

Abstract

ZHONG, JINGHUA. Modeling and Control of Piezoceramic Actuators for Nanopositioning. (Under the direction of Stefan Seelecke.)

Actuators from piezoceramic materials have established themselves in a large number of applications, especially nano-scale positioning tasks because of their high-frequency response behavior and their essentially infinite resolution. However, the potential of these actuators has been impaired by the lack of adequate hysteresis models and control methods to compensate the undesirable non-linear, hysteretic and rate-dependent behavior of piezoceramic materials. In this thesis, representative actuation experiments are conducted on a typical commercially available piezoceramic stack actuator to characterize the rate-dependence of its hysteretic behavior over a large stroke under different loading rates. A strongly physics oriented energy model is implemented to verify its ability in modeling dynamic hysteretic behaviors. By extending this model, a computationally efficient implementation is developed with a drastic improvement in speed. Finally, a preliminary version of the energy model for the piezoceramic material is implemented in an optimal control program to illustrate the features and potential of the method. The implementations constitute the basis for future development of advanced realtime control algorithms for piezoceramic actuators.

MODELING AND CONTROL OF PIEZOCERAMIC ACTUATORS
FOR NANOPositionING

BY
JINGHUA ZHONG

A THESIS SUBMITTED TO THE GRADUATE FACULTY OF
NORTH CAROLINA STATE UNIVERSITY
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

DEPARTMENT OF MECHANICAL AND AEROSPACE ENGINEERING

RALEIGH
AUGUST 2003

APPROVED BY:

CHAIR OF ADVISORY COMMITTEE

To my parents, *Zhong Yangsheng* and *Zhong Suzhen*

Biography

Jinghua Zhong was born in a Hakka family on October 7th 1978, in Guangzhou, China. He attended Pui Ching Primary School and Zhi Xin High School before he went to Beijing in 1997 to attend Tsinghua University. In 2001, after receiving his Bachelor of Engineering in Precision Instruments, Measurement and Control Technology, he enrolled in the Mechanical and Aerospace Engineering Department of NC State University as a Master's student.

Acknowledgments

First I would like to extend the most sincere gratitude to my advisor, Dr. Stefan Seelecke, for being always available when I needed guidance and being so patient and helpful when I procrastinate my thesis. It has been a real pleasure working with you.

Another thank you goes to my committee members, Dr. Ralph Smith and Dr. Richard Keltie, for being on my committee and supporting me. Additionally, I would like to thank Dr. Gregory Buckner, for joining my committee before my defense.

For the research, I am indebted to Dr. Marc Kamlah of Forschungszentrum Karlsruhe in Germany, whose work in Piezoceramics was the starting point of my experiments and modeling.

For encouraging me to pursue a higher degree and helping me through my first year, I would like to thank Dr. Paul Ro, my former advisor and the first person to tell me I have a potential in research.

My labmates Olaf Heintze, Jason Frautschi and Aryn Bergman, thank you. Especially Olaf, who helped me with all the experimental setup and inspired me in the countless discussions of our research. You are the man!

Dr. Jeffrey Eischen and Arron Kiefer, thank you for helping me with my notes when I TAed MAE316 last summer. I could not have survived the job without your help.

I would also like to thank my roommates Le Xu(Thomas), Zheng Li(Jordan) and Yu Shu(Peter) for giving me rides all the time and dealing with my terrible sleeping schedule. I'll miss you all.

I have to thank Likyin. This whole thing would not have happened if you hadn't let me leave China. Thank you for giving me freedom.

And last but not least, I want to thank my parents, Zhong Yangsheng and Zhong Suzhen.

Thank you for always being there for me when I needed you. I love you.

Part of the research in this thesis is supported by the NSF grant DMI-0134464.

Table of Contents

List of Tables	viii
List of Figures	ix
1 Introduction	1
1.1 Motivation and Background	1
1.2 Thesis Objective	4
1.3 Thesis Outline	5
2 Experiments	6
2.1 Experimental Setup	6
2.1.1 Test Specimen	6
2.1.2 Strain Measurement	7
2.1.3 Polarization Measurement	8
2.1.4 The Computer Control System	11
2.2 Actuator Behavior under Pure Electrical Loading	12
2.2.1 Test Procedures	12
2.2.2 Full Major Hysteresis Loops	14
2.2.3 Minor Loops in the Actuator Range	19
2.3 Conclusion	29
3 Modeling of Hysteresis	31
3.1 Energy Model for Single Crystal PZTs	31
3.2 Extension of the Model to Polycrystalline Materials	34
3.2.1 Distribution of Energy Barriers and Effective Electric Fields	34
3.2.2 Implementation of Distributions	37
3.2.3 Simulation and discussion	39
3.3 Parametrization Method	43
3.3.1 Parametrization of the Distribution in E_c	44
3.3.2 Parametrization of the Distribution in E_e	47
3.3.3 Combination of Two Distributions	51
3.3.4 An Alternative for the Combined Effective Barrier.	56

3.3.5	Model Fit to Experiments	61
4	Optimal Control of PZT	66
4.1	Optimal Control Problem	66
4.2	Implementation of the Model Used for Optimal Control	68
4.3	Optimal Control Simulation	68
4.3.1	Case 1: Single-level Set-point Function	68
4.3.2	Case 2: A More Complex Set-point Function	71
4.3.3	Conclusion	73
5	Conclusions	75
	List of References	77
A	Fortran Programs	82
A.1	Quadrature Method	82
A.2	Parametrization Method	99
B	Matlab Model and Control Script	121
B.1	Simulink Model for the Sinusoidal Load Experiments	121
B.2	Matlab Control Script for the Model	123

