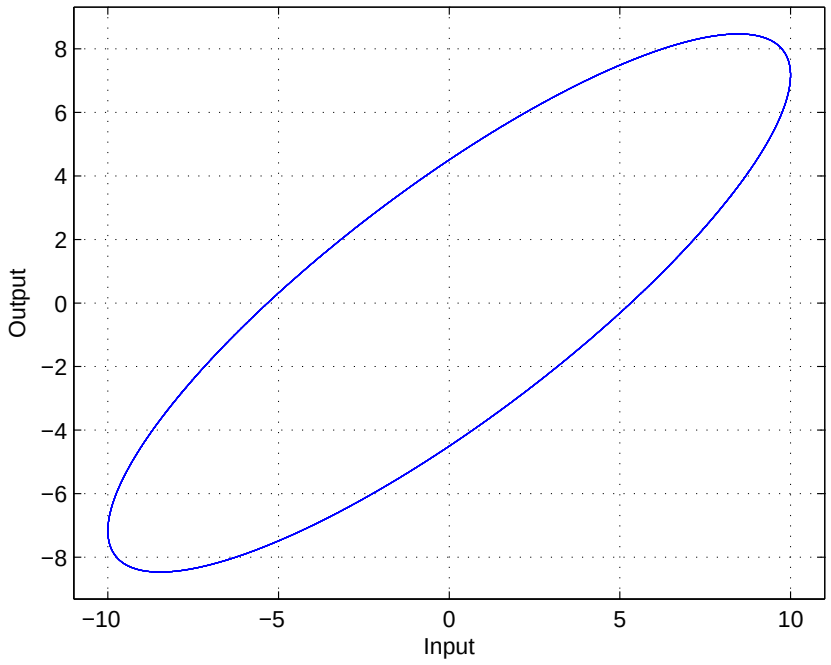


Figure 6.2: Input-output relationship of low pass filtered signal.

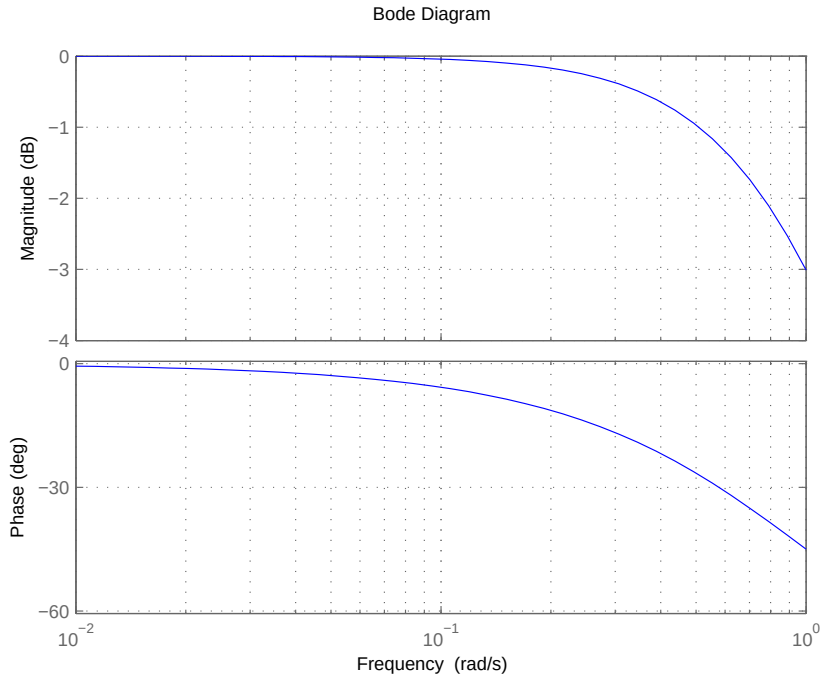


Figure 6.3: Bode diagram of the low pass filter in Equation 6.2.

the low pass filter in Equation 6.2, illustrated in Figure 6.3, where the phase is negative for an input of 0.1 Hz. As a result, the output is phase shifted compared to the input, which again resembles, for instance, the input-output relationship from a system with hysteresis, as shown in Figure 2.4.

Due to this, if negative phase is present in systems with hysteresis, then some part of the observed output lag originates from the negative phase, in addition to the effect from the hysteresis. If the aim is to identify hysteresis, then these negative phase effects should be compensated. This topic will be described in more detail in the following sections.

6.3 Dynamics

The dynamics is how the piezoelectric actuator responds to various input frequencies. The frequency response of the actuator was shown in Figure 5.5 in the previous chapter. This dynamic results in vibrations with a large magnitude resonance peak, limiting the frequencies used in the input [Ragazzon, 2013].

From this dynamic it can also be seen that for inputs with a larger frequency than 800Hz, the output will be out of phase compared to the input, due to the negative phase. In addition, even small amplitude frequencies around the resonance frequency will be greatly amplified,

Table 6.1: Table of Negative Phase in the piezoelectric actuator.

Frequency	Negative Phase [Degrees]
10Hz	-1
100Hz	-2
200Hz	-3
300Hz	-4

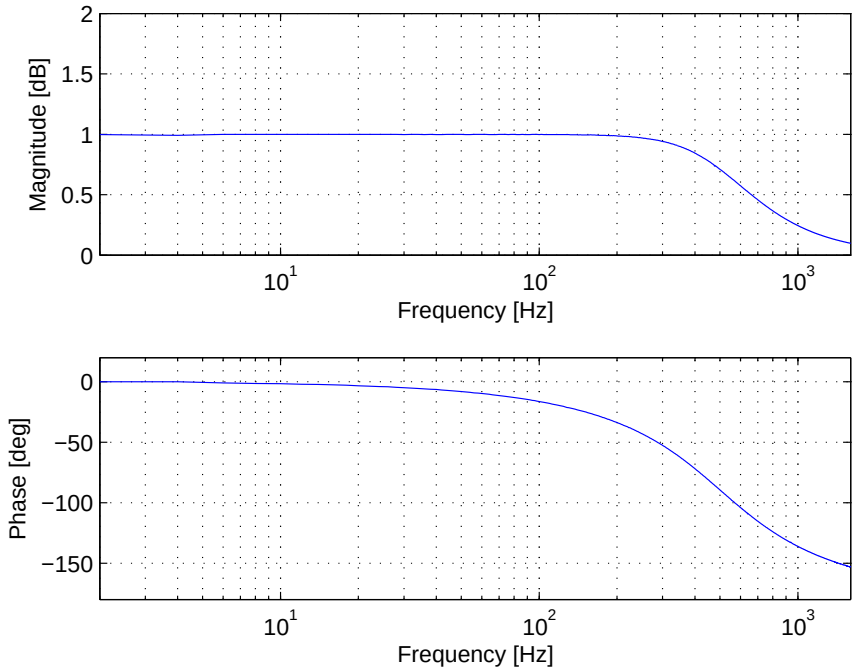


Figure 6.4: Measured frequency response of a low pass filter with cutoff frequency $f_{cutoff} = 500$ Hz.

and cause vibrations.

However, if the input signals can be limited from these frequencies, the negative phase of the dynamics is not that significant for low frequency inputs, and will not phase shift the output considerably, as shown in Table 6.1. For frequencies up to 100 Hz, the phase shift is limited to two degrees, and, as seen in Figure 5.5, the negative phase does not increase considerably before the input frequency surpass 700 Hz.

6.4 Low Pass Filtering

Another source of negative phase is a low pass filter, which is used for both reconstruction and anti-aliasing in the experimental setup in Figure 6.1. The significance of this instrument

Table 6.2: Table of Negative Phase in Low Pass Filter with Cutoff Frequency 500Hz

Frequency	Negative Phase [Degrees]
10Hz	-1.5
100Hz	-16
200Hz	-33
300Hz	-53

on the negative phase of a system greatly depends on the cutoff frequency of the filter. In Figure 6.4, a frequency response of a low pass filter with a cutoff frequency of 500Hz is shown. Compared to the negative phase of the dynamics, the low pass filter shown in Figure 6.4 has a significantly greater influence on the negative phase on low frequencies. This is summarized in Table 6.2, where the negative phase of the low pass filter is -16 degrees, compared to -2 degrees in the dynamics. On the other hand, since the low pass filter is designed by the user, a higher cutoff frequency can be chosen in order to avoid this. As a result, the effect of the negative phase of the frequency can be neglected if the cutoff frequency is chosen correctly.

However, in order to avoid the vibrations due to the resonance in the dynamics, it can be desirable to low pass filter the input signal with a cutoff frequency lower than 800 Hz. In this case, the negative phase of the low pass filter must be considered in the identification of hysteresis, since the measured output will be a product of the low pass filter and the hysteresis effect.

6.4.1 Frequency Dependence Effect on Hysteresis Identification

As seen in Figure 6.4 and Table 6.2, the negative phase of the low pass filter varies greatly with frequency, in contrast to hysteresis which is, as discussed earlier, rate-independent. If a hysteresis model is identified in a case where the output is phase shifted by both a low pass filter and the physical hysteresis, the model tries to encompass both effects. As a result, an inverted input signal calculated from such a model will most likely not give satisfactory results for other input frequencies than those applied in the identification. This is due to the frequency dependency of the negative phase of the low pass filter. Therefore, care should be taken in the identification of hysteresis when low pass filters is included in the experimental setup.

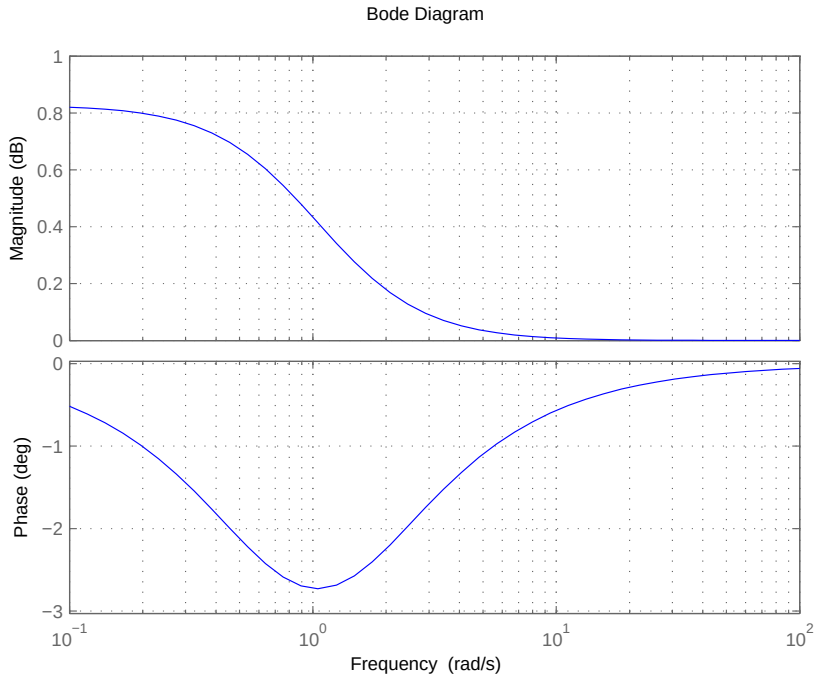


Figure 6.5: Bode diagram of the model in Equation 6.3.

6.5 Creep

Creep is an undesirable property, common in piezoelectric actuators, which can result in significant loss of precision in nanopositioning applications [Devasia et al., 2007]. As mentioned in Chapter 2, creep can be compensated for by a model. Devasia et al. [2007] propose a linear model for such compensation. A simplified description of creep, based on this model is

$$G(s) = 1 + \frac{1}{s+1} \tag{6.3}$$

To illustrate the negative phase from creep, the bode diagram of Equation 6.3 is plotted in Figure 6.5. From this illustration it can clearly be seen that creep has a negative phase for low frequencies, while this effect withdraws as the frequency increases. As a result, this effect is only visible for low frequency inputs.

By sending a step through the piezoelectric actuator the creep can be visualized as a slow creeping effect close to the final value. With the use of this data, the simplified creep model of the system can be approximated as

$$G(s) = \frac{s+1.039}{s+1} \tag{6.4}$$

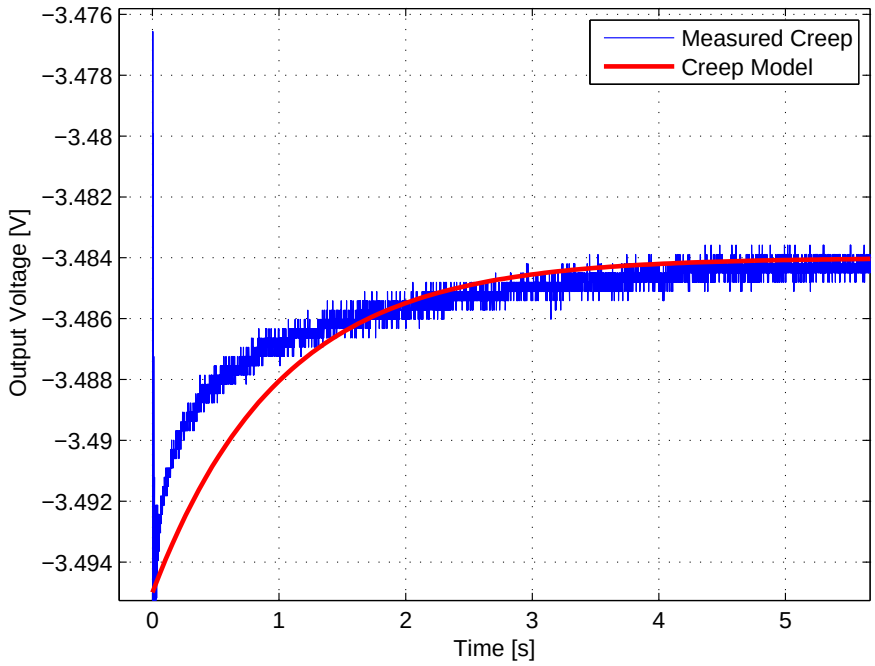


Figure 6.6: Measured creep of the piezoelectric actuator and identified creep model.

The step response of the model in Equation 6.4 is shown together with the measured creep in Figure 6.6, where the red line is the approximated model based on Equation 6.3. From the figure it can be noticed that after about 4 seconds the creep is almost removed from the system. The approximated model was only calculated as an experiment, and was not used in the further work of this thesis.

6.6 Time Delay

A consequence of a time delay, $G(s)$, is that the output is delayed compared to the input, described by Balchen [1990] as a linear reduction in the phase

$$\angle G(s) = \omega\tau \tag{6.5}$$

where ω is the frequency and τ is the time delay in seconds. Time delay has been reported in nanopositioning applications, for instance in Yong et al. [2013] and Ragazzon et al. [2014]. Ragazzon et al. [2014] reports a time delay of $\tau = 4.58e^{-4}$ s in a commercially available AFM, and the resulting bode diagram for such a time delay is shown in Figure 6.7. This is summarized in Table 6.3, where the negative phase can be seen to decrease more from 10-100Hz

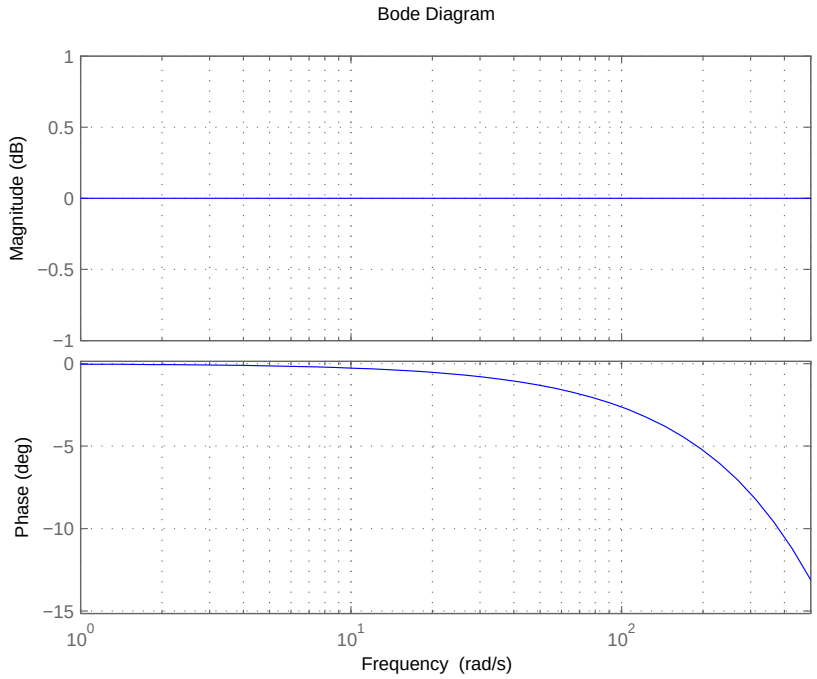


Figure 6.7: Bode diagram of time delay $\tau = 4.58e^{04}$s.

Table 6.3: Table of Negative Phase with a time delay of $\tau = 4.58e^{04}$, based on Figure 6.7.

Frequency	Negative Phase [Degrees]
10Hz	-0.27
100Hz	-2.7
200Hz	-5.3
300Hz	-8.2

than the dynamics, but less than the low pass filter with cutoff frequency at 500Hz.

Ragazzon [2013] speculates that the source of the time delay can be the displacement sensor in the AFM application. On a general basis several types of sensors experience time delay, for instance passive capacitive sensors, optical sensors, inductive sensors and sensors based on modulation. All these sensors convert energy from one form, for instance capacitance, to another, for example voltage. In such a conversion, there will always be a short time delay.

Chapter 7

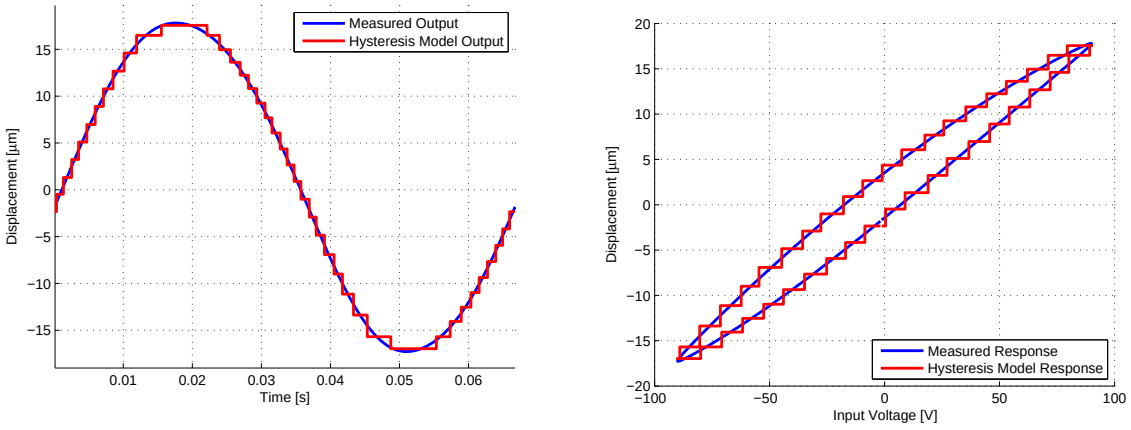
Results & Discussion

In this chapter the results from each of the previous chapters will be presented. Moreover, since most of the results are based on previous results, each topic includes discusses.