List of Tables

2.1	Technical data for P802.00 PZT stack actuator	7
2.2	Maximum loading rates for different voltage ranges.	14

List of Figures

1.1	Left: SMA Martensite twins[40], Right: Piezoceramic domains[28].	4
2.1	PZT actuator mounted with fiberoptic distance sensor.	8
2.2	Schematic of the Sawyer-Tower circuit for polarization measurement.	9
2.3	Polarization vs. electric field hysteresis loop under a maximum field of $\pm 1.5\text{kV/mm}$	13
2.4	Full hysteresis loop experiment with sinusoidal loading at 0.1Hz.	17
2.5	Full hysteresis loop experiment with piecewise linear loading at 0.1kV/mm/s.	18
2.6	Full hysteresis loops under different frequencies and rates. (a) and (b): sinusoidal load at 0.1Hz(solid) and 1Hz(dashed); (c) and (d): linear load at 0.1kV/mm/s(solid) and 1kV/mm/s(dashed).	19
2.7	Actuator loop experiment with sinusoidal loading at 0.1Hz.	21
2.8	Actuator loop experiment with linear loading at 0.1kV/mm/s.	22
2.9	Minor loops type 1: Polarization. (a) Sinusoidal 0.1Hz; (b) Linear 0.1kV/mm/s; (c) Sinusoidal 100Hz; (d) Linear 100kV/mm/s.	24
2.10	Minor loops type 1: Strain. (a) Sinusoidal 0.1Hz; (b) Linear 0.1kV/mm/s; (c) Sinusoidal 100Hz; (d) Linear 100kV/mm/s.	25
2.11	Minor loops type 2: Polarization. (a) Sinusoidal 0.1Hz; (b) Linear 0.1kV/mm/s; (c) Sinusoidal 100Hz; (d) Linear 100kV/mm/s.	25
2.12	Minor loops type 2: Strain. (a) Sinusoidal 0.1Hz; (b) Linear 0.1kV/mm/s; (c) Sinusoidal 100Hz; (d) Linear 100kV/mm/s.	26
2.13	Minor loops type 3: Polarization. (a) Sinusoidal 0.1Hz; (b) Linear 0.1kV/mm/s; (c) Sinusoidal 100Hz; (d) Linear 100kV/mm/s.	27
2.14	Minor loops type 3: Strain. (a) Sinusoidal 0.1Hz; (b) Linear 0.1kV/mm/s; (c) Sinusoidal 100Hz; (d) Linear 100kV/mm/s.	28
2.15	Minor loops type 4: Polarization. (a) Sinusoidal 0.1Hz; (b) Linear 0.1kV/mm/s; (c) Sinusoidal 100Hz; (d) Linear 100kV/mm/s.	29
2.16	Minor loops type 4: Strain. (a) Sinusoidal 0.1Hz; (b) Linear 0.1kV/mm/s; (c) Sinusoidal 100Hz; (d) Linear 100kV/mm/s.	30
2.17	Relaxation: (a) After sinusoidal loads; (b) After linear loads.	30
3.1	PbTiO ₃ crystal lattice cells with positive and negative polarization	32

3.2	(a) Helmholtz energy ψ and Gibbs energy G for increasing field E ; (b) Polarization P as a function of E for a single crystal with uniform lattice.	33
3.3	(a) Nonuniform PZT lattice structure; (b) Free energies associated with lattice widths (i) and (ii); (c) Variations in hysteresis kernel due to different free energies.	35
3.4	E_c Distribution, $\overline{E}_c = 0.5 \times 10^6$, $b_1 = 1.0 \times 10^{11}$, 12 total abscissae points.	38
3.5	E_e Distribution, $b_2 = 1.0 \times 10^{11}$, 12 total abscissae points.	39
3.6	Simulation of PZT polarization using the quadrature method.	40
3.7	Polarization simulated with 9 different combinations of distribution parameters.	42
3.8	Evolution of E_c during a transformation process.	46
3.9	Schematic of $E_c(\Delta x_+)$. (Dashed line: $E = \overline{E}_c$)	47
3.10	Left: Parametrized version; Right: Quadrature version.	47
3.11	Evolution of E_e during a transformation process.	49
3.12	Schematic of $E_e(\Delta x_+)$.	50
3.13	Left: Parametrized version; Right: Quadrature version.	51
3.14	Domains with different combinations of energy barriers and effective fields.	52
3.15	Hysteresis kernels with different combinations of coercive fields and effective fields.	54
3.16	Illustration of the domain of integral in equation 3.32.	55
3.17	Illustration of the domain of integral in equation 3.33.	56
3.18	Example of a simple minor loop.	57
3.19	Domain states at point (1) - Left: x_+ ; Right: x_- .	58
3.20	Domain states at point (2) - Left: x_+ ; Right: x_- .	58
3.21	Domain states at point (3) - Left: x_+ ; Right: x_- .	58
3.22	Major loop and a family of minor loops simulated with two distributions.	59
3.23	Left: Parametrized version; Right: Quadrature version.	60
3.24	Left: Polarization; Right: Strain.	61
3.25	Type 1: Left: Experiment; Right: Simulation.	62
3.26	Type 2: Left: Experiment; Right: Simulation.	62
3.27	Type 3: Left: Experiment; Right: Simulation.	63
3.28	Type 4: Left: Experiment; Right: Simulation.	63
3.29	Relaxation bahavior (Experimental data) - Left: S-E Hysteresis; Right: Strain relaxation over time.	64
3.30	Relaxation bahavior (Parametrized method) - Left: S-E Hysteresis; Right: Strain relaxation over time.	65
3.31	Relaxation bahavior (Quadrature method) - Left: S-E Hysteresis; Right: Strain relaxation over time.	65
4.1	Left: hysteresis curve at 0.125Hz; Right: hysteresis curve at 125Hz.	69
4.2	Optimal control of piezoceramic polarization (Case 1).	70
4.3	P-E behavior of the optimal solution (Case 1).	71
4.4	Optimal control of piezoceramic polarization (Case 2).	72

4.5	P-E behavior of the optimal solution (Case 2).	73
B.1	Main model for sinusoidal load experiments	121
B.2	Signal generation part of the model for sinusoidal loads.	122
B.3	Subsystem for generating the sinusoidal signal with changing amplitudes and offsets.	122

Chapter 1

Introduction

1.1 Motivation and Background

Recent years have seen unprecedented growth in nanotechnology and the emerging of its supporting industry. Despite the fast progress in the field, any commercially viable product inevitably relies on its manufacturing ability. Among all the enabling technologies in nanomanufacturing, the most basic one is the task of positioning. Regardless of the assembly approach, manufacturing of nanoscale structures will always rely on the ability to precisely control the motion of machine tools or manipulating devices.

Devices such as translation stages are commonly used in the area of nano-scale positioning, e.g., scanning microscopy applications, optical fiber alignment, or general nanomanufacturing tasks. Several industrial companies manufacture and supply these stages and their actuators, e.g. [17][27]. They almost all employ actuators made of so-called active materials, specifically piezoelectric or magnetostrictive materials. This is due to the high-frequency response behavior and the essentially infinite resolution of these materials, which are two most desirable features in the context of efficient nanopositioning. In particular, piezoceramics are much more commonly used because of the special power requirement of magnetostrictives [47].

Even though tremendous progress has been made in the area of hardware development to enable higher positioning and sensing speed using active materials[7], it appears that the control algorithms represent the bottleneck with respect to improved performance. Most controllers are based on conventional PID feedback methods, which have originally been

developed for linear systems. Piezoceramics, however, can be considered to behave linearly only to about 15% of their maximum field strength [5]. In addition, they exhibit undesirable characteristics like hysteresis, creep, temperature dependence and rate dependence. The performance of the controller is severely deteriorated by the overshoot, phase delay and stability problems caused by the undesirable material behavior under large strokes and bandwidths. If one is to make more efficient use of the actuator, he has to account for the highly nonlinear and hysteretic material behavior and incorporate it into the control algorithm.

A first step in the direction of improved control algorithms is a method known as adaptive feedforward control. It is based on the representation of the system and the control as finite impulse response filters (FIR), the coefficients of which are determined by least mean square algorithms. A detailed overview of its application to the control of smart structures is given in [4], see [22] and [46] for a general theory of adaptive filters and design. The method, however, is restricted to linear systems or systems that can be linearized and thus is only applicable in the low drive level range. A next step is the introduction of non-linear models into the feedforward control. In [20], a feedforward model-reference control method has been applied to improve the scanning accuracy in a scanning tunneling microscope. Three different models have been used, however, all of them imply that the hysteresis non-linearity has either local memory or symmetrical behavior, both of which do not exactly represent the hysteretic behavior of piezoceramic actuators. Also, a pure feedforward method in a non-linear environment is always prone to drift away from the desired behavior, so a logical improvement is a hybrid method, which couples a non-linear model with a feedback controller. This method is also known as inverse compensator method, and it has been successfully applied to the control of piezoceramic actuators in [19] and [12], magnetostrictive transducers in [34], and shape memory alloys in [6], [13] and [44]. A general overview can be found in [18] and [43].