The results will be presented in the same order as the previous chapters, first, the discrete and continuous models are presented, where the differences and similarities will be discussed. Secondly, the inverted signals of both these methods are shown, where tests have been conducted in order to verify the inverted signals. Furthermore, three methods for choosing the query point in linear interpolation is studied for the discrete inverse method. Thirdly, the experimental design will be discussed, where the aim is to decide which input signals gives most information for identification. Moreover, the choice of input signal will be seen in the context of the theory presented in Chapter 5. Finally, the effect of negative phase on identification is studied, where signals with various frequencies are sent through a low pass filter with a cutoff frequency which affects the input frequencies. In addition, a time delay is introduced in the system, based on the findings of Ragazzon et al. [2014], for studying the consequences this introduces for identification of hysteresis.

7.1 Discrete and Continuous Preisach Model

In this section the difference between the discrete and continuous models are presented. Plots of the identified models are shown, and the output of each model is compared, where both similarities and differences are discussed. First, some examples of the models are shown, before the properties of the input range for the models are presented.



(a) Measured output and hysteresis model output.

(b) Input-output relationship.

Figure 7.1: Model output when identifying using a 20 Hz sine wave, with 90 V amplitude, with $n_h = 20$ as discretization level.

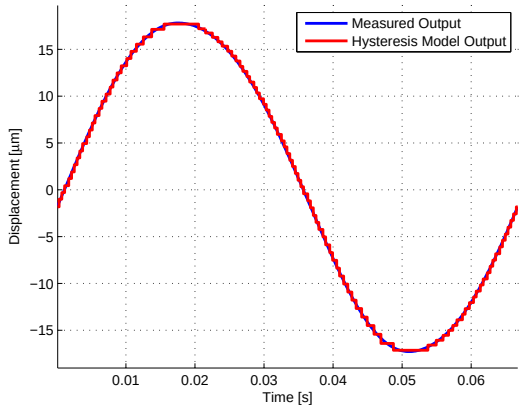
7.1.1 Discrete Model

The discrete nature of the discrete model implies, as argued in Chapter 3, that the output has a finite number of possible values, illustrated in Figure 7.1, where a discrete model is found based on measurement of the input and output from a piezoelectric actuator.

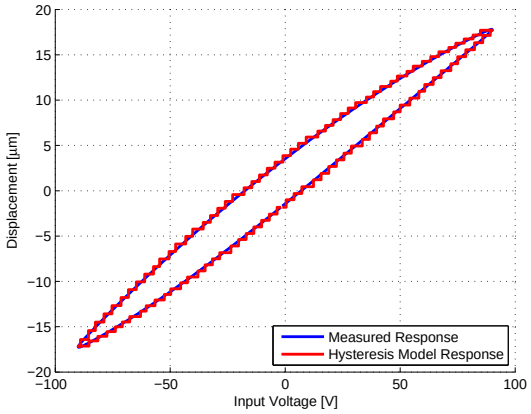
By studying Figure 7.1, the output resembles a staircase signal, which corresponds to, for an increasing input, switching relay values, $y_i(t)$ from -1 to 1. Due to the nonnegative nature of the Preisach weight function $\mu(\alpha,\beta)$, a monotonically increasing input will result in an increasing output. In fact, due to this nonnegative property the output steps will always move in the same direction as the input.

An example of this nonnegative property can be seen in Figure 7.1a where the output has 21 unique values, for the discretization level $n_h = 20$. Theoretically, the discrete output can have $(n_h(n_h+1))/2$ different output values, which in this case corresponds to 210 unique values, i.e. 10 times as many as in Figure 7.1a. However, this requires an input value which has input reversals for all (α,β) values, such that each unique output value can be produced from the discrete model. Furthermore, all Preisach elements in the model must have a nonzero weight, since a zero-valued weight will not change the input even if the corresponding relay is switched from -1 to 1 or opposite. In the results in Figure 7.1 and 7.2, the input is an sine wave with 90 V amplitude, consequently the input is only reversed at the extremal points.

Another effect of the discrete output nature is that there will be an error between the measured and modeled output. This can be seen in Figure 7.1, where the discretization level

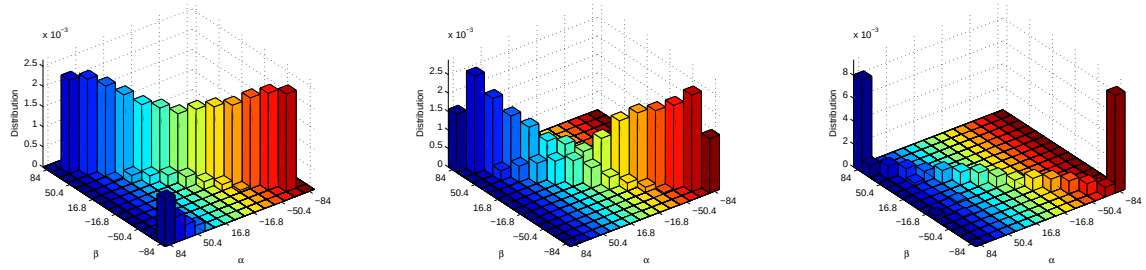


(a) Measured output and hysteresis model output.



(b) Input-output relationship.

Figure 7.2: Model output when identifying using a 20 Hz sine wave, with 90 V amplitude, with $n_h = 50$ as discretization level for the discrete model.



(a) Identified with an input range of [-84,84].

(b) Identified with an input range of [-91,91].

(c) Identified with an input range of [-150,150].

Figure 7.3: The identified Preisach weight function, when using the discrete model, for three different input ranges, while the model is defined for an input in the range $[-90, 90]$.

of the model is $n_h = 20$. In order to limit this error, a higher discretization level on the model can be used for description, illustrated in Figure 7.2, where the model has a discretization level of $n_h = 50$. This clearly improves the fit between the measured- and model output. Concerns related to a higher discretization level is that the model complexity will increase, which in turn will make both identification and inversion more time consuming. However, this is the price to pay in order to achieve an accurate discrete model.

When identifying the Preisach model, there are several important considerations to remember in order to identify a hysteresis model. For instance, input range, low pass filters and time delay. The effect of low pass filters and time delay is thoroughly studied in Section 7.4, while the effect of input range will be discussed in this section. When an identification of the Preisach plane is conducted, an input range is defined, which gives the corresponding α

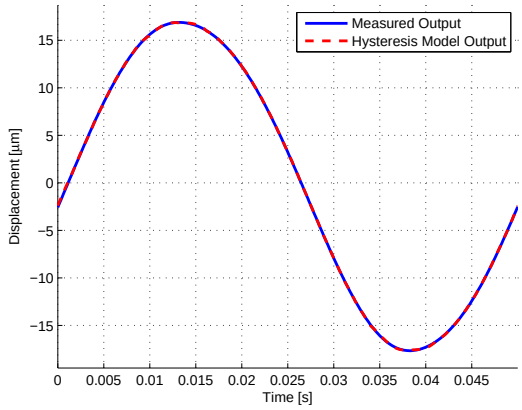
and β values. The effect of a too small or large input amplitude, compared to the input range of the model will now be studied.

In Figure 7.3 the Preisach weights for a model defined in the range $[-90, 90]$ is identified for three different input ranges. in Figure 7.3a, the input range is smaller than the model range, while in Figure 7.3b, the input corresponds to the input range and in Figure 7.3c the input range exceeds the model range substantially.

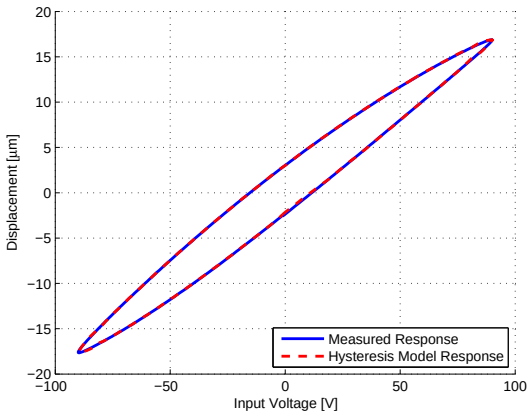
When the input range is too small, not all Preisach weights will be excited. This results in a Preisach weight function which has zero-valued elements on the relays corresponding to these (α, β) values. Since there are no change in these Preisach elements, the identification method does not receive any information about these relays. Consequently, in Figure 7.3a, the edge elements are zero. However, in the corner in the Preisach weight function, there are some nonzero weights, even though the input does not excite these elements. The three smallest of these elements are defined to be -1 initially, as all other elements were as well, but will change to +1 once the input increases beyond its α value. However, a decreasing input cannot change them back to -1, since their corresponding β value is less than the minimum input, this can be due to a creep effect. The corner element, on the other hand, will never change its value, and the contribution from this element will therefore always be negative. If the model had been identified with an input signal with a larger range, then this constant contribution would have been described by η_0 in Equation 2.42. In this situation however, some of this constant contribution are found in the η_0 variable, while the rest is described by the corner weight in Figure 7.3a. Both the corner weight and η_0 represent a constant contribution in this case.

In contrast to the Preisach weights identified with a too small range, the Figure 7.3b describes a situation where the input exceeds the model range by a small value. This is, in fact, necessary for identification of the discrete model, since the input is required to be larger than the largest α value, in this case 90, in order for this element to change value. When comparing to the previous case, it can be seen that the corner element is now zero, which means that a possibly constant contribution is described by the η_0 variable.

The last case is when the input range exceeds the identification range with a large value. In this case, the identified Preisach weights are shown in Figure 7.3c. Here the elements on the end-diagonal are considerably larger than the other diagonal elements. This is because no Preisach elements change values when the input range is exceeded. Consequently, the



(a) Measured output and hysteresis model output.



(b) Input-output relationship.

Figure 7.4: Model output when identifying using a 20 Hz sine wave, with 90 V amplitude, with $n_h = 20$ as discretization level.

output of this model will be the same for all input values larger than the largest α value. Since the input is larger than the range of the identified model, the constant contribution will still be described by the η_0 variable.

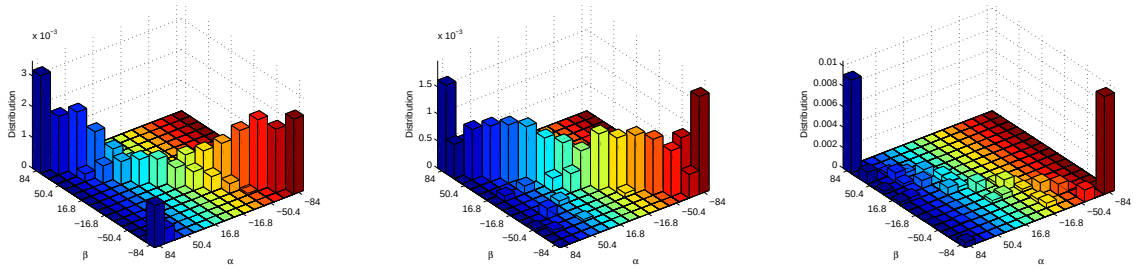
In summary, the range of the input signal should correspond to the range defined for the model to be identified. However, in the discrete case, the input must exceed the largest α value in order to excite all Preisach elements.

7.1.2 Continuous Model

In contrast to the discrete model, the continuous model has an infinite amount of output values, since it relies on the signed area of each Preisach element. Thus, the output can take whatever value in the range of possible outputs, this was also described in Chapter 3.

The output of a continuous model is shown in Figure 7.4a where it can be seen that the output is significantly more accurate than the discrete model shown in Figure 7.1 and 7.2. Instead of being either +1 or -1, each Preisach element is calculated as areas, where all elements the memory curve, ψ, does not pass through, is either +1 or -1, while the rest of the elements have a positive and negative area, which is summed in order to achieve a signed area. For this reason, the output can take an infinite amount of output values.

One disadvantage of using a continuous method for modeling hysteresis, is that the memory curve of the system has to be calculated and stored. This includes all edges of the memory curve, which can, due to the infinite number of possible output levels, be an infinite



(a) Identified with an input range of [-84,84].

(b) Identified with an input range of [-91,91].

(c) Identified with an input range of [-150,150].

Figure 7.5: The identified Preisach weight function, when using the continuous model, for three different input ranges, while the model is defined for an input in the range $[-90, 90]$.

number of values. Consequently, the method for calculating the edges should aim for a less computational expensive method for decreasing the time consumption.

In Figure 7.5 three plots with different identified Preisach weights are given, with the same input ranges as in Figure 7.3 for the discrete case. The largest differences from the two methods is that even when the input range is smaller than the range defined for the model, the identification can calculate values on the end diagonal, as long as the input has values inside the (α, β) coordinate of a relay. However, if the maximal input is smaller, say a maximum amplitude of 50, then the identified model would find weights for all elements where the input value excites the relay. Even though all the diagonal elements are found in Figure 7.5a, the corner weight has a value corresponding to a constant value, which should be described by η_0. The reason for this is, most likely, that since the input only excites a small part of this Preisach element, the least squares algorithm found is optimal in order to place the weight in the corner element instead of in η_0.

When the entire Preisach plane is excited, as in Figure 7.5b, all the diagonal weights are found, without any values in the corner element, just as in the discrete case. Furthermore, when an input that exceeds the model range is applied, the same behavior is observed on the identified Preisach weights, as in the discrete case.

7.1.3 Comparison

The largest difference between the discrete and continuous method is the resolution of the output. This can be observed by comparing Figure 7.1 with Figure 7.4a, where the output is significantly more accurate in the continuous case compared to the discrete case. This

difference are decreased if a larger discretization level is used in the discrete model, as discussed in Figure 7.2. However, a larger model also requires more calculation, as shown in Tan and Baras [2005].

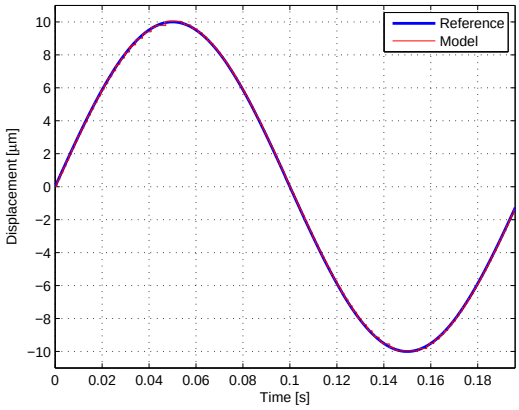
If only the modeling of hysteresis is in question, and not the inversion, the continuous method gives a more accurate description of the output than the discrete method, with a lower discretization level. However, the computation of the continuous method includes calculating the areas of all memory-curve elements, increasing the computational effort, since all the memory curve edges has to be found. Moreover, in the next section the inversion results of these two methods will be presented, and the continuous inversion algorithm is shown to have some challenges creating high frequency inputs.

7.2 Inversion of the Preisach Model

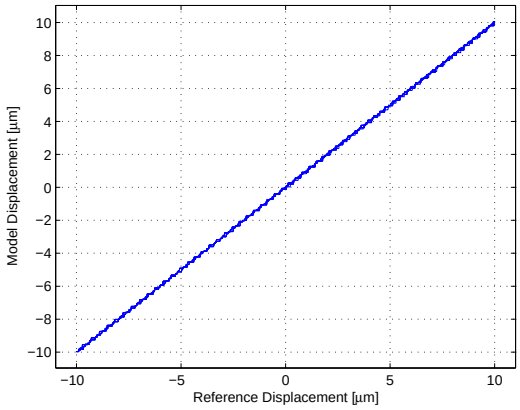
This section will present results for the inversion of both the discrete and continuous Preisach method. First inversion of the discrete model is shown with examples, before the continuous inversion is plotted as well. To avoid a discrete input signal to the actuator and still maintain the compensation of hysteresis, three methods for acquiring the query points for a linear interpolation method is presented. In the continuous method, interpolation is not necessary, since the input is a smooth function. However, some practical properties of the continuous inversion is discussed, such as the choice of a d_{scale} variable, which decide the step size of the continuous closest match algorithm, and the use of a noisy reference signal.

7.2.1 Inversion Based on the Discrete Model

The inversion of the discrete model is based on the discrete version of the closest match algorithm presented in Chapter 4, where the input was chosen such that it would give an output as close as possible to the reference output. Consequently, a larger model would give a more accurate inversion, since the output can take more unique values. This section first presents a simulation of the inversion, which shows that the algorithm compensates for the hysteresis described in the model. This is followed by tests of the inversion routine on the piezoelectric actuator.



(a) Reference and simulated model output of the inversion.



(b) Reference-model relationship.

Figure 7.6: Simulation of a discrete inversion, based on a model identified from a increasing sine wave with 100 reversals and a discretization level of $n_h = 50$.

Simulation of the Discrete Inversion

To validate the closest match algorithm, a simulation of the inversion was conducted. This was done by comparing a reference signal with the model output obtained by applying the inverted input to the hysteresis model. This should result in a approximately linear reference-model relationship if the model is compensated.

The resulting simulation when a model identified with 100 input reversals and discretization level $n_h = 50$ was used, is shown in Figure 7.6. From these plots it is possible to verify the implementation of the inversion algorithm. If the relationship in Figure 7.6b had a shape resembling a hysteresis loop, the inversion algorithm would have been badly implemented. Such a simulation is important since this ensures that the inverted input is correct in the context of the identified model, and consequently it should be able to compensate for hysteresis.

Inversion Using the Discrete Input Directly

An example of a discrete inverted input is illustrated in Figure 7.7, where the input has the familiar staircase shape. The model this inverted input is based on, was identified with an increasing sine wave with 150 reversals and base frequency 5 Hz, with a discretization level of $n_h = 100$. This accounts for all discrete inversions in this section, unless noted otherwise. The reference is a sine wave at 5 Hz with 10 μm amplitude.

The results from a direct inversion, with no interpolation, is shown in Figure 7.8. Fig-

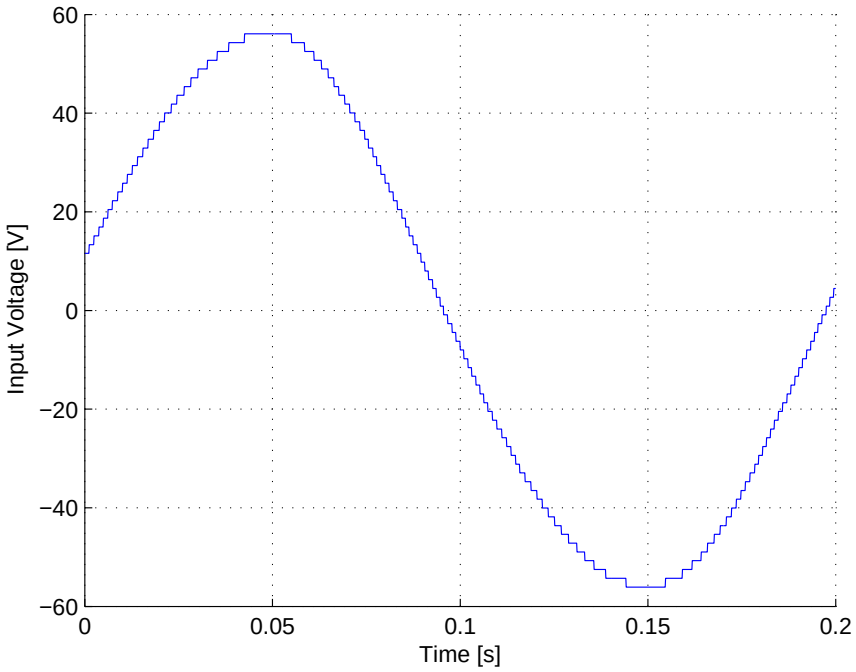


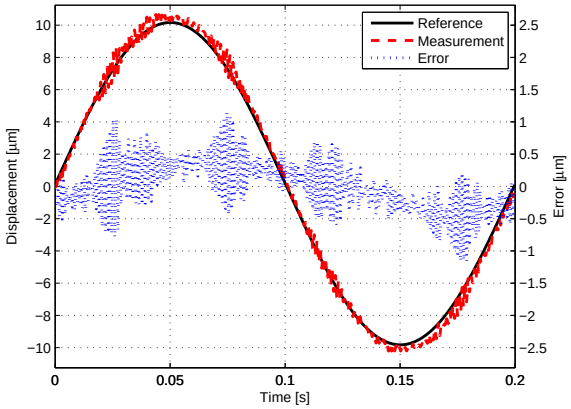
Figure 7.7: Example of discrete input.

ure 7.8a, shows the reference and measured displacement on the left axis, while the error is shown on the right axis. Since the input has high frequencies due to the staircase steps in Figure 7.7, the resonance dynamics of the actuator gives a vibrating and noisy output. Thus, such an input signal does clearly not produce an inverted input which can be used in practice. This conclusion is supported by the reference-measurement relationship shown in Figure 7.8b, where the vibrations in the output are clearly visible. However, even though vibrations are present in the output, some compensation of hysteresis is achieved.

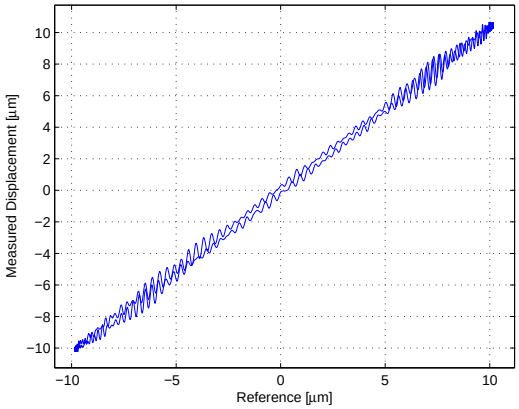
Interpolation

Due to the high frequencies in the input signal in Figure 7.7, the input should be smoothed. As described in Chapter 4, linear interpolation between the input levels can be used for smoothing. However, the literature does not mention what query points should be used for the interpolation.

In order to study the placement of the query points, two methods were initially proposed. The first method used the first point of a new discrete value, as presented in Figure 7.9a, while the second method found query points in the middle of each discrete value, as shown in Figure 7.9b. Due to the results from the two methods above, a third method combining the first two methods were proposed, illustrated in Figure 7.9c, where all the left corners are

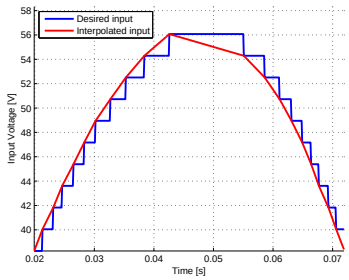


(a) Time-series for the reference signal, with the corresponding measured displacement and error.

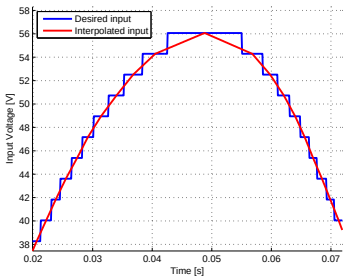


(b) Input-output relationship.

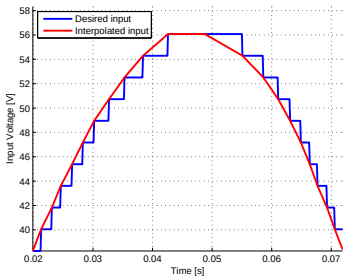
Figure 7.8: Hysteresis compensation when using the inverted input directly. The reference is an sine wave at 5 Hz with 10 μm amplitude. The hysteresis model has a discretization level of $n_h = 100$.



(a) Interpolation of all left corners.

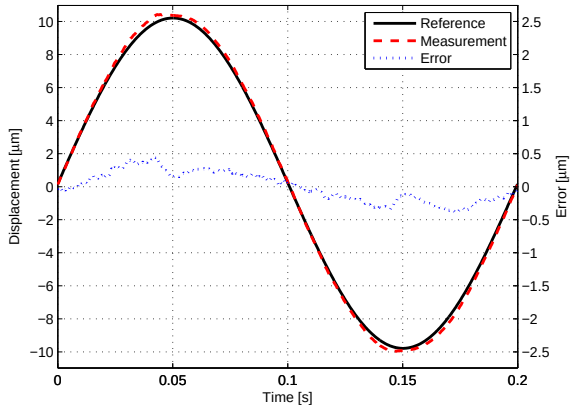


(b) Interpolation of all middle values.

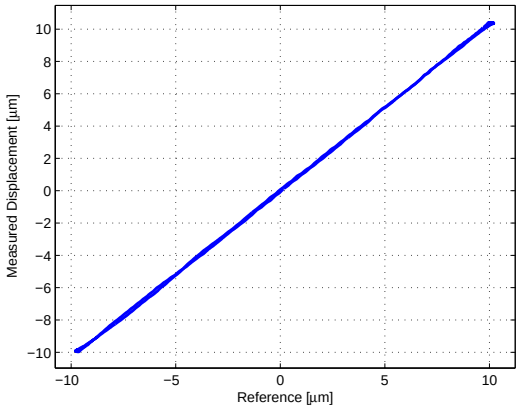


(c) Interpolation of all left corner and the middle of the extremums.

Figure 7.9: The three interpolation methods used.



(a) Time-series for the reference signal, with the corresponding measured displacement and error.



(b) Input-output relationship.

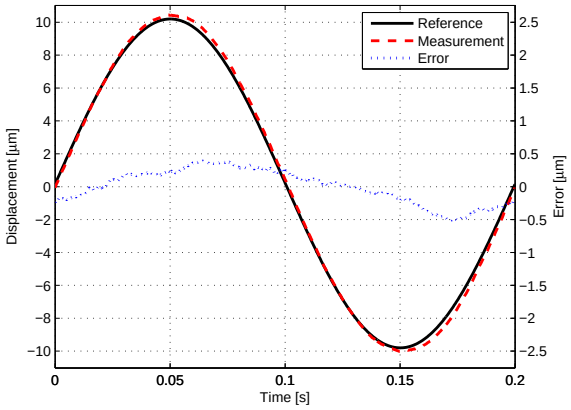
Figure 7.10: Hysteresis compensation when using the interpolation in the left corners. The reference is an sine wave at 5 Hz with 10 μm amplitude. The hysteresis model has a discretization level of $n_h = 100$.

query points in addition to the middle point of all extremums. This was proposed in order to study the end-effects in the reference-measurement relationship.

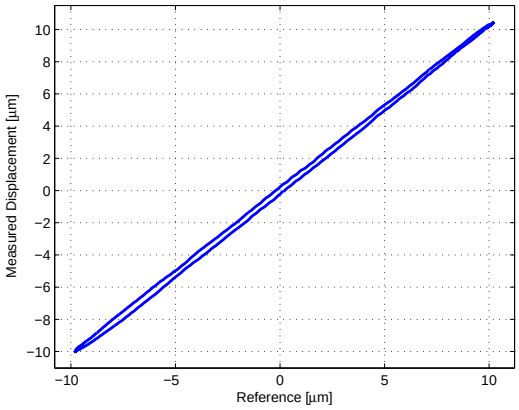
Figure 7.10 presents the results based on an inversion with linear interpolation with query points in the left corners, as shown in Figure 7.9a. Compared to the results with no interpolation there is a significant improvement. The error shown in Figure 7.10a does not contain any high frequency components, which indicates that the input signal does not excite the resonance frequency of the system. The reference-measurement relationship in Figure 7.10b has a linear shape, which means that the hysteresis is compensated. However, even though it is difficult to see, the response in the extremas is not a perfect fit. Consequently, the third query point method for interpolation was proposed.

The results from the second query point method, based on the middle values, are shown in Figure 7.11. It should be noticed that, is that the reference-measurement relationship in Figure 7.11b does not compensate for the hysteresis with the same accuracy as the first query point method. This hysteresis effect can also be seen in Figure 7.11a, where the measured output is lagging behind the reference.

Lastly, the interpolation combining the two methods above was used. The idea with this method is that interpolation with only the left corners gave a small error in the extremums of the reference signal. Therefore, a method which adds the middle of the maximum and minimum to the query points were tested, and the results are shown in Figure 7.12. Comparing this query-point method to the first method, shows no large difference. Thus, this method



(a) Time-series for the reference signal, with the corresponding measured displacement and error.



(b) Input-output relationship.

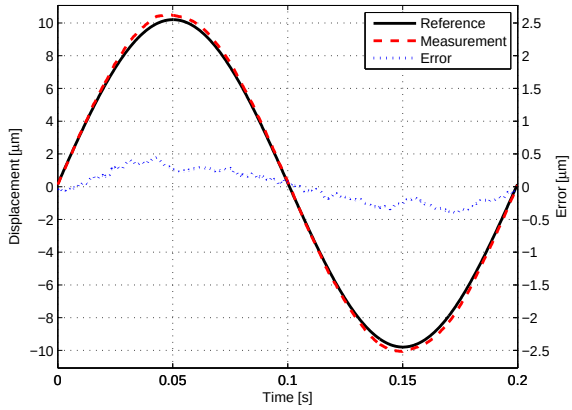
Figure 7.11: Hysteresis compensation when using the interpolation in the middle values. The reference is an sine wave at 5 Hz with 10 μm amplitude. The hysteresis model has a discretization level of $n_h = 100$.

does not improve or worsen the inversion results in any way.

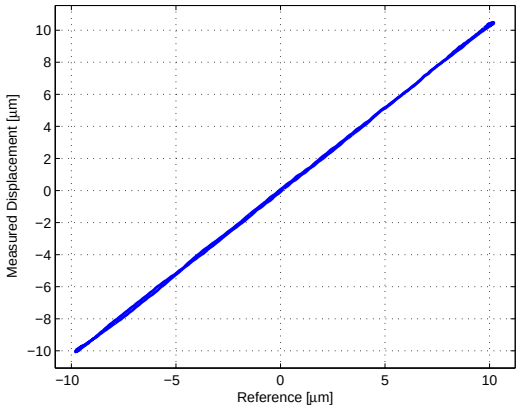
To summarize, the different interpolation methods above all have a maximum error of 0.5 μm, where this occurs mostly at the maximums and minimums. Hence, such an error corresponds to approximately 5% in the extremums. Compared to the hysteresis compensation using the Coleman-Hodgdon model in Eielsen et al. [2012], where the error is about 1%, inversion of the Preisach model is not satisfying. A more accurate interpolation method can increase the performance, especially in the extremums, where none of the proposed methods gave satisfying results.

On the other hand, a well designed Preisach method, can be used for compensation of reference signals with different or varying amplitudes. The Coleman-Hodgdon method is designed for a specific amplitude, and the parameters have to be updated in order to describe other variations in the input. Nevertheless, a 5% error in the inversion procedure can limit the effect of control applications.

All of the above inversion results were done with a discretization level of $n_h = 100$. In order to study if a model with a lesser level of discretization could give similar results, an inversion with $n_h = 50$ was conducted, where the input signal used for identification is the same as in the above cases, and the left corner query point technique is applied. The reference displacement is still a sine wave with 5 Hz frequency and 10 μm amplitude. The results of this inversion is shown in Figure 7.13, where the hysteresis is still compensated very satisfactorily. As commented on in the results for the left query point method for $n_h = 100$, the



(a) Time-series for the reference signal, with the corresponding measured displacement and error.



(b) Input-output relationship.

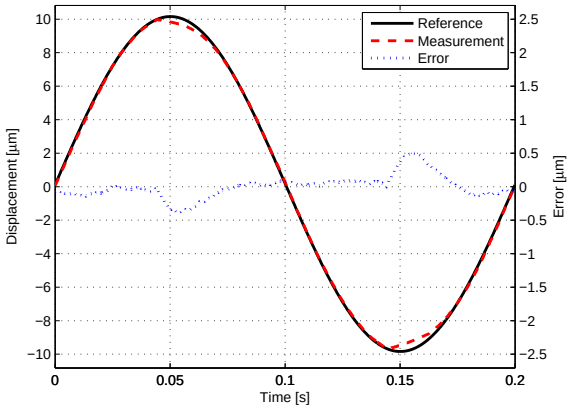
Figure 7.12: Hysteresis compensation when using the interpolation in the left corners, in addition to the middle of the extremums. The reference is a sine wave at 5 Hz with 10 μm amplitude. The hysteresis model has a discretization level of $n_h = 100$.

errors in the maximums and minimums are more significant, this is also the case here. Still, the results from this method implies that a discretization level of 50 seems not to reduce the performance from the $n_h = 100$ case.

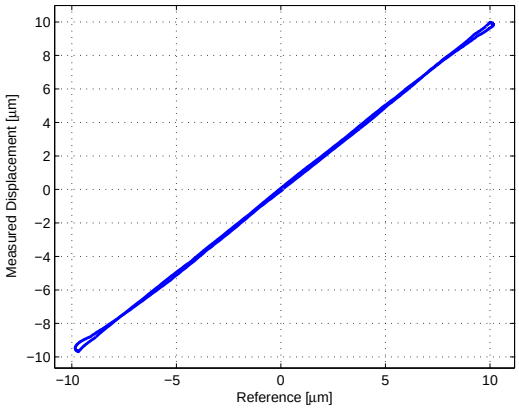
From the results above, the inversion of the discrete method is shown to work satisfactorily, and except of the query point technique based on the middle values, the linear interpolation method was shown to work well. All these results are based on inversions from a model identified with the same input signal, thus the models are based on the same information. However, if different input signals are used for identification of different models, each of these models will be subject to the disturbances and environment of the actuator at that time. This means that, for instance, if a model is identified one day, and the inverse is tested the next day, the piezoelectric actuator can change its response due to temperature differences or other environmental variations. Consequently, all results have to be interpreted based on this behavior of the system.

Interpolation in On-line Applications

Of the three proposed interpolation methods, the two best schemes proved to be the once where the left corners of the inverted input was used as query points. However, this introduces a challenge for on-line applications. These interpolation methods assumes that the next corner of the inverse input is known, which will not be the case in on-line applications. Both the length of the current input value and the height of the next input value depends



(a) Time-series for the reference signal, with the corresponding measured displacement and error.



(b) Input-output relationship.

Figure 7.13: Hysteresis compensation when using the interpolation in the left corners, with discretization level $n_h = 50$. The reference is a sine wave at 5 Hz with 10 μm amplitude.

on the reference signal. For this reason, the interpolation methods presented here will be severely limited by a time delay in the calculation of the interpolated values, since it will lie one input level behind.

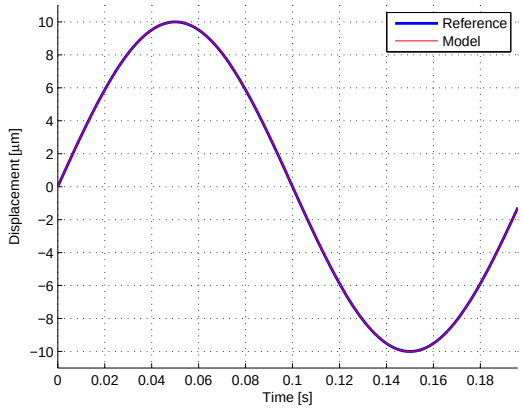
Consequently, this routine has to be improved in order to be used in on-line implementation. In contrast to the discrete case, the continuous method will not suffer from the same limitation, since the inverse is calculated continuously.

7.2.2 Inversion based on the Continuous Model

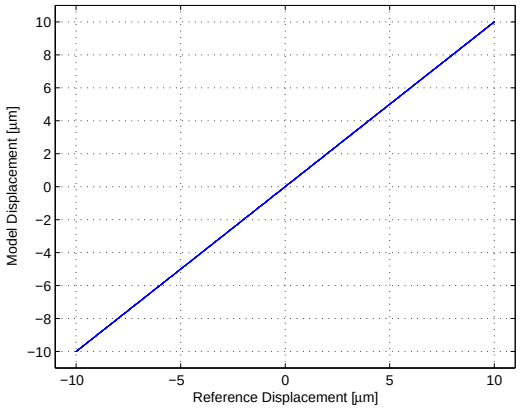
The results from the inversion of the continuous method will be introduced after presenting a few considerations of the inversion algorithm. First, a validation of the inverse model is conducted, followed by a discussion of the choice of a scaling variable in the inversion algorithm, where the error between the desired output and a model output is shown. Secondly, a non-smooth signal is used for inversion in order to show how the output of the inversion algorithm responds. Finally, some inversion results are shown for the continuous model with a discretization level of $n_h = 25$ and $n_h = 50$.

Simulation of the Continuous Inversion

In the same manner as for the discrete inversion, a simulation step was performed to validate the closest match algorithm. From Figure 7.14, the simulated model output can be seen to correspond to the reference signal. Consequently, the inversion routine is well functioning,



(a) Reference and simulated model output of the inversion.



(b) Reference-model relationship.

Figure 7.14: Simulation of a continuous inversion, based on a model identified from an increasing sine wave with 50 reversals and a discretization level of $n_h = 25$.

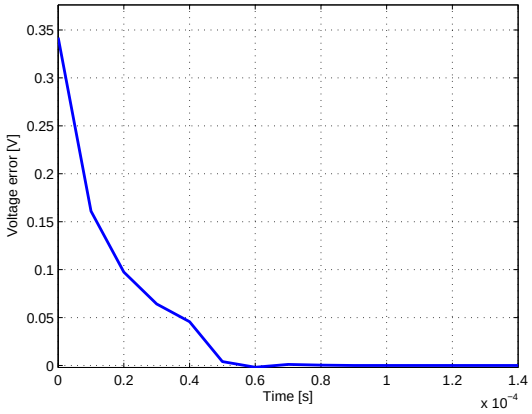
and gives a input signal which compensate for the hysteresis effect described in the model.

D-scale Variable

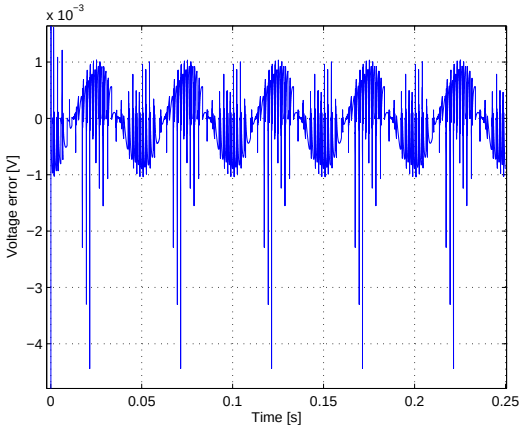
As described in Algorithm 4 in Chapter 4, a d value should be chosen so that $0 < d < \min\{d_1^{(k)}, d_2^{(k)}\}$. In this thesis, this value has been chosen by a d_{scale} value, where $0 < d_{scale} < 1$. In this section, the difference between a high and low value for d_{scale} is studied, where this value can be interpreted as a step length in the closest match algorithm.

By using a sine wave as desired output, the inversion error is shown in Figure 7.15. Since the d_{scale} value refers to a step length in the algorithm, a large value refers to a longer step, which again means that the error will quickly become smaller. For a lesser d_{scale} value the algorithm will use more time in order to converge to a small error, however the error will also stay smaller, since it does not take any large steps away from the correct solution, when it is first found. The large error in the first milliseconds is shown in Figure 7.15a, while, by comparing Figure 7.15b and Figure 7.15c, the error can be seen to be smaller for the case where the d_{scale} variable is small.