All the aforementioned methods have in common that they try to address robustness and stability issues, but none of them takes optimality into account. Optimal control is attractive as it aims to minimize some cost functional like energy consumption or final time. Some attempts toward this end have been made for shape memory alloys and magnetostrictive materials. In [30], a SMA hysteresis model is implemented into an optimal control code

to control the shape of an adaptive beam actuated by a SMA wire. In [41] and [42], the authors compute an optimal control for a magnetostrictive actuator based on an infinite time horizon optimal control problem. Due to the large computational effort, these approaches have up to only recently been confined to offline simulations. In [33], an approach for a magnetostrictive actuator is presented, which combines perturbation feedback control with a nonlinear optimal control method, thus allowing an online implementation. In a series of papers [25], [29], [31] and [32], a similar method has been developed for the optimal control of shape memory alloy actuators. It is based on the concept of a parametric optimal control process combined with a sensitivity analysis, and it reduces the computation time for an optimal control process to the order of nanoseconds. Thus, it has become clearly real-time capable, and, in particular through the combination with a strongly physics-oriented model for the material behavior, the method can be regarded as the basis for the development of extremely efficient control algorithms for active materials actuators.

Regardless of the control theories, the accuracy and efficiency of the model constitute the basis of all model-based control methods. Numerous models have been proposed in the past decades to model hysteresis, most of which are motivated by research in magnetic materials [23]. For piezoceramics, microscopic models at the grain or lattice level are rarely employed for control purposes due to the large number of parameters and states involved. Macroscopic hysteresis models are typically low-order and based on empirical principles, e.g. the classic Preisach model [26] and its variations [10][11], plasticity-based models [21], and other even more phenomenological models based on curve fitting [8][49]. Despite the success of these models in quasi-static or fixed frequency conditions, the absence of physical parameters makes it difficult to incorporate frequency, temperature or load dependencies and adapt to changing operating conditions. Also, most of these models only consider part of the hysteresis loop with a unipolar field [12], which limits the available stroke of the actuators. Semi-macroscopic (mesoscopic) models, including the model that will be used in this thesis, are derived from energy principles and try to combine microscopic mechanisms with macroscopic averages to obtain physical parameters [3][16]. They usually provide a good compromise between capturing the physical properties of the material and being computationally efficient, so that they represent a natural choice for the implementation into a control algorithm.

Due to their close resemblance in microstructure (Figure 1.1) and similarity in macroscopic behavior such as hysteresis, non-linearity and Curie temperature, SMAs, piezoceramics and magnetostrictive materials can be described in a unified framework [36]. In [37], Smith and Seelecke proposed a free-energy-based model for homogeneous single crystal piezoceramics, which is a direct extension of the Müller-Achenbach-Seelecke theory for shape memory alloys [25][29]. The effects of non-uniform lattices, variations in effective fields, and polycrystallinity are introduced by incorporating appropriate distributions in the free energy formulations. The physical parameters in the model greatly alleviate the difficulty to accommodate changing operating conditions for traditional Preisach models, which typically use a large number of unphysical parameters (see [35] for details).

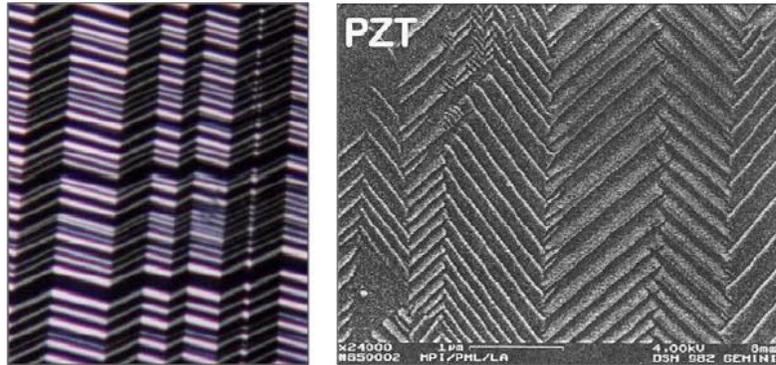


Figure 1.1: Left: SMA Martensite twins[40], Right: Piezoceramic domains[28].

1.2 Thesis Objective

The objective of this thesis consists of three major components: experimental investigation on the hysteretic behavior of a typical piezoceramic actuator, modeling of hysteresis, and feasibility studies of the model in an optimal control problem.

First, we will conduct several types of representative actuation experiments on a typical commercially available piezoceramic stack actuator. We are mostly interested in the rate-dependence of its hysteretic behavior over a long travel range under different loading rates.

The second and also the major part of the work is to improve and implement the energy

model introduced at the end of section 1.1. In terms of improvement, we will focus on the computational efficiency of the model while retaining its energy basis, with the goal in mind that the model will be used for control applications.

Finally, we will study the feasibility of applying the energy model for the piezoceramic material in an optimal control problem. This will serve as the starting point for future development of advanced realtime control algorithms.

1.3 Thesis Outline

The thesis is divided into five chapters as follows:

Chapter 1 motivates the necessity of the research along with a literature review of the previous work on modeling and control of active materials. This chapter also outlines the objectives and structure of the thesis.

Chapter 2 details the experiments conducted on a commercially available piezoceramic actuator. In particular, actuator response is recorded under electrical loads of different orders of reversal. Sinusoidal and piecewise linear loads are applied to the material, with frequencies and rates from as low as quasi-static to as high as the experimental equipments will allow.

Chapter 3 provides a thorough derivation of the model. The chapter starts by reviewing the energy model for single crystal piezoceramics and its extension for polycrystalline material. With the physical principles established, we will first present a straightforward but computationally expensive implementation of the model. We then propose a different implementation with significantly improved efficiency. The improved model is verified against the first implementation and a set of parameters will be identified for the material behavior recorded in Chapter 2.

Chapter 4 presents the first results of an optimal control approach to piezoceramic actuators. An earlier and simplified version of the energy model is implemented into an optimal control package, and two exemplary cases are simulated to illustrate the features and potential of the method.

Finally, Chapter 5 will provide concluding remarks on the results and comment on future work of the project.

Chapter 2

Experiments

This chapter introduces the experimental setup that was used to characterize the hysteretic behavior of a piezoceramic actuator and summarizes the experimental results. The actuator is investigated under pure electrical loading only. Sinusoidal and piecewise linear loads are applied to the material, with various magnitudes of frequencies and rates.