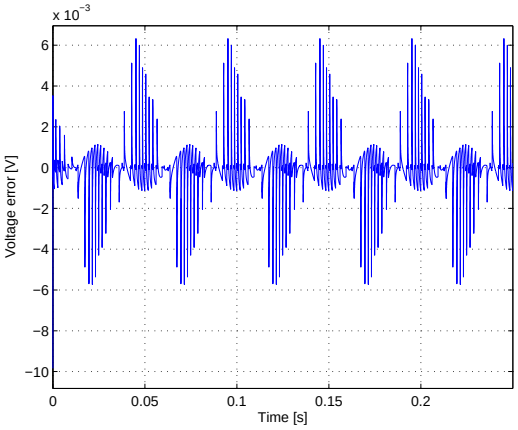
Another aspect of the choice of the d_{scale} variable, is that a smaller value will increase the run time of the algorithm when the error is large, since it requires several iterations in order to come close enough. On the other hand when the error is small the run time of the algorithm with a small d_{scale} value will be shorter compared to a large d_{scale} value.



(a) Inversion with $d_{scale} = 0.1$, zoomed in on the first milliseconds.

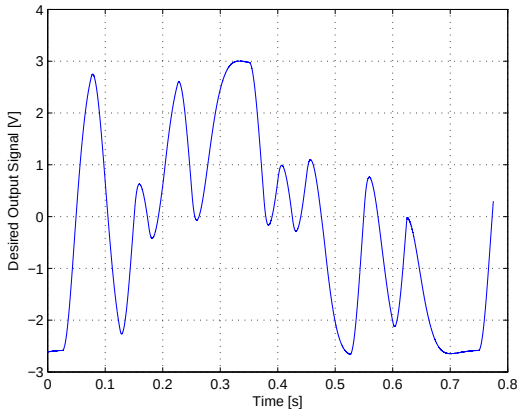


(b) Inversion with $d_{scale} = 0.1$, zoomed in on the main error.

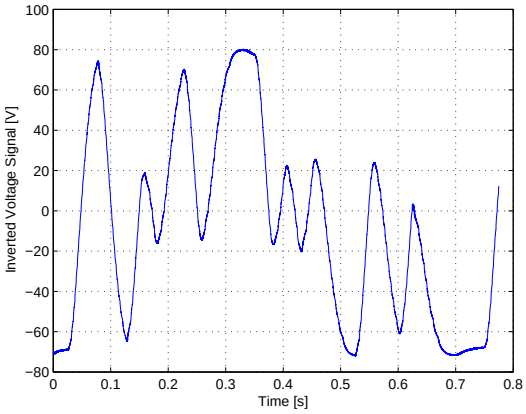


(c) With $d_{scale} = 0.9$.

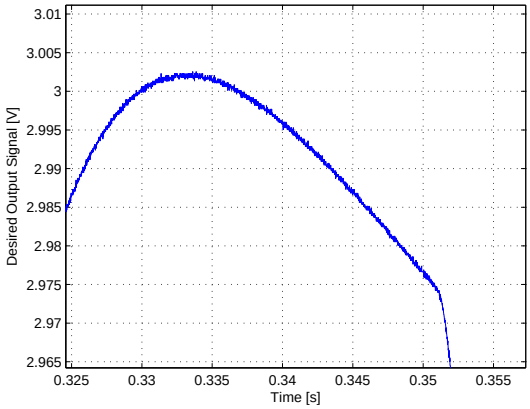
Figure 7.15: Simulation of an inversion. A sine wave is used as the reference output of the inversion algorithm, where the model used for inversion is based on a increasing triangular input. In Figure (a) and (b) the error between the reference and the model output is shown, with the inverted signal, found by using $d_{scale} = 0.1$, as input. In (a) the first milliseconds is shown, while (b) shows the main part of the error. Figure (c) is the same error, with the inverted signal found by using $d_{scale} = 0.9$ in the inversion.



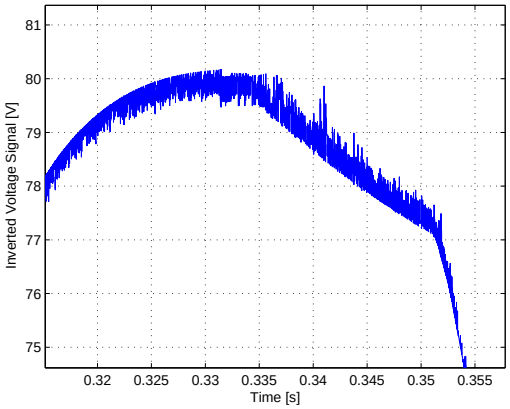
(a) The desired output.



(b) The calculated inverted input.



(c) Zoomed in version of Figure 7.16a.



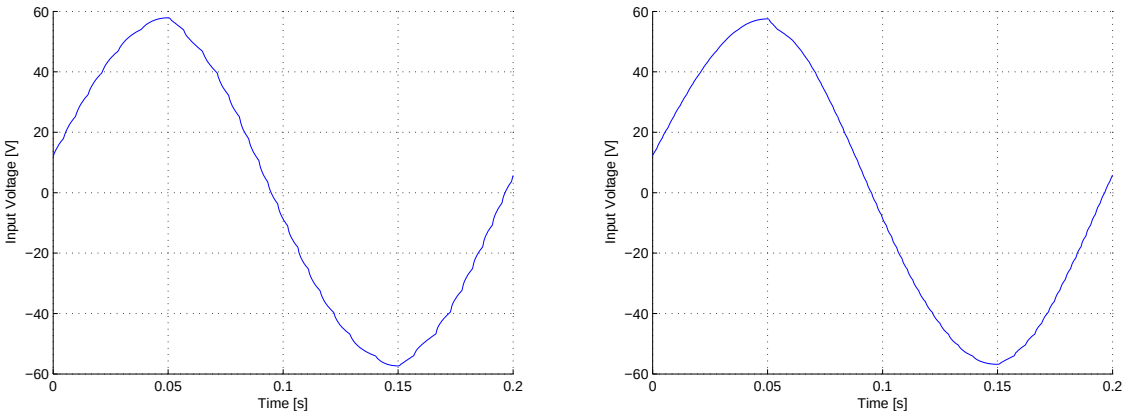
(d) Zoomed in version of Figure 7.16b.

Figure 7.16: Inversion of the signal in (a), which is contaminated by noise. The resulting input from the closest match algorithm is shown in (b). In (c) and (d) the same part of the signals are enlarged in order to show the noise, from (a) and (b) respectively. Due to the not exact fit of the closest match algorithm, the noise is magnified in the calculated input. The inversion was done with $d_{scale} = 0.1$

Nonsmooth Reference Signal

In this section, a noisy signal is used as the desired output for the inversion algorithm. The noise is not large, as seen in Figure 7.16c, however, the main issue with such an input is that it changes direction frequently, which makes the closest match algorithm search for the desired input in different directions.

The signal in Figure 7.16a is used as the desired input. The resulting inverted input can be seen in Figure 7.16b, which does not indicate a noisy signal. However, when taking a closer look at the signals, that is, Figure 7.16c and 7.16d, it can be noticed that the noise in the desired output has been magnified for the inverted input. In Figure 7.16, the d_{scale} value was 0.1, if this was larger, the disturbance on the inverted signal would also be larger. As



(a) Input with $n_h = 25$.

(b) Input with $n_h = 50$.

Figure 7.17: Example of input signals when using the continuous inversion routine.

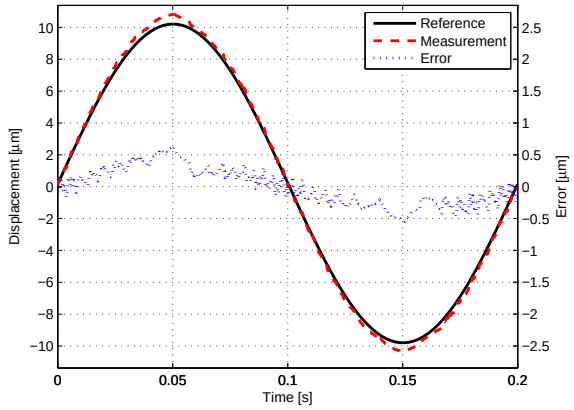
discussed in previous chapters, the piezoelectric actuator has a resonance frequency at 800 Hz, which will be excited when the input contains high frequency elements, such as noise.

Consequently, the desired input should be a smooth function, which will avoid the high frequency noise in the input signal, shown in Figure 7.16d. This also applies for the discrete model, in perhaps a larger extent than the continuous, since the inverted input switches between different discrete values.

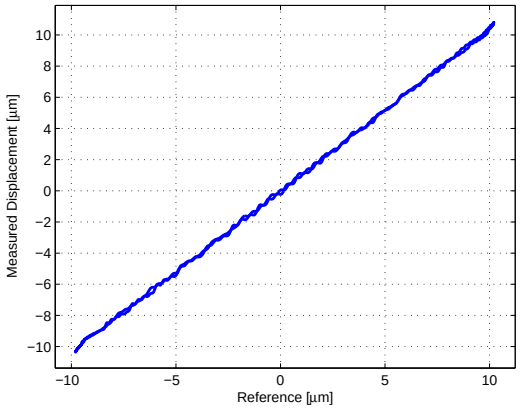
Results of Continuous Inversion

The results for the inversion of the continuous method is shown in this section, where the identified models are based on the same input signal as in the discrete case, namely an increasing sine wave with 150 reversals and base frequency 5 Hz. Additionally, two different discretization levels were tested, $n_h = 25$ and $n_h = 50$.

The input signals calculated from the inversion of the models are shown in Figure 7.17, for the inversion based on a model with a discretization level of both $n_h = 25$ and $n_h = 50$. In Figure 7.17a, it should be noticed that the effect of the triangular Preisach elements. As shown in Figure 3.6, assuming an increasing input, the area of the input increases at a smaller rate in the bottom of a triangular cell, than in the higher regions. This effect influences the inverted input as well. If the reference signal is increased at a constant rate, the inverted input will increase slowly when the memory curve passes through the higher regions of the cell. When the input increases to the next Preisach relay, it suddenly has to increase at a faster rate in order to create an output which follows the constant rate of the reference. This assumes

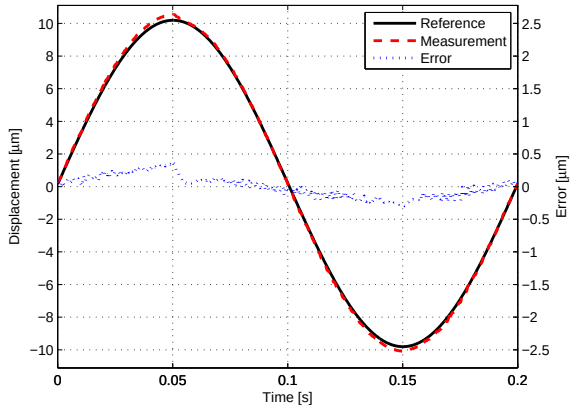


(a) Measured output and hysteresis model output.

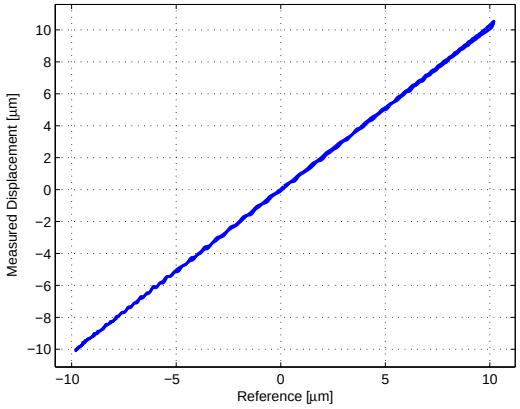


(b) Input-output relationship.

Figure 7.18: Inversion with $n_h = 25$ based on the continuous method.



(a) Measured output and hysteresis model output.



(b) Input-output relationship.

Figure 7.19: Inversion with $n_h = 50$ based on the continuous method.

that the weights on the diagonal of the Preisach plane has dominant values, which seems to be the case for the piezoelectric actuator, for example, the distribution seen in Figure 7.5.

The effect of the varying slope, is higher frequencies in the input signals, which again can excite the resonance frequency of the system. This can, for instance, be seen in Figure 7.18, where the results from the inversion of the continuous model with discretization level of $n_h = 25$ is shown. By studying the error, vibrations can be seen in the measured output. In addition, the reference-measurement relationship in Figure 7.18b have a weak vibrating effect.

The vibrations can be limited by increasing the model size, as shown in Figure 7.19, where the discretization level is $n_h = 50$. The input signal is smoother than in the case for $n_h = 25$,

which causes less vibrations in the actuator. The resulting error in this model is about the same level as in the discrete case, however, as mentioned above, the computational effort is larger for continuous models. As a result, the continuous method does not provide any significant increase in performance compared to an interpolated discrete method, when the model complexity is considered as well. For on-line applications, on the other hand, this method might be better suited than the discrete method. It does not suffer from the same issues that occur when query points for the interpolation should be found. Still, the model complexity should be small when using such methods on on-line applications, which again introduces vibrations in the inverted signals for higher frequency reference signals. Consequently, there are both pros and cons for use of the Preisach model in on-line applications.

7.3 Experimental Design

This section will study the various input signals, shown in Chapter 5, used for identification of the Preisach model. This will be seen in the context of the theory of PE input signals, also presented in Chapter 5. First, a measure for comparing the error in the input signal is presented, followed by the inversion errors in table format.

Additionally, a simulation of the small amplitude input signal is illustrated. However inversion results of this method has not been tested.

7.3.1 Choice of Input Signal for Identification

In order to study which input signal gives the most information about the system, there are several factors to consider. As discussed in Section 7.1, the input needs to cover the range defined in the model, to give a valid identified model. Moreover, there is a question of how to achieve the most information about the actuator. To answer this question, the error between the reference voltage y_d and the measured voltage y_m was compared by summing the square errors. These errors corresponds to a displacement of 5 μm/V, as noted in 5.5

$$\text{error} = \sum_{i=1}^{N} (y_{d_i} - y_{m_i})^2 \tag{7.1}$$

where i refer to each input value, where the signal consists of N values. Since the piezoelectric actuator is sensitive for changes in the environment, the bias in the error was removed, since this can be due to environmental variations in the laboratory equipment. This gives a

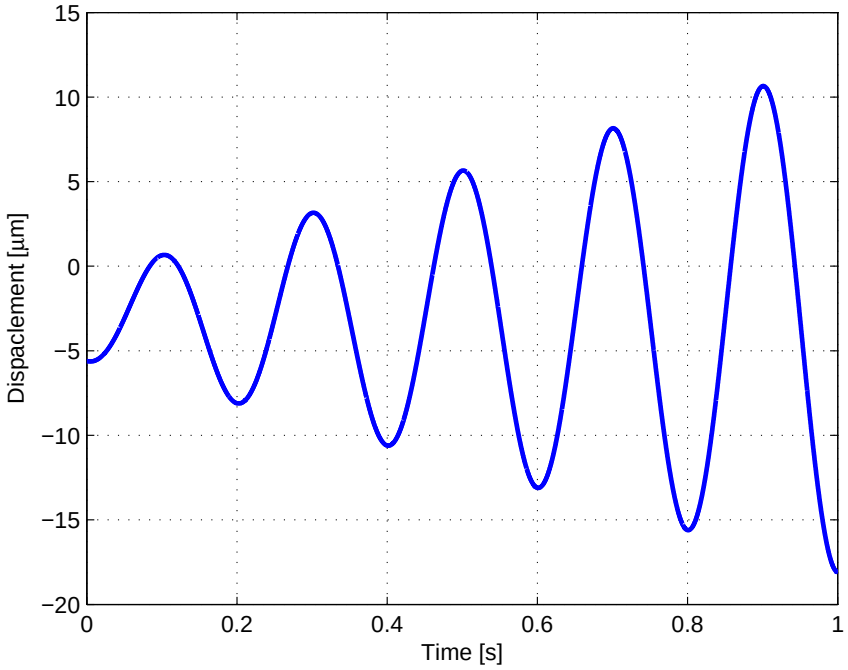


Figure 7.20: Reference signal used for testing the best input.

Table 7.1: Comparison of the squared error of models identified from different input signals. The value is an average of three experiments with each model having $n_h = 50$ as discretization level, and a sampling frequency of 10000 Hz.

Type of Signal	**Average Error** $[V^2]$
Sine with amplitude 91	331.1
PRBN signal	125.4
Increasing sine signal - 50 reversals	63.68
Increasing sine signal - 100 reversals	52.29
Triangular signal - 50 reversals	56.56
Triangular signal - 100 reversals	51.11

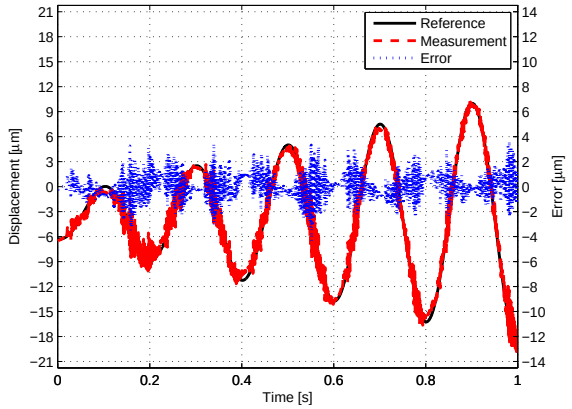
good measure of the quality of the inversions, independent of bias and the length of the signal. However, the creep nonlinearity was not compensated, which may have caused a slow drift in the measurements. On the other hand, assuming that the frequencies of the input signals are approximately identical, this creep effect can be neglected in the comparison.

The reference signal used for testing the input signals is shown in Figure 7.20, and consists of a linearly increasing sinusoidal input signal of five waves with 5 Hz frequency. Since the Preisach model is rate-independent, a different frequency will not influence the calculated inverted input. However, vibration dynamics in the actuator can be excited, causing vibrations in the measurements.

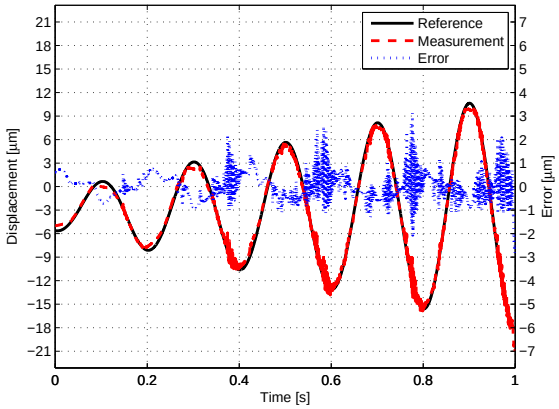
Table 7.1 summarizes the results from the tests of the inversion signals by averaging the error voltage in Equation 7.1 of three experiments based on each model. It can easily be seen that the error from the model based on the single amplitude sine wave is by far the largest, followed by the error based on the PRBN input signal. The large difference between these two signals is most likely due to the increased number of input reversals in the PRBN signal, compared to the single-amplitude sine. While the single-amplitude sine wave only has reversals for the maximum and minimum relay elements, the PRBN signal consists of several reversals for different input levels. Hence, it gives more information about the hysteresis effect than the single-amplitude sine wave, but still has too few reversals to describe the actuator model fully. It can also be noticed that the triangular signal appears to give a slightly better performance than the rising sine signal. The reason for this is unclear, however it can be that a triangular shaped signal showa the hysteresis effect more accurately, or it could just a be coincidence. Either way, this should be studied more closely in the future.

Due to the discretization level of $n_h = 50$, the last four input signals in Table 7.1 have an equal or larger amount of input reversals than discretization level. Due to this, the identified models based on these signals have one reversal for each discretization level. Consequently, the lagging effect due to hysteresis, is seen for all discretization levels, giving information about each discrete input level. By studying the results in Table 7.1, these reversal inputs are seen to give a significant smaller error than the single-amplitude sine wave or the PRBN signal. Furthermore, an input signal with more reversals will give an even more accurate description of the model, seen for both the sine wave and the triangular signal. These signals consists of 100 reversals each, corresponding to two input reversals in each discretization level, removing effects from noise in the measurements.

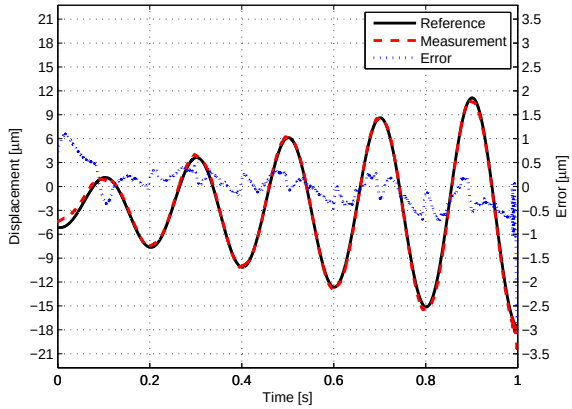
Figure 7.21 gives another perspective of the difference between the reference and measured displacement for the various input signals. The two first plots, containing the results from the single-amplitude sine wave and the PRBN input, show large oscillations in the measured signal. This is due to the not complete model description these input signals lead to for this discretization level. When the input signal used for identification do not have enough input reversals, the resulting Preisach weight function will include zero valued elements in the diagonal. Due to the nature of the closest match inversion algorithm described in Chapter 4, this will result in an inverted input signal containing discontinuities for continuous models and oscillations in discrete models. This problem will be described in detail in Sec-



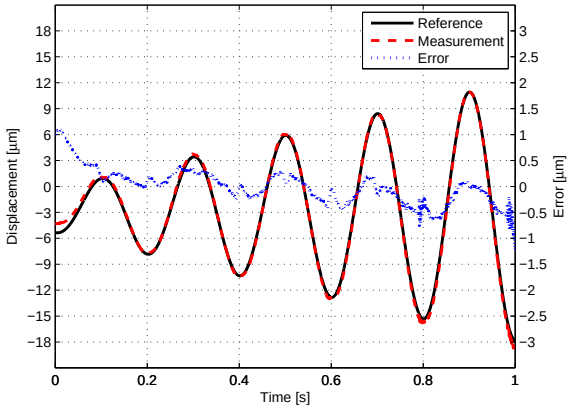
(a) Sine with large amplitude.



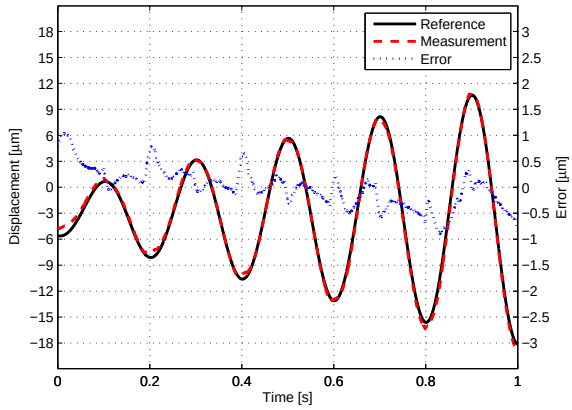
(b) Pseudo random binary noise.



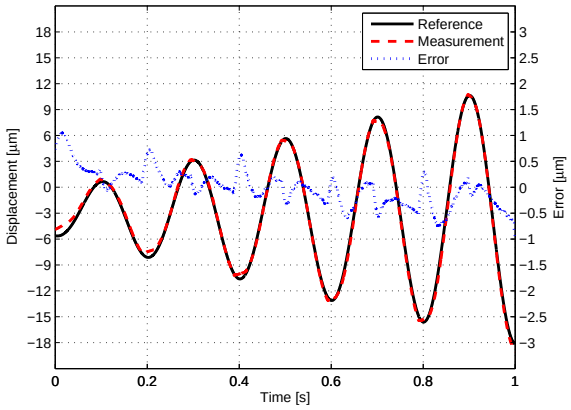
(c) Increasing sine with 50 reversals.



(d) Increasing sine with 100 reversals.

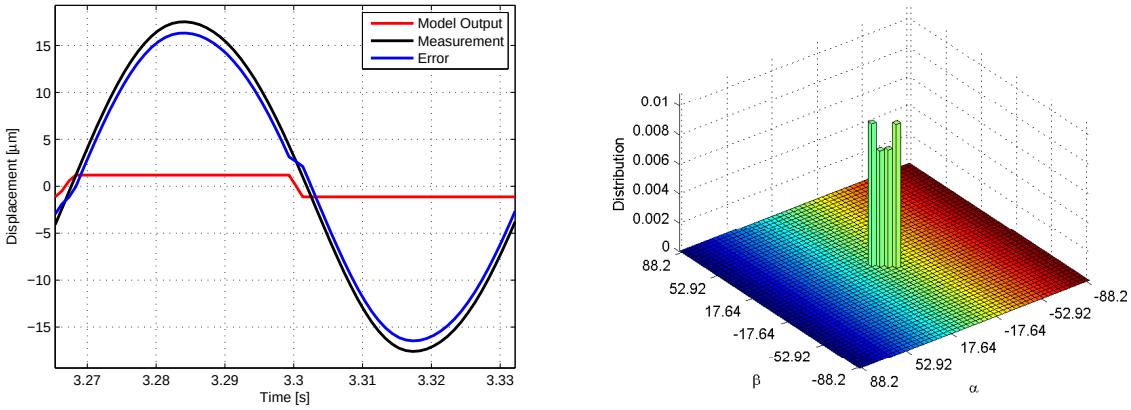


(e) Increasing triangular signal with 50 reversals.



(f) Increasing triangular signal with 100 reversals.

Figure 7.21: Output and error from the experimental design test for each of the tested signals. Notice that the error axis in (a) and (b) has a larger scale than (c)-(f). All models had $n_h = 50$ as discretization level.



(a) Output of simulation.

(b) Distribution of Preisach weights.

Figure 7.22: Simulated test of the model based on the small amplitude input signal, where the measurements of a large amplitude is used as input. The blue line in (a) is the error between the estimated output and the measured output, while the green and red lines are the measured and fitted output respectively. In (b) the Preisach weights found from the identification of the small amplitude sine is plotted.

tion 7.4.1. Resulting from the larger amount of reversals in the PRBN signal, the oscillations in the input signal is less significant than for the single-amplitude input.

The last four input signals are plotted from Figure 7.21c - 7.21f, where the oscillations experienced in the first two input signals does not appear. Hence, the inputs used to identify these models have more information, which consequently gives nonzero diagonal elements in the Preisach distribution.

As shown previously, when identifying a model with a too small input range, the corresponding Preisach weight function will have several zero elements, since not all discretization levels were excited. This is also shown in Figure 7.22, where a simulation of the Preisach model is performed by applying a sine input to a model identified from a small-amplitude input signal. This responds to a large error between the output from the model and the measurement. The reason for this can be seen in Figure 7.22b, where only a small part of the Preisach plane has non zero weights. Thus, the output from this model can only take values in a small range around zero. This is why no inversion tests were performed by using the model based on the small-amplitude input signal.

To conclude, an input signal used for identification should contain enough input reversals, that is, at least as many as the discretization level. If this is not the case, then the zero-elements of the diagonal causes oscillations in the inverted input signal for the discrete model, or discontinuous in the continuous model. Additionally, more reversals will give a

more accurate description of the piezoelectric actuator.

7.4 Errors in the Signal Loop

As indicated in Chapter 6, the instrumentation in the signal loop, such as low pass filters and sensors, introduces negative phase in the system. Consequently, the measurement from the system will experience a so-called “phase lag”. These effects will increase the observed “lagging” effect, resulting in an identified model trying to describe the sum of both hysteresis and negative phase effects.

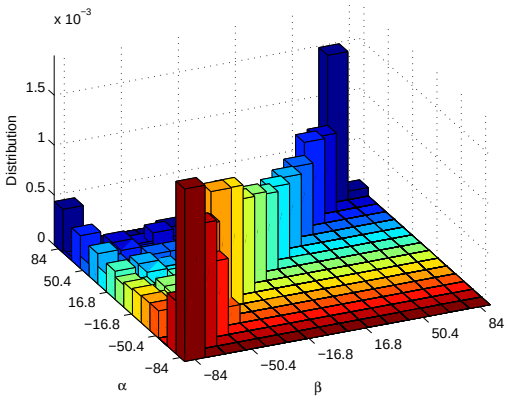
This section will study the effect of negative phase in hysteresis identification, mainly due to low pass filters and time delay. First, however, an effect of zero-elements on the diagonal in the Preisach distribution is discussed, before it is shown that both a low pass filter and time delay can introduce such zero-elements in the Preisach distribution. In addition, several xy-relationships between model and measured displacement illustrate the negative phase effect on the Preisach operator.

7.4.1 Zero-elements on the Diagonal

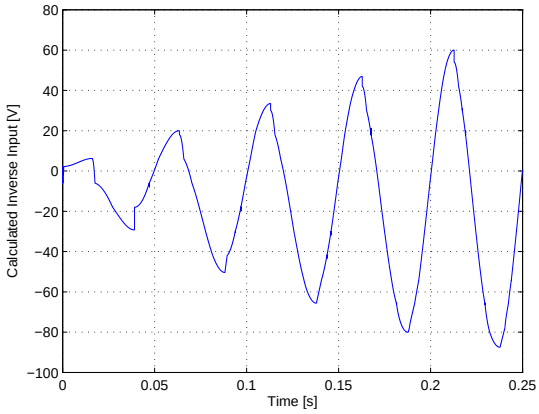
This section shows an example of a continuous model, however the same arguments can be transfered to the discrete case as well.

The negative phase from low pass filters affects the identified Preisach weights. In Figure 7.23a the weight function found by using an increasing sine wave with 15 reversals as input signal, for a continuous Preisach model with discretization level $n_h = 15$ is shown. Based on the theory described in Chapter 5, together with the results from the previous section, this should be sufficient to achieve a valid description of the model. However, the presented distribution contains several zero-elements on the diagonal, due to low pass filtering of the measured signal. Remembering the discussion of existence and continuity of the inverse of the Preisach model, it was argued that if the Preisach weight function is nonzero on a thin strip along the diagonal, then the Preisach operator is invertible with a continuous inverse.

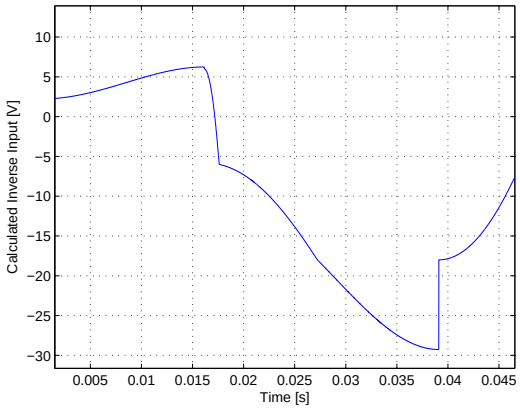
However, when this is not the case, the existence and continuity of the inverse Preisach model cannot be guaranteed. The example in Figure 7.23 illustrates this. When inverting the model for a reference signal the inverted input will contain discontinuities, as shown in Figure 7.23b and 7.23c. This appears when the reference signal switches direction from



(a) Distribution of a Preisach weight function with zero-elements on the diagonal.



(b) The calculated inverse input.



(c) Zoomed in on the worst region of the calculated inverse input. The inverted input is discontinuous at the minimum.

Figure 7.23: Example of the consequence of identification with a Preisach model containing zero-elements on the diagonal. Continuous example.

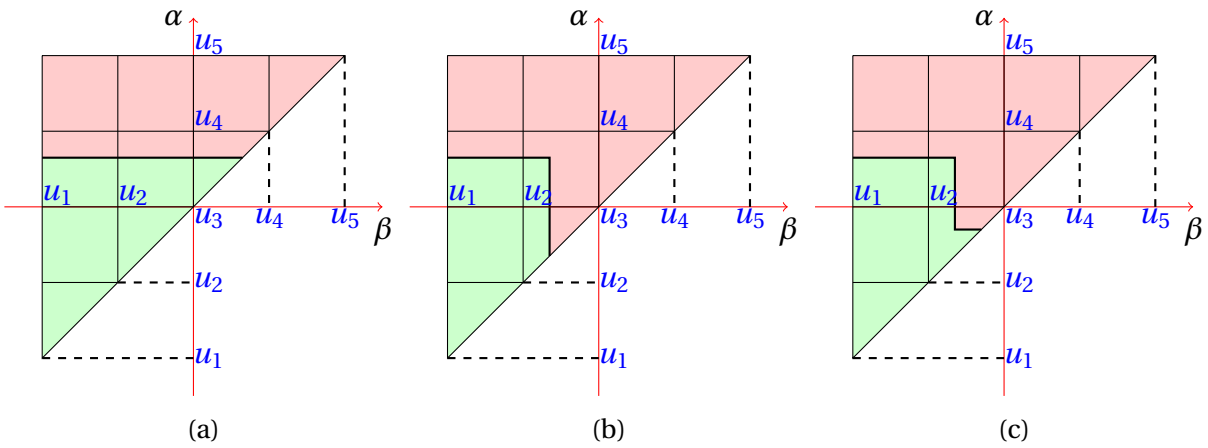


Figure 7.24: An example of the change of the Preisach plane. The weight on the diagonal element between input value u_2 and u_3 is assumed zero.

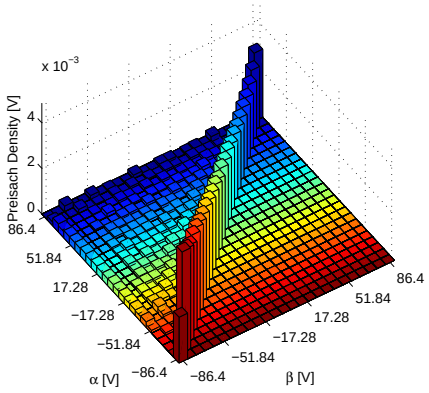
positive to negative or opposite.

The reason for this is best explained by an example. Assume that the Preisach plane corresponding to the maximum in Figure 7.23c is presented in Figure 7.24a. Furthermore, when the input decreases, the corresponding memory curve moves as shown in Figure 7.24b, where the diagonal element the curve passes through has weight zero. All the negative contribution comes from the other elements where the memory curve changes. Then, at some time, the input switches direction, and the current memory curve is in a diagonal element with weight zero. A small increase in the input, shown in Figure 7.24c, will give no effect on the output. Hence, the input has to increase to the next Preisach element in order to change the output. This caused the discontinuity shown in Figure 7.23c.

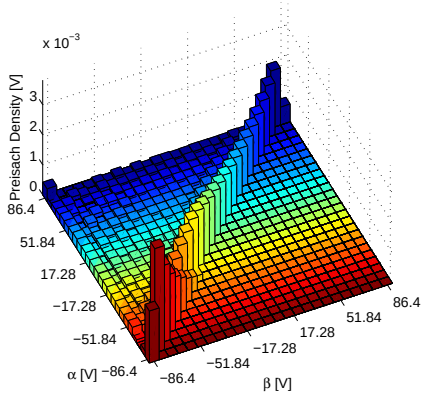
The example shown here corresponds to the continuous model, however, zero-elements on the diagonal will also affect the discrete model. In the previous section, the inversion based on both a large-amplitude sine wave and the PRBN signal suffered due to this problem. In this case the algorithm causes the input to switch, since an increase or decrease will not result in a changed input signal.

7.4.2 Effect of Low Pass Filtering

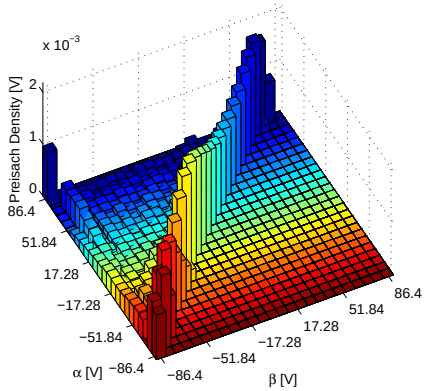
In this section, the effect of low pass filtering is investigated for identification of the Preisach models. As implied in the previous sections, negative phase from low pass filtering influences the output of the identified model. First, the distributions of the Preisach weight functions for various input frequencies are illustrated, when the signal loop contains a low pass



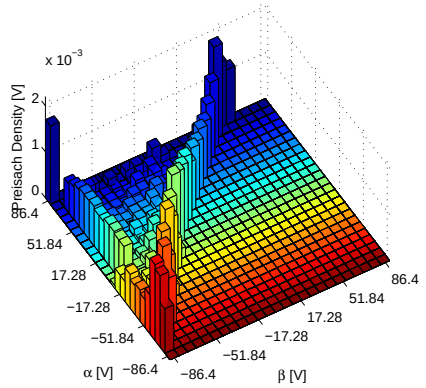
(a) 20 Hz triangular signal, not low pass filtered.



(b) 20 Hz triangular signal, low pass filtered.



(c) 40 Hz triangular signal, low pass filtered.



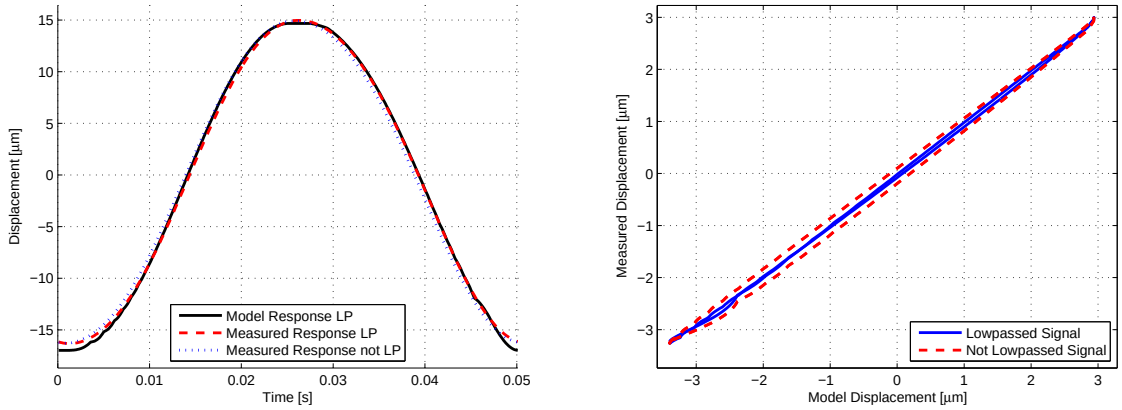
(d) 100 Hz triangular signal, low pass filtered.

Figure 7.25: Comparison of the identified Preisach density function for low pass filtered signals. The low pass filter used had a cutoff frequency on 500 Hz. Example with $n_h = 25$ as discretization level.

filter with cutoff frequency 500 Hz as an anti-aliasing filter. Further, three signals with frequency 20, 40 and 100 Hz will be used as references to the same model for inversion.

Preisach Distribution

Figure 7.25 presents four different Preisach weights, where the first distribution is identified with a 20 Hz triangular signal without a low pass filter in the signal loop. The three following distributions results from using a low pass filter with 500 Hz as cutoff frequency, when the identified signal has 20, 40 and 100 Hz frequency respectively. As seen, the Preisach plane clearly changes shape due to the low pass filter. While Figure 7.28a has a flat diagonal, the models resulting from low pass filtered signals have changed structure significantly, especially when the input frequency increases.



(a) Model and measured output.

(b) Model-measurement relationship.

Figure 7.26: Comparison of a model based on a low pass filtered signal and a model based on a signal which was not low pass filtered. Discretization level $n_h = 25$.