2.1 Experimental Setup

2.1.1 Test Specimen

The actuator chosen for the experiments is a commercially available PZT stack actuator from Physik Instrumente (www.pi.ws), Model P-802.00. This model has the best open-loop resolution in the product line and lowest current requirement, allowing easier setup and measurement. The actuator is made of the PIC151 soft PZT ceramic material proprietary to the PI Ceramics Company. The time effects of bulk PIC151 materials under electrical loading have been detailed in [45] and [47]. Previous research on this material provides us with basic information on the properties of the material and helps us verify our own experimental results, but the experiments in [47] are focused on the behavior of the material under constant electric fields and full cyclic electric fields under quasistatic conditions, while we are more interested in the material behavior in its actuator range under dynamic, higher frequency loads. Important technical data of the actuator is listed in Table 2.1.

The actuator comes with epoxy insulation and heat shrink tube for protection and strain relief of the wire leads to facilitate mounting and driving. The standard operating voltage

Table 2.1: Technical data for P802.00 PZT stack actuator

Open-loop travel @ 0 to 100V	6 $\mu\text{m} \pm 20\%$
Open-loop resolution	0.06 nm
Push/pull force capacity	1000/5 N
Electrical capacitance	0.7 $\mu\text{F} \pm 20\%$
Dynamic operating current coefficient(DOCC)	15 $\mu\text{A}/(\text{Hz} \times \mu\text{m})$
Unloaded resonant frequency(f_0)	55 kHz $\pm 20\%$
Standard operating voltage range	0 to 100V
Standard operating temperature range	-20 to 80 $^{\circ}\text{C}$
Dimensions(with heatshrink packaging)	7 \times 7 \times 9 mm
Dimensions(without packaging)	5 \times 5 \times 9 mm
Layer thickness	0.1 mm
Number of active layers	80
Equivalent surface area	2000 mm ²

range of the actuator is defined in a range where non-linearity is less obvious, and the actuator has been prepoled to full positive polarization in the manufacturing process. With the goal to understand and model the hysteresis, we drive the actuator with a much wider voltage range. In practice, we found the actuator to work properly with at least $\pm 150\text{V}$, which converts to an electric field of $\pm 1.5\text{kV}/\text{mm}$. The material itself can survive an electric field of up to $5\text{kV}/\text{mm}$ [47] but the stack layer packaging of the actuator is not rated for such high voltage ranges. We have found one of the specimens to break down after being driven at 200V ($2\text{kV}/\text{mm}$) for an extended period of time. The displacement response of the broken actuator is almost unaffected, but the leakage current is significantly higher than normal. Therefore we limit our experiments to a range of $\pm 150\text{V}$ ($\pm 1.5\text{kV}/\text{mm}$). The largest negative field without repoling the actuator to negative polarization is $-0.4\text{ kV}/\text{mm}$, so we consider -0.4 to $1.5\text{kV}/\text{mm}$ to be the largest range for use as an actuator. The amplifier for driving the actuator (Model PZD350 Dual Channel Piezo Driver, Trek, Inc., Medina, NY) is capable of outputting $\pm 350\text{V}$ at a slew rate greater than $500\text{V}/\mu\text{s}$ with a current limit of 200mA .

2.1.2 Strain Measurement

To measure the strain of the actuator, a fiberoptic displacement sensor (Philtec RC12, Philtec, Inc. Annapolis, Maryland U.S.A.) is used to measure the displacement at the

front surface of the actuator. The sensor has a standard resolution of 0.075 micron from DC-20kHz. Accounting for the additional noise in our connection, we measured the the sensor signal to have an effective resolution of 0.1 micron. The strain of the actuator is calculated by dividing the measured displacement by the total thickness of active layers. At 150V(1.5kV/mm), the actuator has an extension of 13.3 microns, which converts to a strain of $13.3/(100 \times 80) \times 100\% = 0.16625\%$. Figure 2.1 shows a picture of the mounted actuator together with the tip of the foberoptic sensor. The actuator is glued to the mounting cap with epoxy, and its front surface is covered with a reflective tape to satisfy the optical requirement of the fiberoptic sensor.

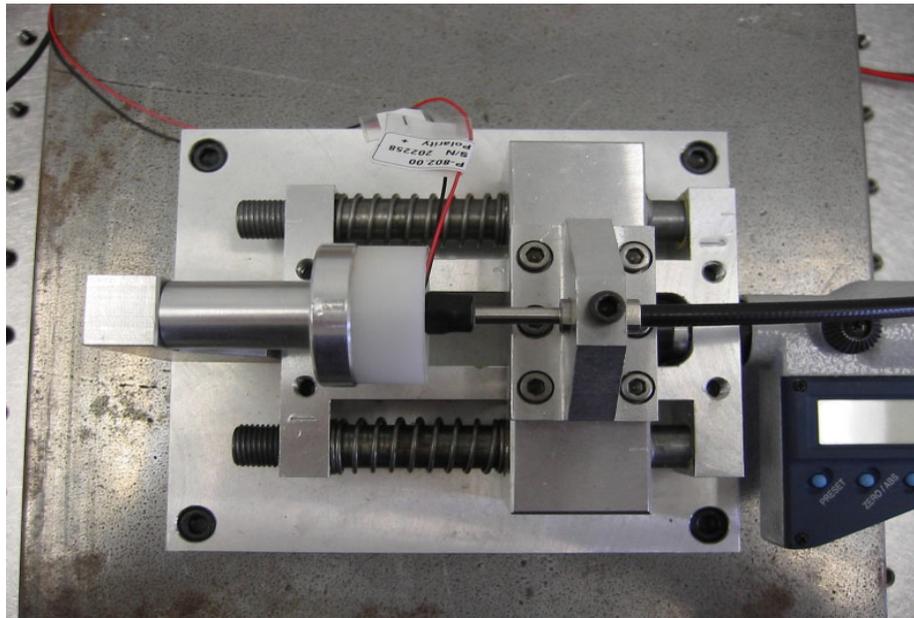


Figure 2.1: PZT actuator mounted with fiberoptic distance sensor.

2.1.3 Polarization Measurement

The polarization in piezoceramic materials under electrical loading is most commonly measured using the Sawyer-Tower method [9][47], which is also what we use in our experiments. The schematic drawing of the test circuit is shown in Figure 2.2.