The main reason for this, as implied earlier, is the negative phase introduced by the low pass filter. If one assumes that the distribution given in Figure 7.25a is the correct distribution, which is a good assumption, since the identification was done without any negative phase effects in the signal loop, it is possible to say that the model loses information about the hysteresis as the input frequency and phase lag increases. This happens because the model tries to encompass both the hysteresis effect and the effect from the negative phase. Since the Preisach model is rate-independent, a different frequency on the input signal will give a different distribution of the Preisach plane.

The last two distributions in Figure 7.25, have zero-elements on the diagonal, which again invalidates the continuity properties of the inverse, shown in the previous section. Consequently, low pass filtering of signals should be avoided when identifying the Preisach model.

Model Behavior

In order to further study the effect of low pass filtering, a model based on the 20 Hz low pass filtered signal was used to create a continuous model with discretization level $n_h = 25$. This model was then used to compare the effects from the 20 Hz signal that was not low pass filtered, and the low pass filtered 40 Hz and 100 Hz signals.

First, the two 20 Hz signals were compared. Then the measurement from the 20 Hz signal, that was not low pass filtered, were compared to the model output, illustrated in Figure 7.26. If studied closely, the measured response from the 20 Hz signal that was not low pass

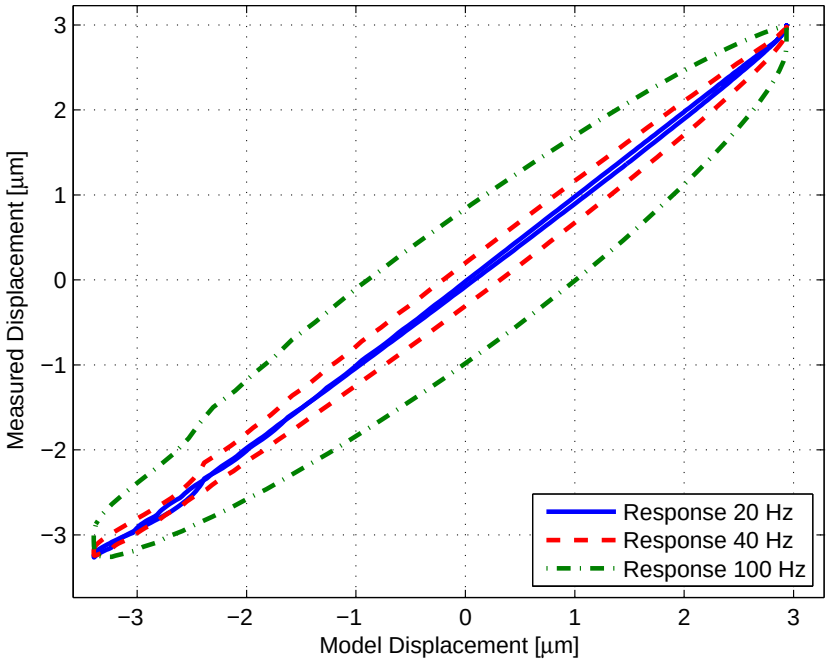
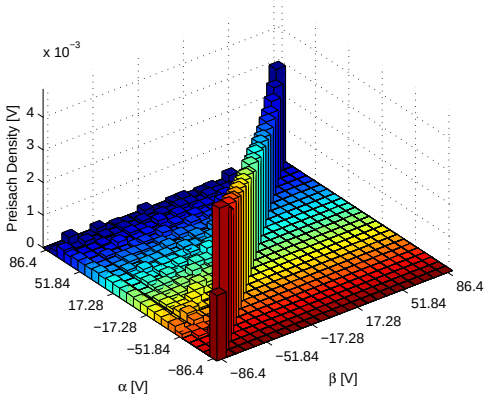


Figure 7.27: Comparison of three low pass filtered signals, where the model is based on a 20 Hz input signal. Discretization level of models $n_h = 25$.

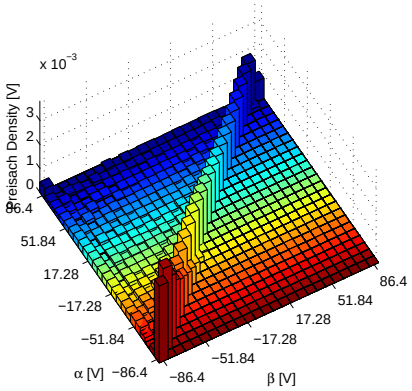
filtered can be seen to lie before the model output. This is also seen in Figure 7.26b, where the displacement from the model and measurement is plotted together. The model based on the low pass filtered signal, the blue line, almost compensates for all the lagging effect, which shows that the model can be identified, despite having some low pass filtered effects in the signal loop. However, the model does not fit when the input has not been low pass filtered. This is shown by the red dashed line in the Figure 7.26b.

Comparing the measurements from the 40 Hz and 100 Hz signals with the model output, as in Figure 7.27, it can easily be seen that a higher frequency also increases the lagging effect. This was of course expected based on negative phase effects of low pass filters, described in the previous chapter.

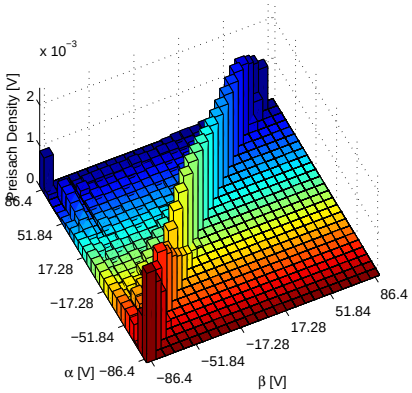
To conclude, low pass filters can severely limit the usability of the Preisach model. In fact, it can make the model impossible to invert without oscillations or discontinuities, since the responding Preisach weight function includes zero-elements on the diagonal. A model based on a low pass filtered signal is only valid for inputs of the same frequency and low pass filter. If possible, low pass filters with cutoff frequencies low enough to introduce phase lag should not be used in systems where hysteresis is being identified.



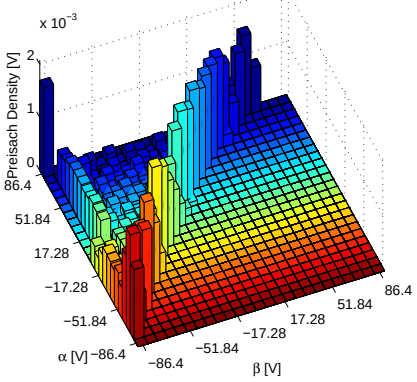
(a) 20 Hz triangular signal, not time delayed.



(b) 20 Hz triangular signal, time delayed.



(c) 40 Hz triangular signal, time delayed.



(d) 100 Hz triangular signal, time delayed.

Figure 7.28: Comparison of the identified Preisach density function for time delayed signals, with $4.58e^{-4}$ s delay. The identified continuous models has discretization level $n_h = 25$ and were calculated by an triangular input signal with 50 reversals with the frequencies mentioned in the subcaptions.

7.4.3 Effect of Time Delay

In order to study the effect of time delayed signals on the Preisach plane, the measurement of the displacement was delayed with $\tau = 4.58e^{-4}$ s, corresponding to the time delay reported in Ragazzon et al. [2014]. Since the aim is to study the effect of time delay on the identification, no effort was made to determine if there exist any time delay in the instrumentation for this experimental setup. However, the capacitance based sensors are known to exhibit some time delay, therefore this effect should be studied more closely in the future.

As described in Chapter 6, time delay introduces a negative phase to the system in the same manner as a low pass filter. The effect of time delay on the Preisach plane, can be seen in Figure 7.28 where the distributions for four signals are shown, one without any time delay, and three with. The Preisach plane can be seen to be affected in the same way as for low pass

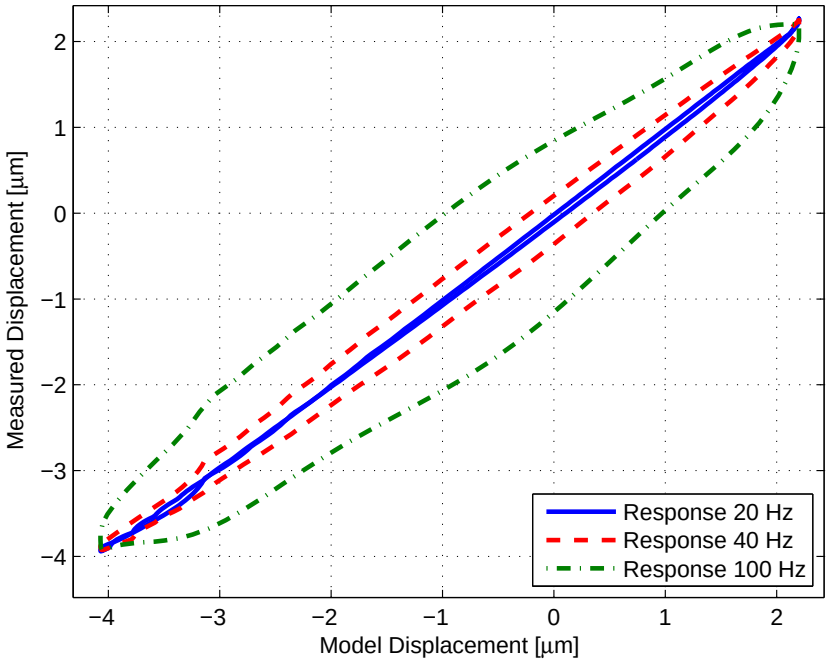
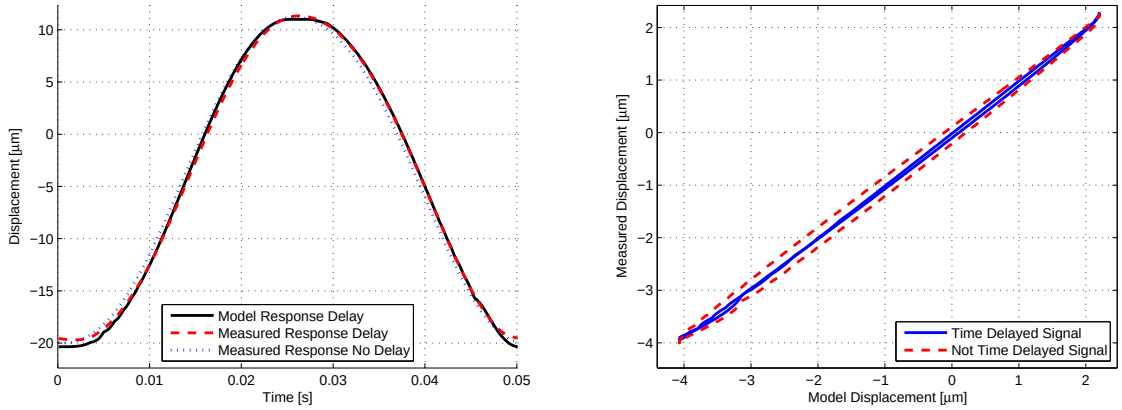


Figure 7.29: Comparison of three time delayed signals, where the model is based on a 20 Hz input signal. Discretization level of models $n_h = 25$.

filtered signals. For high frequencies the diagonal elements become zero, which will make the inverse model invalid.

The distribution in Figure 7.28a, should in theory be equal to Figure 7.25a, since both signals are neither low pass filtered or time delayed. By studying them, this also seems to be the case. More surprisingly is the fact that the distributions in Figure 7.28b - 7.28d are very similar to the corresponding distributions in Figure 7.25, since the introduced negative phase from the time delay should be considerably less than the low pass filter introduces. This was also shown in the previous chapter, where the negative phase for the low pass filter at 100 Hz was found to be -16 degrees, while the same frequency for the time delay should give -2.7 degrees. Thus, the phase lag in the time delay case should be considerably less than for the low pass filter case.

The results from the Preisach distributions are supported by the findings in Figure 7.29, where the phase lag is shown to be approximately the same as in the low pass filter case. This was confirmed by repeated experiments, where the low pass filter was set to 25 kHz, which is assumed to not influence the measurements with negative phase. It is possible that other effects in the system also introduces negative phase, however, these effects should then influence the low pass filtered signals as well. The time delayed effect on the identification



(a) Model and measured output.

(b) Model-measurement relationship.

Figure 7.30: Comparison of a model based on a time delayed signal and a model based on a signal which was not time delayed. Discretization level $n_h = 25$.

should, consequently, be studied more closely in the future.

Lastly, the model fit of the time delayed signal was compared to the same signal without any time delay. This corresponds to the results for the low pass filter in Figure 7.26. As seen in Figure 7.30, the model seems to be able to describe a small time delay. However when the model is found by this time delayed signal, an input without time delay with the same frequency, produces an output which is not sufficiently described by the model. This is illustrated in Figure 7.30a, where the measured response without time delay lies in front of the time delayed signal.

7.5 System Description

This brief section presents the model-measurement relationship where the hysteresis model is identified with an increasing sine wave of 20 Hz. The output of this model is compared to the measured displacement, in order to see if the effect of an increased frequency on the system without any low pass filtering or time delay. The results are described in Figure 7.31, where both the 20 Hz and 40 Hz measurement fit the model nicely, however for the 100 Hz signal, the measurement can be seen to lag behind. This is probably due to the negative phase in the actuator, and perhaps also some effects from the 25 kHz reconstruction and anti-aliasing filters, or time delay in the sensor. However, the effects are very small compared to the model-measurement relationships shown for low pass filtering and time delay. As mentioned in the previous section, these effects in the experimental setup should be ex-

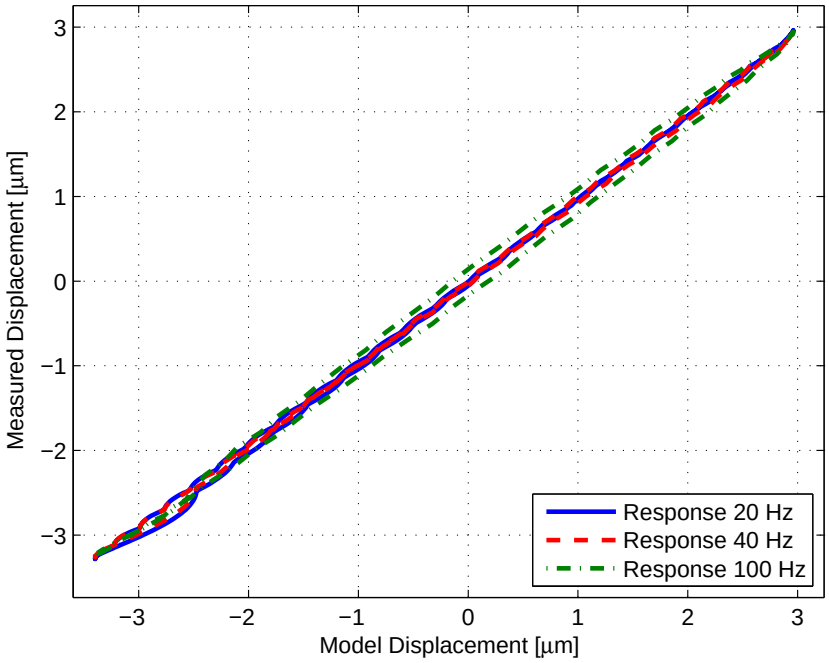


Figure 7.31: Three different frequencies applied to the system. The measured output is compared to the model output from the 20 Hz signal.

plored mored throughly in the future.

Chapter 8

Concluding Remarks

8.1 Conclusion

The main theoretical results from the specialization project were presented. This includes a description of the nanopositioning problem, a review of common sensors and actuators and control issues related to this. Several hysteresis models were investigated, mainly focusing on Preisach-type operators. Common identification methods used for hysteresis modeling were presented, together with a proposed implementation strategy.

A discrete and a continuous method for describing the output of the Preisach operator was introduced. A detailed review of their differences was presented, comparing the output calculation based on the memory curve and the Preisach plane. Aspects concerning implementation of these models were illustrated.

Furthermore, an algorithm for computing the output of the continuous model was proposed, based on calculating the areas of the memory curve elements. No optimization properties of the algorithm were given, but it was indicated that there is a potential for reducing the run time drastically. In addition, the output of the continuous model was shown to be piecewise continuous, based on an argument of the areas in the square and triangular elements.

The general approach for hysteresis compensation was presented, consisting of approximating a right inverse of the hysteresis model. A few inversion algorithms were briefly introduced, and the closest match algorithm was chosen as the inversion procedure, due to the property of yielding an input which minimizes the difference between reference and model output. Continuity and existence properties for the closest match algorithm were shown,

followed by a detailed description of the implementation of the closest match algorithm for both the discrete and inverse method.

To design input signals which yield a maximum amount of information, experimental design was examined, presenting the factors for a well designed experiment in the general case. PE for hysteresis identification was studied. In contrast to linear systems, where the amount of frequency components characterize the PE property, the number of input reversals gives this property for systems exhibiting hysteresis. Consequently, the input signals were constructed to fulfill this. In addition, the properties of a well designed experiment, demonstrating hysteresis, was given.

Effects from instruments in the signal loop affecting the hysteresis identification are reviewed. Such effects include creep, dynamics, low pass filtering and time delay. Furthermore, the effect of negative phase was studied, focusing on linear phase lag that arises due to this.

Both the discrete and continuous method was shown to describe hysteresis. For the discrete model, an increased discretization level significantly improved the description of the hysteresis nonlinearity. A lower discretization level, compared to the discrete model, described the phenomenon for the continuous model. During identification it was shown that it is important to have an input range that corresponds to the range of the identified model. Both a larger or smaller input range, compared to the range of the identified model, could decrease the accuracy of the resulting Preisach weight function.

The inversion algorithms were verified by simulations, and used for compensation of hysteresis. To compensate with the use of the discrete model, the inverted input was smoothed by linear interpolation. Interpolating with query points in the left corners of the discrete input gave the best results. For the continuous model, the inverted signal consisted of higher frequency components compared to the discrete model. Thus, there are both cons and pros with both models, in real time applications an optimal linear interpolation method could be challenging to implement. However, the high frequency components of the continuous inversion introduces vibrations in the piezoelectric actuator.

Testing of different input signals for identification was conducted, reveling a significant improvement when using inputs with a sufficient amount of input reversals. Both sine and triangular reversed input signals were tested, indicating that the triangular signal could perhaps give a slightly better result.

Non-optimal signals and errors in the signal loop caused the identified models to become invalid. If the Preisach weight function has zero-elements on the diagonal, both the discrete and continuous inverse will become invalid due to oscillations and discontinuities, respectively. This introduces high frequencies in the input signal exciting the vibration dynamics in the actuator.

Both low pass filters and time delay was shown to introduce frequency dependent phase lag, influencing the measurement used for identification. The Preisach model cannot encompass such rate dependent properties. Thus, an input signal with a different frequency than used in the identification, will show a lagging effect on the output. The time delay effect, during repeated test, was shown to be approximately the same as for low pass filtering. The reason for this is not obvious, since the theory clearly states that the lagging effect from low pass filters should surpass the effect from time delay.

8.2 Further Work

8.2.1 Creep Compensation

The reference signals used in this work had a lower limit of 5 Hz. In this frequency range it is possible that creep influences the phase of the measurements. Consequently a creep compensation could show if the performance of hysteresis compensation also increases due to this.

8.2.2 Output of Continuous Model

The calculations of the output of the continuous model is computationally expensive, consequently work should be done to reduce this complexity. Now, all Preisach element is calculated for each time step, however it is only needed for updating the elements where there has been changes. Doing this in some optimal manner, maybe by dynamical programming could greatly improve the computational complexity, opening the way for real time implementations of this method.

8.2.3 Inverse Algorithms

Study extensions for the proposed inversion algorithms which may be able to reduce the effect of a diagonal with zero-elements. This should be especially applicable for the discrete method, since it experiences this issue for a monotonically increasing reference signal, and not just when the reference is reversed, as in the continuous case.

Additionally, testing the performance of the pseudo compensation routine, briefly described in Chapter 4, and comparing the performance to the results shown for the numerical inversion of the Preisach operator could be of interest. This method only requires one identification procedure, and no inversion, since it directly calculates the inverted output. Consequently this model should be simpler to implement than the methods shown here.

8.2.4 Interpolation

Interpolation of the discrete input signals was shown to greatly improve the hysteresis compensation performance compared to direct use of the input function. However, the method proposed are not the best suited for real-time applications. Interpolation strategies for both off-line and on-line applications should therefore be studied further. In this work only linear interpolation was used, however other interpolation methods increasing the continuity properties of the input signal can be better suited.

8.2.5 Input Signal Design

A more detailed study should be conducted on the design of input signal for identification. This should include a thorough study of the difference between the increasing sine and increasing triangular signal. Does the triangular signal achieve a better accuracy, or was the results in this thesis just coincidence. Moreover, the number of reversals of the input signal should be studied more closely with experiments. This should include both a smaller and larger amount of reversals than was presented in this work.

8.2.6 Modified Continuous Model

Study if it is possible to create a modified continuous model based on square cells on the diagonal. This will increase the continuity properties of the continuous inverse. However, when square cells are used on the diagonal, an issue arises when the input to the model

is reversed, since the memory curve will experience a discontinuous step. If this could be circumvented, perhaps by interpolation, the resulting model could be able to increase the accuracy of continuous inversion since the high frequency components are reduced.

8.2.7 Time Delay Effects

The effect of time delay on hysteresis identification should be studied further, since the results presented here show a similar effect from time delay as for low pass filtering through repeated testing. Based on the theory presented, low pass filtering should give a significantly larger phase lag than time delay.

8.2.8 Prandtl-Ishlinskii Operator for Inversion

The PI operator has been showed to have an analytical inverse, consequently, the performance of this model should be compared to the Preisach model. There are no proper study on the performance of Preisach type operators compared to other hysteresis compensation schemes.

8.2.9 On-line Implementation

If this should be used for nanopositioning applications, the hysteresis compensation scheme should be implemented on-line. To manage this, the computational complexity has to be reduced in some manner. Testing both implementational aspects and performance of the compensation method on on-line applications would be of great interest.

8.2.10 Level of Discretization

For the discrete model, both a discretization level of $n_h = 100$ and $n_h = 50$ gave satisfactory results. A question for the future is to study how low n_h it is possible to use, and at the same time achieve a good accuracy when compensating.

Appendix A

Contents of Attached Zip-file

The delivered files, which can be found in the zip attachment, is mainly grouped in two folders; one for discrete identification and inversion and one for continuous identification and inversion. A proposed execution order of the files is to first run the identification script, identifying a Preisach weight function, followed by running the inversion script based on this identification data. In addition, a file folder for generation input signals for testing in the dSPACE environment is appended.

A short explanation of the main files is given below.

A.1 File Description

A.1.1 Matlab Files

area_operator.m Calculate the are of each memory curve cell.

calc_preisach_output_cont.m Main file for calculating the Preisach output, runs several other functions.

find_coor_vector.m Find coordinates of corners in memory curve.

find_edge_vector.m Find all edges of the corners in memory curve.

generate_input_signals.m Generate signals for use in identification with dSPACE.

get_minmax_vector.m Find past minimums and maximums of input.

get_OnOff_vector.m Find states of all Preisach elements.

id_model_cont.m Identification of continuous model.

id_models.m Script for running discrete identification.

id_models_cont.m Script for running continuous identification.

invert_cont_ref.m Script for running continuous inversion.

invert_discrete_ref.m Script for running discrete inversion.

invert_preisach_cont.m Continuous closest match algorithm.

invert_preisach_disc.m Discrete closest match algorithm.

preisach_id_cont.m Function used in continuous identification.

preisach_id_cont.m Identify the bulk contribution η_0.

relay_operator.m Output of delayed relay operator.

A.1.2 Data Files

no_lp_29.mat Matlab data file for increasing sine signal. Measured from dSPACE setup. Usen in continuous identification.

mu_rising_sine_20hz_100_disc_N50.mat Model data used in inversion for the discrete model. Generated by id_models.m.

mudata_20hz.mat Model data used in inversion for the continuous model. Generated by id_models_cont.m.

rising_sine_20hz_100.mat Matlab data file for increasing sine signal. Measured from dSPACE setup. Usen in discrete identification.

A.1.3 Simulink File

dspace_test.mdl Simulink program for dSPACE setup.

Appendix B

Matlab Code

This appendix includes the implemented algorithms described in Algorithm 1 - 4. See the zip-file attachment for additional files.

B.1 relay_operator.m

```
function [ y,S ] = relay_operator( alpha,beta,s,x,h )
%RELAY_OPERATOR Output of a delayed relay operator
switch s
    case -1
        if x > alpha + (h/2)
            y =1;
            S =1;
        else
            y = -1;
            S = -1;
        end
    case 1
        if x < beta - (h/2)
            y = -1;
            S = -1;
        else
            y = 1;
            S = 1;
        end
end
```

B.2 area_operator.m

```
function [ area_vector ] = area_operator( edge_vector, minmax_coor, ...
    pgrid )
%AREA_OPERATOR Calculates the areas of all memory curve cells
h = (pgrid(2)-pgrid(1))/2;
k = 2;
```

```
area_vector = zeros(1,length(edge_vector(1,:)));
for i=1:length(edge_vector(1,:))
    max_y = pgrid(edge_vector(2,i)) + h;
    min_y = pgrid(edge_vector(2,i)) - h;
    max_x = pgrid(edge_vector(1,i)) + h;
    min_x = pgrid(edge_vector(1,i)) - h;
    A_plus = 0;
    A_minus = 0;

    %First if the relay element is triangluar, that is the last in the edge
    %vector:
    if i== length(edge_vector(1,:))
        %If edges in the triangle:
        while (k <= length(minmax_coor(1,:))-1 && ...
            inpolygon(minmax_coor(1,k),minmax_coor(2,k), ...
            [min_x,min_x,max_x], [min_y,max_y,max_y]))
            x_k_1 = minmax_coor(1,k-1);
            x_k = minmax_coor(1,k);
            y_k_1 = minmax_coor(2,k-1);
            y_k = minmax_coor(2,k);

            if y_k < y_k_1
                width = x_k - (pgrid(edge_vector(1,i))-h); %can replace ...
                    pgrid(..)-h with min_x
                height = min(y_k_1 - y_k, pgrid(edge_vector(2,i))+h - ...
                    y_k); %can replace pgrid(..)+h with max_y
                A_plus = A_plus + width*height;
            elseif x_k > x_k_1
                width = min(x_k-x_k_1,x_k-(pgrid(edge_vector(1,i))-h)); ...
                    %can replace pgrid(..)-h with min_x
                height = pgrid(edge_vector(2,i))+h-y_k;
                A_minus = A_minus + width*height;
            end
            k = k+1;
        end
        if k < length(minmax_coor(1,:))
            error('Not all corners detected in area_operator');
        end
        x_k_1 = minmax_coor(1,k-1);
        x_k = minmax_coor(1,k);
        y_k_1 = minmax_coor(2,k-1);
        y_k = minmax_coor(2,k);

        if x_k > x_k_1
            height = pgrid(edge_vector(2,i))+h - y_k; %can replace ...
                pgrid(..)+h with max_y
            width = min(x_k-x_k_1,x_k - (pgrid(edge_vector(1,i))-h)); ...
                %can replace pgrid(..)-h with min_x
            A_minus = A_minus + height*width;
            A_minus = A_minus + (height*height)/2;
            height = y_k - (pgrid(edge_vector(2,i))-h); %can replace ...
                pgrid(..)-h with min_y
            A_plus = A_plus + (height*height)/2;
        elseif y_k < y_k_1
            width = x_k - (pgrid(edge_vector(1,i))-h); %can replace ...
                pgrid(..)-h with min_x
            height = min(pgrid(edge_vector(2,i))+h-y_k, y_k_1-y_k); ...
                %can replace pgrid(..)+h with max_y
```

```
53              A_plus = A_plus + width*height;
54              A_plus = A_plus + (width*width)/2;
55              width = pgrid(edge_vector(1,i))+h - x_k;%can replace ...
                    pgrid(..)+h with max_x
56              A_minus = A_minus + (width*width)/2;
57          else %the point is in the lower or higher corner
58              if x_k < pgrid(1)
59                  A_minus = A_minus + ((2*h)^2)/2;
60              else
61                  A_plus = A_plus + ((2*h)^2)/2;
62              end
63          end
64      else %The preisach element is a square
65          iscorner = 0;
66          while (k <= (length(minmax_coor(1,:))-1) && max_y >= ...
                minmax_coor(2,k) && min_y <= minmax_coor(2,k) && max_x >= ...
                minmax_coor(1,k) && min_x <= minmax_coor(1,k))
67              iscorner = 1;
68              x_k_1 = minmax_coor(1,k-1);
69              x_k = minmax_coor(1,k);
70              y_k_1 = minmax_coor(2,k-1);
71              y_k = minmax_coor(2,k);
72
73              if y_k < y_k_1
74                  if k > 2 && minmax_coor(2,k-2) < max_y
75                      width = min(x_k - (pgrid(edge_vector(1,i))-h),x_k_1 ...
                            - minmax_coor(1,k-2));
76                  else
77                      width = x_k - (pgrid(edge_vector(1,i))-h);
78                  end
79                  height = min(2*h,y_k_1-(pgrid(edge_vector(2,i))-h));
80                  A_plus = A_plus + width*height;
81              elseif x_k > x_k_1
82                  width = min(2*h, (pgrid(edge_vector(1,i))+h) - x_k_1);
83                  if k > 2 && minmax_coor(1,k-2) > min_x
84                      height = min(pgrid(edge_vector(2,i)) +h ...
                            -y_k_1,minmax_coor(2,k-2) -y_k_1);
85                  else
86                      height = pgrid(edge_vector(2,i))+h-y_k_1;
87                  end
88                  A_minus = A_minus + width*height;
89              end
90              k = k+1;
91          end
92          x_k_1 = minmax_coor(1,k-1);
93          x_k = minmax_coor(1,k);
94          y_k_1 = minmax_coor(2,k-1);
95          y_k = minmax_coor(2,k);
96
97          if iscorner
98              if y_k < y_k_1
99                  height = y_k_1 - (pgrid(edge_vector(2,i))-h);
100                 width = min(x_k_1 - (pgrid(edge_vector(1,i))-h),x_k_1 - ...
                        minmax_coor(1,k-2));
101                 A_plus = A_plus + height*width;
102                 A_minus = A_minus + height*(2*h-(x_k_1 - ...
                        (pgrid(edge_vector(1,i))-h)));
103             elseif x_k > x_k_1
```

```
104                 width = pgrid(edge_vector(1,i))+h - x_k_1;
105                 height = min(pgrid(edge_vector(2,i))+h - y_k_1, ...
                        minmax_coor(2,k-2) - y_k_1);
106                 A_minus = A_minus + width*height;
107                 A_plus = A_plus + width*(2*h ...
                        -(pgrid(edge_vector(2,i))+h - y_k_1));
108             end
109         elseif iscorner ==0
110             if y_k < y_k_1
111                 height = 2*h;
112                 width = x_k - (pgrid(edge_vector(1,i))-h);
113                 A_plus = A_plus + height*width;
114                 A_minus = A_minus + height*(2*h-width);
115             elseif x_k > x_k_1
116                 height = y_k - (pgrid(edge_vector(2,i))-h);
117                 width = 2*h;
118                 A_plus = A_plus + height*width;
119                 A_minus = A_minus + (2*h-height)*width;
120             else
121                 if minmax_coor(end) == pgrid(1);
122                     A_plus = 0;
123                     A_minus = ((2*h)^2);
124                 else
125                     A_minus = 0;
126                     A_plus = ((2*h)^2);
127                 end
128             end
129         end
130     end
131     if i==length(edge_vector(1,:))
132         A_plus = A_plus/(((2*h)^2)/2);
133         A_minus = A_minus/(((2*h)^2)/2);
134 
135     else
136         A_plus = A_plus/((2*h)^2);
137         A_minus = A_minus/((2*h)^2);
138     end
139     summing = A_plus+A_minus;
140 
141     %error checking:
142 
143     if summing < 0.99 && summing > 1.01
144         msg = ['A_plus+A_minus = ', num2str(A_plus+A_minus), ', i = ', ...
                num2str(i)];
145         disp(msg);
146         error('A_plus + A_minus is not equal to 1');
147     end
148 
149     area_vector(i) = A_plus - A_minus;
150 end
151 end
```

B.3 invert_preisach_disc.m

```
1 function [ inv_input,y_model_out ] = ...
      invert_preisach_disc(desired_output,pgrid,isleft, A, mu, d)
```

```
%Calculates the inverted signal based on a reference signal.


h = (pgrid(2)-pgrid(1));
N = length(pgrid);
Q = (N*(N+1))/2;

global u_glob u_prev_glob first_5 OnOff_vector_prev OnOff_matrix_prev ...
    minimum maximum prev_desired

if isempty(first_5)
    first_5 = 1;

    u_glob = pgrid(1);
    u_prev_glob = NaN;
    OnOff_vector_prev = -1*ones(1,Q);
    OnOff_matrix_prev = zeros(N,N);
    minimum = [];
    maximum = [];
    prev_desired = -100;

    %check if the identified weighs are valis are valid:
    distribution = zeros(N,N);
    k = 0;
    for i = N:-1:1
        for j = i:-1:1
            k = k+1;
            distribution(i,j)= mu(k);
        end
    end
    check_var = 0;
    for i = 1:N
        if (distribution(i,i) == 0)
            check_var = 1;
        end
    end
    if check_var == 1
        warning('The distribution of the identified mu contains zero ...
            diagonal elements. This will result in a bad inverse input! ...
            you SHOULD use a better model for inverting a signal. This ...
            model is NOT good enough.');
    end
end

[minmax_coor, isleft, minimum, maximum] = ...
    get_minmax_vector(u_glob,u_prev_glob,pgrid,h,isleft, minimum, maximum);
u = minmax_coor(1,end);
[y, OnOff_vector_prev, OnOff_matrix_prev] = ...
    calc_preisach_output_cont(minmax_coor, minimum, maximum, isleft, ...
    pgrid, A, mu, d, OnOff_vector_prev, OnOff_matrix_prev);

while 1 %NOT an infinite loop, when the closest match is found, it ...
    returns out of this function.
    if (y == desired_output)
        inv_input = u;
        y_model_out = y;
        return;
    elseif (y < desired_output)
```

```
52
53          lower = 1;
54          while(lower)
55              if u >= pgrid(end)+h/2
56                  inv_input = pgrid(end)+h/2;
57                  y_model_out = y;
58                  return;
59              else
60                  u_past = u;
61                  u = u + h;
62                  minimum_past = minimum;
63                  maximum_past = maximum;
64                  [minmax_coor, isleft, minimum, maximum] = ...
                        get_minmax_vector(u,u_past,pgrid,h,isleft,minimum,maximum);
65                  y_past = y;
66                  [y, OnOff_vector_prev, OnOff_matrix_prev] = ...
                        calc_preisach_output_cont(minmax_coor, minimum, ...
                        maximum, isleft, pgrid, A, mu, d, OnOff_vector_prev, ...
                        OnOff_matrix_prev);
67                  if (y == desired_output)
68                      inv_input = u;
69                      y_model_out = y;
70                      u_glob = u;
71                      u_prev_glob = u_past;
72                      return;
73                  elseif (y < desired_output)
74                      %run the while loop once again
75                      u_glob = u;
76                      u_prev_glob = u_past;
77                  else
78                      lower = 0;
79                  end
80              end
81          end
82          y_diff = abs(y-desired_output);
83          y_past_diff = abs(y_past-desired_output);
84          if( y_diff < y_past_diff)
85              u_glob = u;
86              u_prev_glob = u_past;
87          else
88              inv_input = u_past;
89              y_model_out = y_past;
90              %u_glob and u_prev_glob is the same as before.
91              minimum = minimum_past;
92              maximum = maximum_past;
93              return;
94          end
95      elseif (y > desired_output);
96          higher = 1;
97          while(higher)
98              if u <= pgrid(1)-h/2;
99                  inv_input = pgrid(end)-h/2;
100                 y_model_out = y;
101                 return;
102             else
103                 u_past = u;
104                 u = u-h;
105                 minimum_past = minimum;
```

```
106                 maximum_past = maximum;
107                 [minmax_coor, isleft, minimum, maximum] = ...
                        get_minmax_vector(u,u_past,pgrid,h,isleft,minimum,maximum);
108                 y_past = y;
109                 [y, OnOff_vector_prev, OnOff_matrix_prev] = ...
                        calc_preisach_output_cont(minmax_coor, minimum, ...
                        maximum, isleft, pgrid, A, mu, d, OnOff_vector_prev, ...
                        OnOff_matrix_prev);
110                 if (y==desired_output)
111                     inv_input = u;
112                     y_model_out = y;
113                     u_glob = u;
114                     u_prev_glob = u_past;
115                     return;
116                 elseif (y > desired_output)
117                     u_glob = u;
118                     u_prev_glob = u_past;
119                     %run the while loop once more
120                 else
121                     higher = 0;
122                 end
123             end
124         end
125
126         y_diff = abs(y-desired_output);
127         y_past_diff = abs(y_past-desired_output);
128         if( y_diff < y_past_diff)
129             u_glob = u;
130             u_prev_glob = u_past;
131         else
132             inv_input = u_past;
133             y_model_out = y_past;
134             minimum = minimum_past;
135             maximum = maximum_past;
136             return;
137         end
138     end
139 end
140 end
```