The key to the Sawyer-Tower method is the use of a reference capacitor connected in

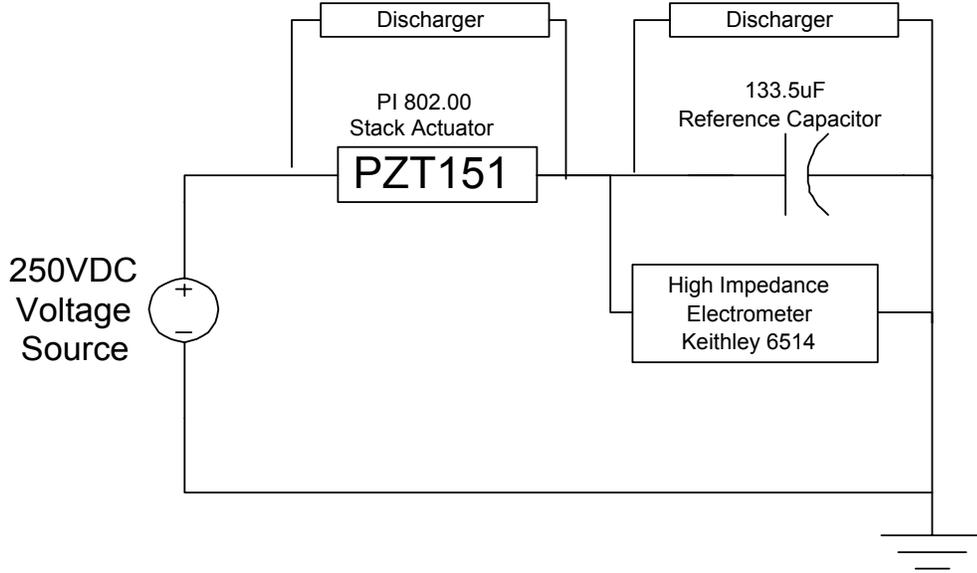


Figure 2.2: Schematic of the Sawyer-Tower circuit for polarization measurement.

series with the PZT actuator, as shown in Figure 2.2. Polarization is defined as the dipole moment per unit volume or alternatively, the amount of charge per unit area. The charge on the PZT can not be measured directly, but the reference capacitor always takes the same amount of charge as the PZT because they are connected in series. The charge on the reference capacitor is

$$Q_r = U_r C_r, \quad (2.1)$$

where U_r is the voltage across the reference capacitor and C_r is the capacitance of the reference capacitor. The polarization of the PZT actuator can then be calculated as

$$P_{pzt} = \frac{Q_r}{A_{pzt}} = \frac{U_r C_r}{A_{pzt}},$$

where A_{pzt} is the equivalent area of the stack actuator. For the P802.00 actuator, $A_{pzt} = 2000\text{mm}^2$. In the experiments, the voltage on the reference capacitor is first fed to a Keithley 6514 electrometer, which acts as a high impedance buffer to improve measurement quality. The voltage is then monitored and collected by the AD acquisition card in the computer control system, which is detailed in the next section.

The choice of capacitance for the reference capacitor plays an important role in the measurement and control of the experiments. In principle, the capacitance of the reference capacitor should be as large as possible, so that its voltage U_r is negligible compared to the voltage on the PZT and the voltage output from the amplifier U_{amp} acts almost entirely on the PZT. This also makes it easier for the AD system to measure the voltage on the reference capacitor, eliminating additional signal conditioning for high voltages outside the range of the AD board. To call the reference voltage U_r negligible, the reference capacitance C_r must be at least 1000 times larger than the PZT actuator capacitance C_{pzt} . However, the capacitance of the P802 PZT actuator is at least 0.8 uF ¹, making it almost impossible to find a reference capacitor with a capacitance three orders of magnitudes large.

In our setup, we use 4 high-quality low-leakage polypropylene capacitors (Model X300, ASC Capacitors, Ogallala, NE) in parallel to get a reference capacitor of $C_r = 133.5 \text{ uF}$. C_r is large enough to bring the reference voltage U_r into a desirable range of the AD acquisition system, but U_r can still be as high as 6V when the amplifier output U_{amp} is equal to 150V. In this case, the voltage on the PZT is only $U_{pzt} = U_{amp} - U_r = 150 - 6 = 144 \text{ V}$. This is an error of 4% if we do not account for the voltage drop on the reference capacitor and consider U_{pzt} to be equal to the amplifier output U_{amp} . Therefore, we do not neglect the reference voltage when applying the voltage to the PZT. The voltage on the reference capacitor is measured and added to the prescribed PZT voltage in the control computer in real-time and the sum is used as the control signal to the amplifier. When the sampling frequency of the control system is high enough, the error of the PZT voltage due to the voltage drop on the reference capacitor will be compensated. This way we guarantee that the voltage on the PZT is indeed the voltage that we prescribe. Our computer control system samples at 20kHz and the voltage error on the PZT is maintained below 0.5%.

Another requirement in the measurement of polarization is the task of properly discharging the PZT and the reference capacitors after each experiment, otherwise the charge on the reference capacitor will keep accumulating until it overflows the AD/DA system. For this reason, two discharger circuits using magnetic relays are added to the system, which are also controlled by the computer. When the experiments are finished and the discharger circuits

¹This is the capacitance in the normal operating range. In the repoling process, the dynamic equivalent capacitance of the PZT becomes much larger.

are activated, the PZT actuator and the reference capacitor will be separately discharged with a $10\text{k}\Omega$ resistor.

2.1.4 The Computer Control System

The computer control system for the experiments consists of two computers: One host computer for prescribing the test process, and one target computer dedicated to the execution of the experiments.

The experiments are conducted using MATLAB 6.5/Simulink 5.0 with xPC Target 2.0. Simulink models for the experiments are first compiled into xPC Target programs in MATLAB/Simulink on the host PC running the Windows2000 operating system. Subsequently, the target programs are transferred to the target PC, the only task of which is to run the target program in a special real-time operating system environment. The performance of the host PC has no influence on the performance of the program running on the target PC. The target PC features an Athlon XP 1.6GHz CPU with 1GB of memory and runs the xPC Target 2.0 Real-time Operating System. For the Simulink applications used in our experiments, the target computer is capable of executing the model equations and storing outputs in RAM within a time step of 0.00005 second, which is equal to a sampling frequency of 20kHz.

After compiling and downloading the model application to the target computer, additional control on the prescribed test voltages is available from the host computer using a Matlab script file that updates the model parameters on the target computer. The collected experimental data is sent back to the host computer after each experiment for post processing and storage. The simulink models and associated Matlab control scripts for the experiments are documented and included in the appendices of the thesis.

The control voltage to the amplifier and discharger, the voltage signal from the reference capacitor, the voltage and current monitor signal from the amplifier, and the voltage signal from the fiberoptic displacement sensor are all sent or received through one multi-channel AD/DA acquisition board (NI 6052E, National Instruments Corporation, Austin, TX) in the target computer.

2.2 Actuator Behavior under Pure Electrical Loading

This section summarizes the test results of the P802.00 PZT stack actuator under pure electrical loading. Within the capability of the amplifier, both full major hysteresis loops and actuator loops (partial loops in the actuation range) are measured from quasistatic to the highest loading rates allowed. Sinusoidal and piecewise linear loads are applied to the actuator to better characterize the polarization vs. field and strain vs. field behavior of the actuator. Significant rate-dependence of the hysteresis loops are observed from the data. Furthermore, creep-like relaxation behavior under constant electric field is also recorded and analyzed.

2.2.1 Test Procedures

The strong hysteretic properties of the material behavior lead to different polarization states at zero electric field depending on the prior history. Ideally, the actuator should start from a state of either full positive or full negative polarization so that the history of the loads can be reliably tracked with respect to the full outer hysteresis loop. In all the experiments, we choose to start from a full positive polarization because the P802 stack actuator has been fully pre-poled to a positive polarization in its manufacturing process, and depoling it to a full negative polarization would require a voltage that could be harmful to its layer packaging. The PIC151 soft PZT can be considered fully poled at an electric field of 2kV/mm [47], but as described in section 2.1.1 we have found the actuator to start breaking down after working at this range for an extended period of time. On the other hand, at 1.5 kV/mm, the actuator already begins to show saturation in its phase transformation process, as can be observed in Figure 2.3, which shows a polarization vs. electric field hysteresis loop between ± 1.5 kV/mm. Based on these considerations, we always load the actuator to 1.5kV/mm and let the actuator saturate at this field before any of the experiments. In addition to loading the actuator to different return points, we hold the electric field constant after reaching the last point to observe the creep-like relaxation behavior of the actuator at different fields in different loading directions. At the end of the experiments, the electric field is slowly unloaded back to zero. The standard loading procedures for all the experiments are:

1. Load the actuator from 0 to 1.5kV/mm with a 0.1Hz sinusoidal load (5 seconds);
2. Hold the electric field constant at 1.5kV/mm for 10 seconds to allow for full polarization;
3. Start the experiment from 1.5kV/mm;
4. Hold the electric field constant at what the experiment ends with for 10 seconds to record the relaxation;
5. Unload the actuator back to zero field with a 0.1 Hz sinusoidal load (5 seconds).

In the above procedures, steps 1 and 2 initialize the PZT to its maximum positive polarization and step 5 finalizes the experiment. Only the data in steps 3 and 4 are relevant to each experiment and recorded in the post-processing stage on the host computer.

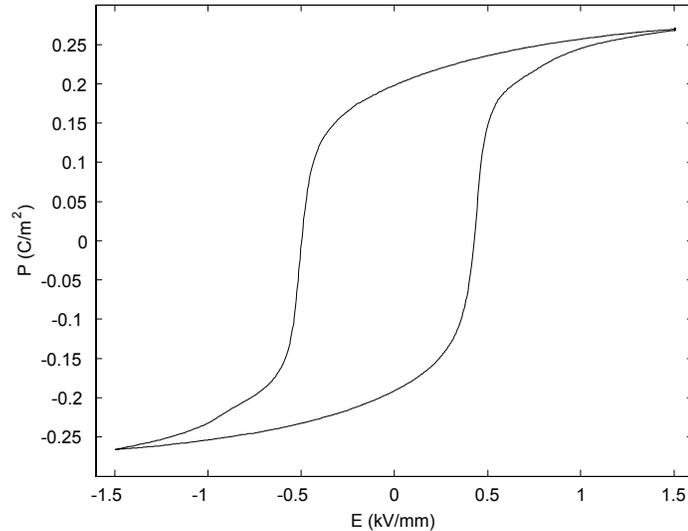


Figure 2.3: Polarization vs. electric field hysteresis loop under a maximum field of ± 1.5 kV/mm.

The frequency range of the experiments is mainly limited by the maximum current available from the amplifier. The relationship between maximum amplifier output current, voltage and operating frequency can be approximated as (from PI Website)

$$i_{\max} \approx f_{\max} \cdot \pi \cdot C_{pzt} \cdot U_{p-p},$$

where i_{\max} is the peak amplifier current[A], f_{\max} is the maximum sinusoidal operating frequency[Hz], C_{pzt} is the PZT actuator capacitance[F], and U_{p-p} is the peak to peak drive voltage. Therefore the maximum frequency is

$$f_{\max} \approx \frac{i_{\max}}{\pi \cdot C_{pzt} \cdot U_{p-p}}. \quad (2.2)$$

The amplifier that we have is limited to a maximum current of 200 mA. In the actuator range from -40V to 150V, U_{p-p} is equal to 190V and C_{pzt} is approximately $0.8\mu\text{F}$, so according to Equation 2.2 $f_{\max} \approx 0.2/(\pi \times 0.0000008 \times 190) \approx 418$ Hz. For a triangular wave input, $f_{\max}^{lin} \approx f_{\max} \times \pi/2 \approx 656$ Hz, which converts to a loading rate of 250 kV/s. In the full range from -150V to 150V, $f_{\max} = 265$ Hz and $f_{\max}^{lin} = 416$ Hz, which converts to about 170 kV/s. These are ideal approximations of the maximum rates. In reality, several other conditions limit the maximum operating frequency.

First, the PZT capacitance can not be considered linear because more current is needed when phase transformation occurs. $0.8\mu\text{F}$ is the lowest static measurement of C_{pzt} . During the experiments, C_{pzt} will be dynamically changing and the maximum frequency will be lower than the theoretical approximation, especially in a full hysteresis loop where a large number of domains are transforming. Second, when the applied voltage switches direction at a reversal point, the current will be changing its sign. The amplifier output will start fluctuating when this sudden change in current combined with the voltage exceeds its power limit. This fluctuation will cause a large error in the output voltage at the reversal points and so should be avoided. Based on these considerations, the maximum reliable rates for the amplifier that we found are shown in table 2.2

Table 2.2: Maximum loading rates for different voltage ranges.

	Sinusoidal(Hz)	Linear(V/s)
Actuator range (-40V to 150V)	100	10000
Full range (-150V to 150V)	1	100

2.2.2 Full Major Hysteresis Loops

The first set of experiments are full major hysteresis loops between the full range of $\pm 1.5\text{kV/mm}$. These are the most commonly investigated loops in hysteresis research. Figure 2.4 shows an example of all the data recorded in such an experiment, in which the

actuator is driven by one cycle of a sinusoidal load at the quasistatic frequency of 0.1Hz^2 . The prescribed electric field, polarization, strain and current are all recorded along with the time, and the relationships between these data are shown in the 8 subfigures. Subfigures (a), (c), (e) and (f) are the evolution of the electric field, polarization, strain and current over time. Subfigure (b) shows the hysteretic relationship between the polarization and strain. Subfigures (d) and (f) show the most important polarization vs. electric field ($P - E$) and strain vs. electric field ($S - E$) hystereses. The $S - E$ curve is often referred to as the butterfly curve because of its shape. Finally, subfigure (h) shows the creep-like relaxation in strain over the last 10 seconds of the experiment, during which the electric field is held constant at the end point of the load. In this example, the end point is also the starting point at $E = 1.5\text{kV/mm}$, therefore the phase transformation is almost saturated and the polarization and strain increased only very little during the 10-second holding period.

Several important physical properties of the PZT material are revealed in a full outer hysteresis loop, such as the remnant polarization and the coercive field, which are crucial in the modeling of hysteresis. The remnant polarization is the polarization after the material is unloaded to zero field. The coercive field is often defined as the field at which the material obtains zero polarization. They can both be read off subfigure (d) in Figure 2.4, which shows a polarization vs. electric field hysteresis loop.