B.4 invert_preisach_cont.m

```
1  function [ v,u_prev ] = invert_presiach_cont( ...
       desired_output,pgrid,isleft, A, mu, d,d_scaling)
2  %INVERT_PRESIACH_CONT Continuous inverse based on a desired output Closest
3  %match algorithm
4
5  %REMARK:
6  %This inversion seems to work well for a desired output which corresponds
7  %to a input value inside the input range, but gives error when the desired
8  %output exceeds this level. If time, this should be fixed in order to have
9  %more robust code. However, the inversion works splendidly if the
10 %corresponding input is in the range.
11
12 h = (pgrid(2)-pgrid(1))/2;
13 N = length(pgrid);
```

```
14 Q = (N*(N+1))/2;
15
16 global x x_prev first_3 OnOff_vector_prev OnOff_matrix_prev minimum ...
       maximum prev_desired
17
18 if isempty(first_3)
19     x = 0;
20     x_prev = NaN;
21     first_3 = 1;
22     OnOff_vector_prev = zeros(1,Q);
23     OnOff_matrix_prev = zeros(N,N);
24     minimum = [];
25     maximum = [];
26     prev_desired = -100;
27
28     distribution = zeros(N,N);
29     k = 0;
30     for i = N:-1:1
31         for j = i:-1:1
32             k = k+1;
33             distribution(i,j)= mu(k);
34         end
35     end
36     check_var = 0;
37     for i = 1:N
38         if (distribution(i,i) == 0)
39             check_var = 1;
40         end
41     end
42     if check_var == 1
43         warning('The distribution of the identified mu contains zero ...
               diagonal elements. This will result in a bad inverse input! ...
               you SHOULD use a better model for inverting a signal. This ...
               model is NOT good enough.');
44     end
45
46     if d_scaling <= 0 || d_scaling >=1
47         error('d_scaling should be larger than zero and less than 1');
48     end
49 end
50
51 diff = x - x_prev;
52 [minmax_coor, isleft, minimum, maximum] = ...
       get_minmax_vector(x,x_prev,pgrid,h,isleft, minimum, maximum);
53 v_k = minmax_coor(1,end);
54 [u_prev, OnOff_vector_prev, OnOff_matrix_prev] = ...
       calc_preisach_output_cont(minmax_coor, minimum, maximum, isleft, ...
       pgrid, A, mu, d, OnOff_vector_prev, OnOff_matrix_prev);
55 num_of_iter = 0;
56
57 if (desired_output == prev_desired) %if the same as in the previous ...
       case, then use the previous input.
58     v = x;
59 else
60     iteration_cond = 1;
61     while iteration_cond
62         % %Changes in the iterations, but must be set equal to the ...
               current state at
```

```
63          % %the start:
64          OnOff_vector_iteration = OnOff_vector_prev;
65          OnOff_matrix_iteration = OnOff_matrix_prev;
66
67          coordinates = find_coor_vector(minmax_coor,pgrid);
68          if desired_output > u_prev
69              if (pgrid((coordinates(1,end))) + h) ~= v_k
70                  d1_k = (pgrid((coordinates(1,end))) + h) - v_k;
71              else
72                  d1_k = (pgrid((coordinates(1,end))+1) + h) - v_k;
73              end
74              if length(minmax_coor(2,:)) < 3
75                  d2_k = pgrid(end) + h - v_k;
76              else
77                  d2_k = minmax_coor(2,end-2) - v_k;
78              end
79
80              d_iter = d_scaling*min([d1_k,d2_k]);
81              isleft_iteration = isleft;
82              minimum_iteration = minimum;
83              maximum_iteration = maximum;
84              [minmax_coor_iteration, isleft_iteration, ...
                    minimum_iteration, maximum_iteration] = ...
                    get_minmax_vector(x + d_iter,x,pgrid,h,isleft_iteration, ...
                    minimum_iteration, maximum_iteration);
85              u_iteration = ...
                    calc_preisach_output_cont(minmax_coor_iteration, ...
                    minimum_iteration, maximum_iteration, isleft_iteration, ...
                    pgrid, A, mu, d, OnOff_vector_iteration, ...
                    OnOff_matrix_iteration);
86
87                  Output_diff = u_iteration - u_prev;
88              if (Output_diff == 0)
89
90                  warning('Output_diff is zero in invert_preisach_cont. ...
                        This is probable due to zero elements on the ...
                        triangle cells in the mu vector. Use a identified ...
                        model which dont have zero elements on the diagonal');
91                  Output_diff = 1e-10;
92              end
93              a1_k = 1;
94              run_again = 1;
95              while run_again
96                  a2_k = Output_diff/d_iter^2 - a1_k/d_iter;
97
98                  if a2_k >= 0
99                      run_again = 0;
100                 else
101                     a1_k = a1_k/2;
102                 end
103             end
104             Desired_diff = desired_output - u_prev;
105
106             d0_k = (-a1_k + sqrt(a1_k^2 - ...
                    4*a2_k*(-Desired_diff)))/(2*a2_k);
107             if d0_k < 0
108                 error('d0_k is less than zero for u_des > u_prev');
109             end
```

```
110
111             d_k = min([d0_k,d1_k,d2_k]);
112             v_k = v_k + d_k;
113
114         elseif desired_output < u_prev
115             if v_k ~= pgrid((coordinates(2,end))) - h
116                 d1_k = v_k - (pgrid((coordinates(2,end))) - h);
117             else
118                 d1_k = v_k - (pgrid((coordinates(2,end)-1)) - h);
119             end
120             if length(minmax_coor(2,:)) < 3
121                 d2_k = v_k - (pgrid(1) - h);
122             else
123                 d2_k = v_k - minmax_coor(1,end-2);
124             end
125
126             d_iter = d_scaling*min([d1_k,d2_k]);
127             isleft_iteration = isleft;
128             minimum_iteration = minimum;
129             maximum_iteration = maximum;
130             [minmax_coor_iteration, isleft_iteration, ...
                minimum_iteration, maximum_iteration] = ...
                get_minmax_vector(x - d_iter,x,pgrid,h,isleft_iteration, ...
                minimum_iteration, maximum_iteration);
131             u_iteration = ...
                calc_preisach_output_cont(minmax_coor_iteration, ...
                minimum_iteration, maximum_iteration, isleft_iteration, ...
                pgrid, A, mu, d, OnOff_vector_iteration, ...
                OnOff_matrix_iteration);
132
133             Output_diff = u_iteration - u_prev;
134             if Output_diff == 0
135                 warning('Output_diff is zero in invert_preisach_cont. ...
                    This is probable due to zero elements on the ...
                    triangle cells in the mu vector. Use a identified ...
                    model which dont have zero elements on the diagonal');
136                 Output_diff = -1e-10;
137             end
138             a1_k = -1;
139             run_again = 1;
140             while run_again
141                 a2_k = Output_diff/d_iter^2 - a1_k/d_iter;
142                 if a2_k <= 0
143                     run_again = 0;
144                 else
145                     a1_k = a1_k/2;
146                 end
147             end
148             Desired_diff = desired_output - u_prev;
149             a1_k = -1*a1_k;
150             a2_k = -1*a2_k;
151             Desired_diff = -1*Desired_diff;
152             d0_k = (-a1_k + sqrt(a1_k^2 - ...
                4*a2_k*(-Desired_diff)))/(2*a2_k);
153             if d0_k < 0
154                 error('d0_k is less than zero for u_des > u_prev');
155             end
156             d_k = min([d0_k,d1_k,d2_k]);
```

```
157             v_k = v_k - d_k;
158         end
159         x_prev = x;
160         [minmax_coor, isleft, minimum, maximum] = ...
               get_minmax_vector(v_k,x_prev,pgrid,h,isleft, minimum, maximum);
161         [u_prev, OnOff_vector_prev, OnOff_matrix_prev] = ...
               calc_preisach_output_cont(minmax_coor, minimum, maximum, ...
               isleft, pgrid, A, mu, d, OnOff_vector_prev, OnOff_matrix_prev);
162         x = v_k;
163         if d_k == d0_k || num_of_iter >100
164             iteration_cond = 0;
165         elseif num_of_iter > 100
166             iteration_cond = 0;
167             warning('Iteration count reached 100 and inversion ...
                   algorithm exits, the output might not be optimal');
168         else
169             num_of_iter = num_of_iter +1;
170         end
171     end
172     v = v_k;
173 end
174 prev_desired = desired_output;
175 end
```

Bibliography

Abramovitch, D., Andersson, S., Pao, L., and Schitter, G. (2007). A tutorial on the mechanisms, dynamics, and control of atomic force microscopes. pages 3488–3502, New York, NY.

Adly, A. b. and Abd-El-Hafiz, S. c. (1998). Using neural networks in the identification of preisach-type hysteresis models. *IEEE Transactions on Magnetics*, 34(3):629–635.

Al Janaideh, M., Rakheja, S., and Su, C.-Y. (2011). An analytical generalized prandtl-ishlinskii model inversion for hysteresis compensation in micropositioning control. *IEEE/ASME Transactions on Mechatronics*, 16(4):734–744.

Alexander, S., Hellemans, L., Marti, O., Schneir, J., Elings, V., Hansma, P., Longmire, M., and Gurley, J. (1989). An atomic-resolution atomic-force microscope implemented using an optical lever. *Journal of Applied Physics*, 65(1):164–167.

Atkinson, A. C., Donev, A. N., and Tobias, R. D. (2007). *Optimum Experimental Designs, with SAS*. Oxford University Press.

Azzerboni, B., Carpentieri, M., Finocchio, G., and Ipsale, M. (2004). Super-lorentzian preisach function and its applicability to model scalar hysteresis. *Physica B: Condensed Matter*, 343(1-4):121–126.

Balchen, J., Andresen, T., and Foss, B. (2004). *Reguleringsteknikk*. NTNU trykk.

Balchen, J. G. (1990). Rational transfer function approximations to transport delay. *Modeling, Identification and Control*, 11(3):127–140.

Ballato, A. (1996). Piezoelectricity: History and new thrusts. volume 1, pages 575–583, San Antonio, TX, USA. IEEE, Piscataway, NJ, United States.

Bashash, S. and Jalili, N. (2007). Robust multiple frequency trajectory tracking control of piezoelectrically driven micro/nanopositioning systems. *IEEE Transactions on Control Systems Technology*, 15(5 SPEC. ISS.):867–878.

Bertotti, G. (1998). *Hysteresis in magnetism: for physicists, materials scientists, and engineers*. Academic Press New York.

Binnig, G., Rohrer, H., Gerber, C., and Weibel, E. (1982). Surface studies by scanning tunneling microscopy. *Physical Review Letters*, 49(1):57–61.

Binnig, G. b. c., Quate, C. d., and Gerber, C. c. (1986). Atomic force microscope. *Physical Review Letters*, 56(9):930–933.

Brokate, M. and Sprekels, J. (1996). *Hysteresis and phase transitions*, volume 121. Springer.

Callister, W. D. and Rethwisch, D. G. (2011). *Materials Science and Engineering*. John Wiley & Sons.

Chien, C., Wu, Y., Chiou, Y., Hsieh, C., Chen, Y., Chen, T. d., Tsai, M., and Wang, C. (2006). Nanoscale deformation measurement by using the hybrid method of gray-level and holographic interferometry. *Optics and Lasers in Engineering*, 44(1):80–91.

Conway, J. B. (2012). *A Course in Abstract Analysis*, volume 141. American Mathematical Soc.

Croft, D., Shed, G., and Devasia, S. (2001). Creep, hysteresis, and vibration compensation for piezoactuators: Atomic force microscopy application. *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, 123(1):35–43.

Defoy, B., Genty, L., and Mahfoud, J. (2013). Experimental investigations on the effectiveness of electromagnetic actuator as sensor. *Mechanics and Industry*, 14(4):247–252. cited By (since 1996)0.

Devasia, S., Eleftheriou, E., and Moheimani, S. (2007). A survey of control issues in nanopositioning. *IEEE Transactions on Control Systems Technology*, 15(5 SPEC. ISS.):802–823.

DeVito, C. L. (1978). *Functional Analysis*. Academic Press.

Dixson, R., Koning, R., Fu, J., Vorburger, T., and Renegar, B. (2000). Accurate dimensional metrology with atomic force microscopy. volume 3998, pages 362–368, Santa Clara, CA, USA. Society of Photo-Optical Instrumentation Engineers, Bellingham, WA, United States.

Du, J., Feng, Y., Su, C.-Y., and Hu, Y.-M. (2009). On the robust control of systems preceded by coleman-hodgdon hysteresis. pages 685–689, Christchurch.

Egeland, O. and Gravdahl, J. T. (2002). *Modeling and Simulation for Automatic Control*. Marine Cybernetics Trondheim, Norway.

Eielsen, A. A. (2012). *Topics in Control of Nanopositioning Devices*. PhD thesis, Norwegian University of Science and Technology, Department of Engineering Cybernetics.

Eielsen, A. A., Gravdahl, J. T., Pettersen, K., and Vogl, L. (2010). Tracking control for a piezoelectric nanopositioner using estimated states and feedforward compensation of hysteresis. In *Proceedings of the 5th IFAC Symposium on Mechatronic Systems*.

Eielsen, A. A., Gravdahl, J. T., and Pettersen, K. Y. (2012). Adaptive feed-forward hysteresis compensation for piezoelectric actuators. *Review of Scientific Instruments*, 83(8):085001.

Eielsen, A. A., Polóni, T., Johansen, T. A., and Gravdahl, J. T. (2011). Experimental comparison of online parameter identification schemes for a nanopositioning stage with variable mass. In *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics*.

Esbrook, A., Tan, X., and Khalil, H. (2013). Control of systems with hysteresis via servocompensation and its application to nanopositioning. *IEEE Transactions on Control Systems Technology*, 21(3):725–738.

Fleming, A. (2011). A method for reducing piezoelectric non-linearity in scanning probe microscope images. pages 2861–2866, San Francisco, CA.

Fleming, A., Wills, A., and Moheimani, S. (2008). Sensor fusion for improved control of piezoelectric tube scanners. *IEEE Transactions on Control Systems Technology*, 16(6):1265–1276.

Galinaitis, W. and Rogers, R. (1998). Control of a hysteretic actuator using inverse hysteresis compensation. volume 3323, pages 267–277, San Diego, CA.

Garcia, R. b.a, R. and Perez, R. (2002). Dynamic atomic force microscopy methods. *Surface Science Reports*, 47(6-8):197–301.

Goldfarb, M. and Celanovic, N. (1997). Modeling piezoelectric stack actuators for control of micromanipulation. *IEEE Control Systems Magazine*, 17(3):69–79.

Goodwin, G. C. and Payne, R. L. (1977). *Dynamic System Identification: Experimental Design and Data Analysis*. Academic Press.

Gorbet, R., Morris, K., and Wang, D. (1999). Control of hysteretic systems: a state-space approach. In *Learning, control and hybrid systems*, pages 432–451. Springer.

Henze, O. and Rucker, W. (2002). Identification procedures of preisach model. *IEEE Transactions on Magnetics*, 38(2 I):833–836.

Hughes, D. and Wen, J. (1997). Preisach modeling of piezoceramic and shape memory alloy hysteresis. *Smart Materials and Structures*, 6(3):287–300.

Ioannou, P. A. and Sun, J. (2003). *Robust Adaptive Control*. tapir akademiske forlag.

Ismail, M., Ikhouane, F., and Rodellar, J. (2009). The hysteresis bouc-wen model, a survey. *Archives of Computational Methods in Engineering*, 16(2):161–188.

Iyer, R. and Shirley, M. (2004). Hysteresis parameter identification with limited experimental data. *IEEE Transactions on Magnetics*, 40(5):3227–3239.

Iyer, R. and Tan, X. (2009). Control of hysteretic systems through inverse compensation. *Control Systems, IEEE*, 29(1):83–99.

Iyer, R., Tan, X., and Krishnaprasad, P. (2005). Approximate inversion of the preisach hysteresis operator with application to control of smart actuators. *IEEE Transactions on Automatic Control*, 50(6):798–810.

Jiles, D. and Atherton, D. (1986). Theory of ferromagnetic hysteresis. *Journal of Magnetism and Magnetic Materials*, 61(1-2):48–60.

Juhász, L., Maas, J., and Borovac, B. (2011). Parameter identification and hysteresis compensation of embedded piezoelectric stack actuators. *Mechatronics*, 21(1):329–338.

Korpelainen, V. and Lassila, A. (2007). Calibration of a commercial afm: Traceability for a coordinate system. *Measurement Science and Technology*, 18(2):395–403.

Leang, K.-K. and Fleming, A.-J. (2009). High-speed serial-kinematic spm scanner: design and drive considerations. *Asian Journal of Control*, 11(2):144–153.

Macki, J. W., Nistri, P., and Zecca, P. (1993). Mathematical models for hysteresis. *SIAM Review*, 35(1):94–123.

Mayergoyz, I. (2003). *Mathematical Models of Hysteresis and their Applications*. Academic Press.

Mulder, M. (1997). *Basic Principles of Membrane Technology*. Kluwer Academic Publishers.

Natale, C., Velardi, F., and Visone, C. (2001). Identification and compensation of preisach hysteresis models for magnetostrictive actuators. *Physica B: Condensed Matter*, 306(1-4):161–165.

Newcomb, C. and Flinn, I. (1982). Improving the linearity of piezoelectric ceramic actuators. *Electronics Letters*, 18(11):442–444.

Ragazzon, M. R. P. (2013). Nanopositioning in atomic force microscopes : Robust control design, order reduction and numerical implementability. Master's thesis, Norwegian University of Science and Technology, Department of Engineering Cybernetics.

Ragazzon, M. R. P., Eielsen, A. A., and Gravdahl, J. T. (2014). Hinf reduced order control for nanopositioning: Numerical implementability. *19th World Congress of the International Federation of Automatic Control*.

Riccardi, L., Naso, D., Janocha, H., and Turchiano, B. (2012). A precise positioning actuator based on feedback-controlled magnetic shape memory alloys. *Mechatronics*, 22(5):568–576.

Rosenbaum, S., Ruderman, M., Ströhla, T., and Bertram, T. (2010). Use of jiles-atherton and preisach hysteresis models for inverse feed-forward control. *IEEE Transactions on Magnetics*, 46(12):3984–3989.

Sadeghzadeh, A., Asua, E., Feuchtwanger, J., Etxebarria, V., and García-Arribas, A. (2012). Ferromagnetic shape memory alloy actuator enabled for nanometric position control using hysteresis compensation. *Sensors and Actuators, A: Physical*, 182:122–129.

Sadowski, N., Batistela, N., Bastos, J., and Lajoie-Mazenc, M. (2002). An inverse jiles-atherton model to take into account hysteresis in time-stepping finite-element calculations. *IEEE Transactions on Magnetics*, 38(2 I):797–800.

Spall, J. (2010). Factorial design for efficient experimentation. *IEEE Control Systems Magazine*, 30(5):38–53.

Stakvik, J. (2013). Parameter identification for hysteresis models in nanopositioning. Project Thesis, NTNU.

Tan, X. (2002). *Control of smart actuators*. PhD thesis, University of Maryland.

Tan, X. and Baras, J. (2005). Adaptive identification and control of hysteresis in smart materials. *IEEE Transactions on Automatic Control*, 50(6):827–839.

Tan, X. and Bennani, O. (2008). Fast inverse compensation of preisach-type hysteresis operators using field-programmable gate arrays. pages 2365–2370, Seattle, WA.

Tan, X., Venkataraman, R., and Krishnaprasad, P. (2001). Control of hysteresis: Theory and experimental results. volume 4326, pages 101–112, Newport Beach, CA.

Tan, X. c. and Baras, J. b. (2004). Modeling and control of hysteresis in magnetostrictive actuators. *Automatica*, 40(9):1469–1480.

Tao, G. and Kokotovic, P. V. (1995). Adaptive control of plants with unknown hystereses. *IEEE Transactions on Automatic Control*, 40(2):200–212.

Ueno, T., Qiu, J., and Tani, J. (2003). Magnetic force control with composite of giant magnetostrictive and piezoelectric materials. *IEEE Transactions on Magnetics*, 39(6):3534–3540.

Visintin, A. (1994). *Differential models of hysteresis*. Springer Berlin.

Wei, Z., Sandström, R., and Miyazaki, S. (1998). Shape-memory materials and hybrid composites for smart systems - part i shape-memory materials. *Journal of Materials Science*, 33(15):3743–3762.

Yong, Y., Bhikkaji, B., and Reza Reza Moheimani, S. (2013). Design, modeling, and fpaa-based control of a high-speed atomic force microscope nanopositioner. *IEEE/ASME Transactions on Mechatronics*, 18(3):1060–1071. cited By (since 1996)12.

Zhang, X., Zappe, S., Bernstein, R., Sahin, O., Chen, C.-C., Fish, M., Scott, M., and Solgaard, O. (2004). Micromachined silicon force sensor based on diffractive optical encoders for characterization of microinjection. *Sensors and Actuators, A: Physical*, 114(2-3):197–203.

Zhao, X. and Tan, Y. c. d. e. (2006). Neural network based identification of preisach-type hysteresis in piezoelectric actuator using hysteretic operator. *Sensors and Actuators, A: Physical*, 126(2):306–311.