In subfigure (d) and (f), we can see that the absolute value of polarization and the strain at -1.5kV/mm are both slightly lower than those at $+1.5\text{kV/mm}$. This is due to the fact that the material had been fully pre-poled to a positive polarization and we start the experiment from $+1.5\text{kV/mm}$. Since we started from a positive polarization, at -1.5kV/mm we have not repoled all the domains with a positive polarization, so the loop we observe is actually an inner loop that is slightly biased to the positive side of the polarization. To obtain a completely symmetric outer hysteresis loop, the material must either start from a "virgin" state (the initial state after being cooled down from above its Curie temperature at zero external electric field, where the initial overall polarization becomes zero so that a symmetric alternating load starting from this point will result in a symmetric loop), or

²This is the lowest frequency of all the experiments. Because of the requirement to compensate for the voltage error caused by the reference capacitor, the target computer must sample at 20kHz even though the experiments are at a much lower rate. xPC target does not provide options to decimate collected data online. Data files generated from experiments lower than 0.1Hz are too large for the host computer to download and process. This problem is expected to be solved by the next version of the xPC target system.

be fully repoled to a negative polarization during the experiment. Neither of these two conditions could be satisfied due to the pre-packaging of the PZT stack.

In subfigure (b), a significant hysteresis is observed in the strain vs. polarization curve, which is counter-intuitive to the usual assumption that the strain is quadratically related to the polarization and the S-P curve is a perfect parabola. This phenomenon is also observed and discussed in the work of [45], [48] and [47]. In particular, [48] proposed a tentative explanation that the hysteresis is caused by the domains undergoing two successive slower 90 degree switching processes instead of a faster 180 degree switching process. The hysteresis is expected to vanish at higher frequencies, at which the 90 degree switching does not have enough time to take place and the 180 degree switching process dominates. Although our test results at a low frequency confirmed their findings, we did not observe the parabolic curve under the highest reliable frequency range allowed by our equipment. For a full outer loop, the maximum current exceeds the limit of the amplifier at a frequency of 4Hz, therefore we load at a maximum frequency of 1Hz to guarantee the reliability of the data. At 0.1Hz and 1Hz, the S-P hysteresis curves that we observed are almost exactly the same. Further understanding of this hysteresis will require experiments at higher frequencies and more thorough studies of the micromechanisms of the material, which are not at the scope of this thesis.

The current signal shown in subfigure (g) takes its peak value exactly when the polarization takes the sharpest change and draws the most power. The peak current in this 0.1Hz example is about 0.6 mA. As we have discussed in the previous section, the maximum current is proportional to the loading frequency, so the maximum current will approach the 200mA limit at as low as 4Hz, which is why we have limited the reliable maximum frequency to 1Hz for this particular type of experiment.

A second example of outer hysteresis loop is shown in Figure 2.5. The only difference in this experiment is that the electric field loads are piecewise linear at a rate of 0.1kV/mm/s, as shown in subfigure (a). For the linear load experiments, the loads are also applied at two rates, the lowest being 0.1 kV/mm/s and the highest being 1 kV/mm/s. The characteristics of the curves are very similar to those observed in the sinusoidal experiment.

In all the characteristics and properties of the actuator, we are most interested in the rate dependence of the P-E and S-E hystereses for modeling purposes. Full hysteresis loops

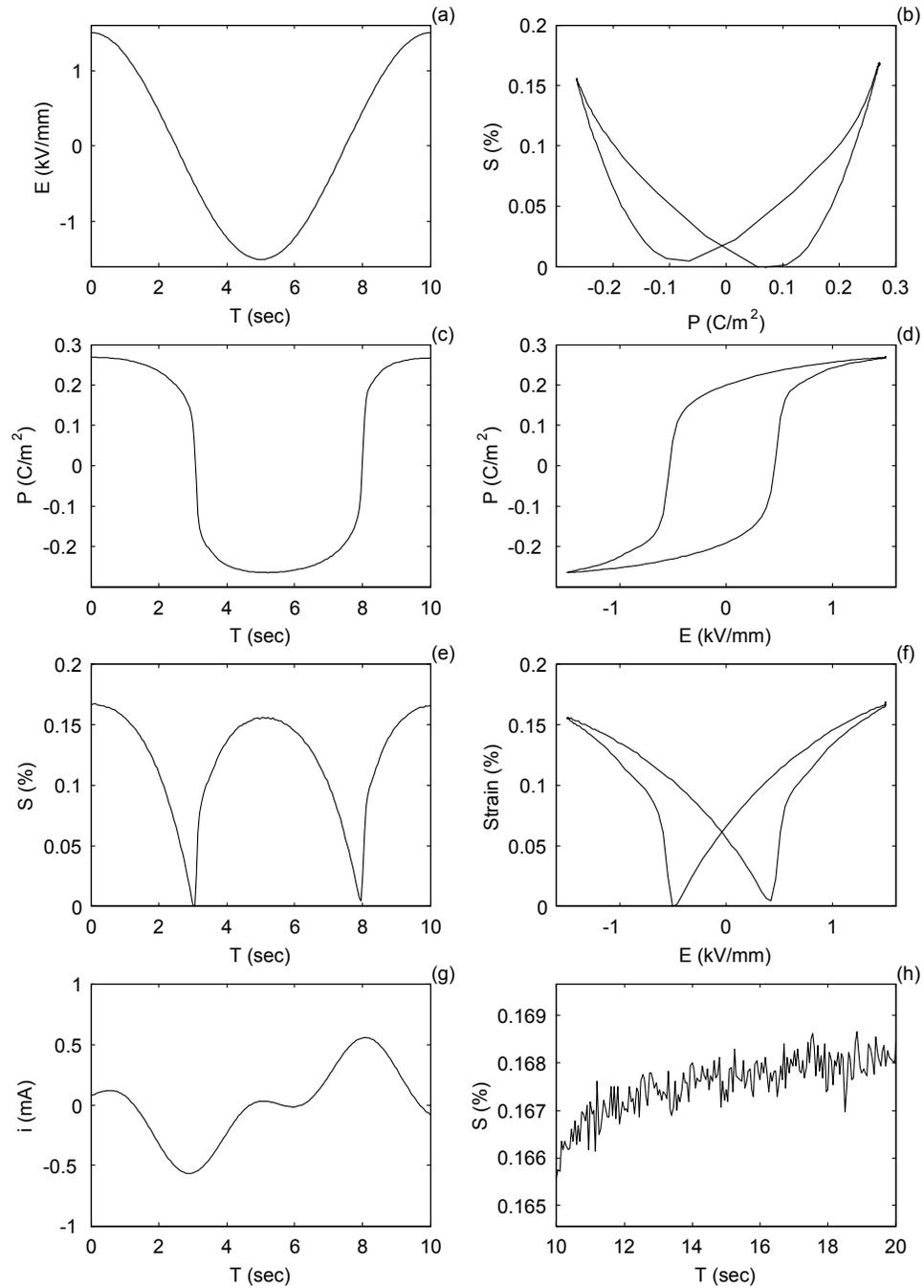


Figure 2.4: Full hysteresis loop experiment with sinusoidal loading at 0.1Hz.

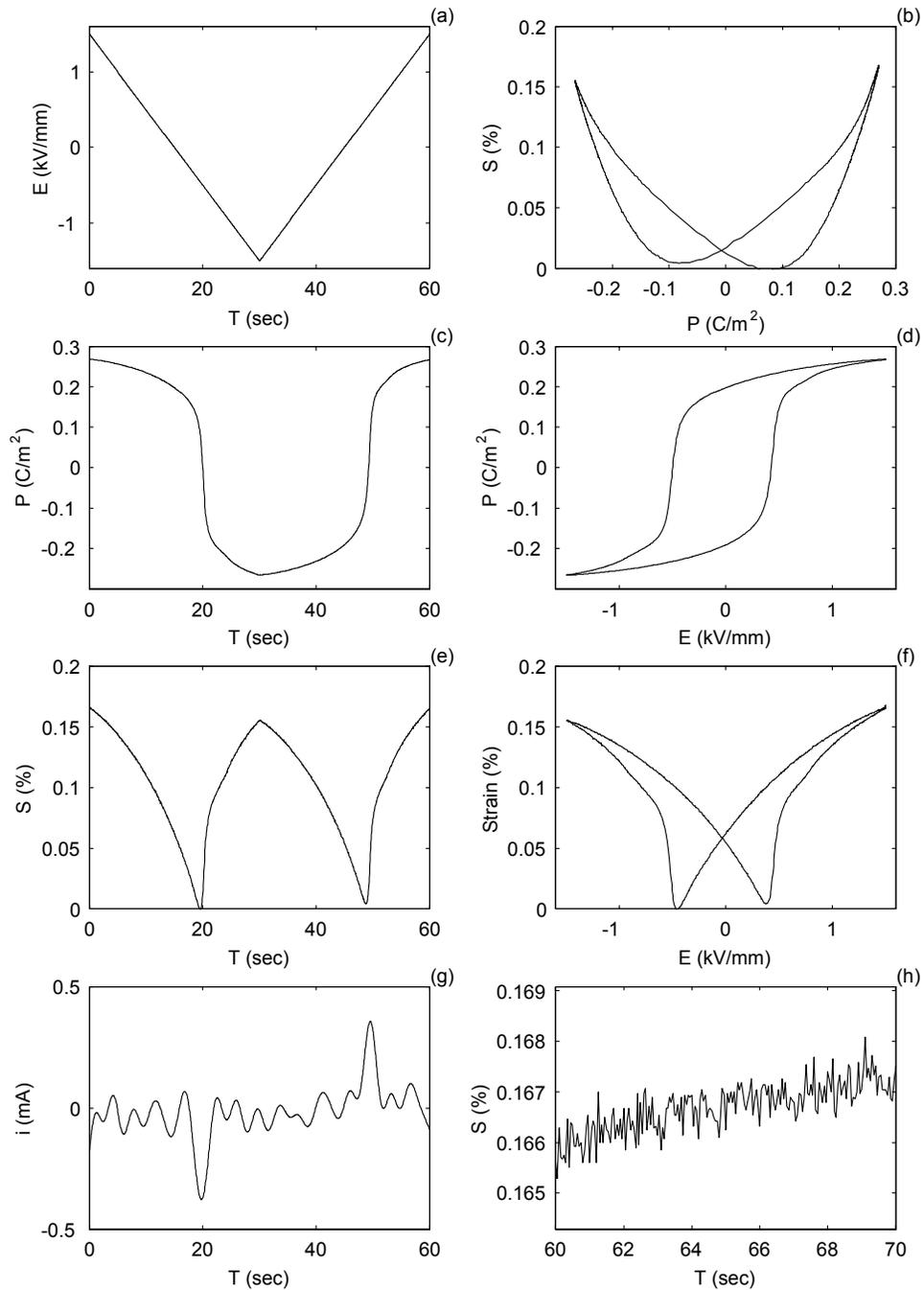


Figure 2.5: Full hysteresis loop experiment with piecewise linear loading at 0.1 kV/mm/s .

at different rates are plotted together in Figure 2.6. The curves in the top row are under sinusoidal loads at 0.1Hz(solid) and 1Hz(dashed), and the curves in the bottom row are under linear loads at 0.1kV/mm/s(solid) and 1kV/mm/s(dashed). Regardless of the type of loading, the hysteresis loops are all widened at higher loading rates, indicating a higher coercive field. The faster loops have a smaller polarization and strain at -1.5kV/mm . This shows that less domains are able to transform within the shorter loading cycle, a clear evidence of rate dependence.

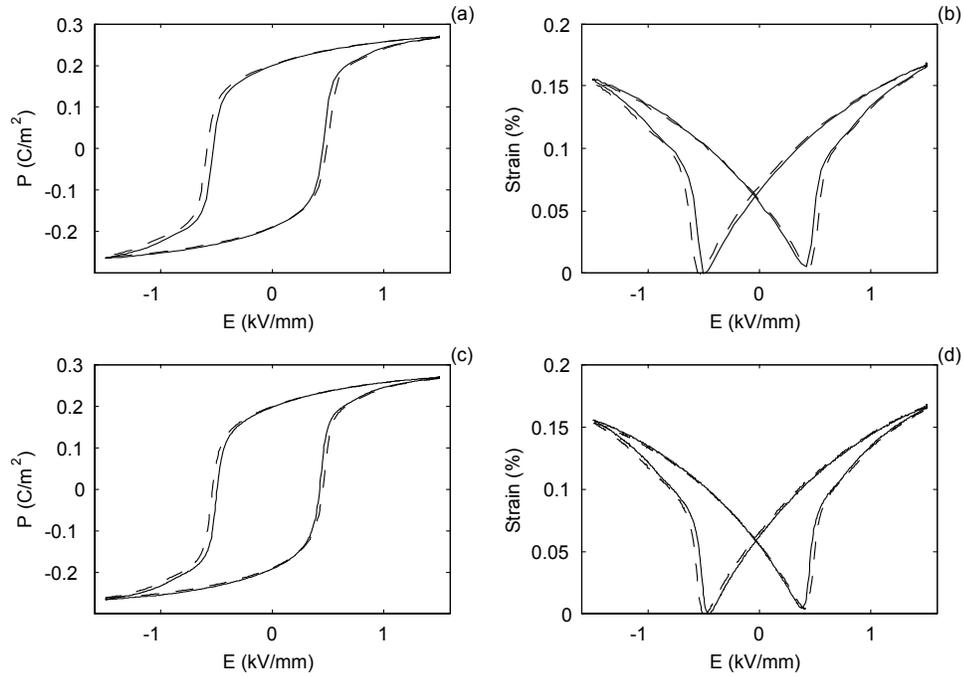


Figure 2.6: Full hysteresis loops under different frequencies and rates. (a) and (b): sinusoidal load at 0.1Hz(solid) and 1Hz(dashed); (c) and (d): linear load at 0.1kV/mm/s(solid) and 1kV/mm/s(dashed).

2.2.3 Minor Loops in the Actuator Range

Inspection of the butterfly loop, e.g. Figure 2.5(f), reveals that for actuation, it is not useful to go through the whole range of electric fields. We can see that the maximum range of actuation is achieved between 1.5kV/mm and -0.5kV/mm . Lowering the field below the

latter value results in an increase in strain rather than a further decrease. Typical actuator applications drive the piezoceramic between 0kV/mm and 1kV/mm, where the behavior is only mildly nonlinear and the hysteresis is comparatively small. In our experiment, we cover mainly the full range from -0.4kV/mm to 1.5kV/mm , which increases the actuator stroke by 100%. However, in this range, the hysteresis can not be considered small any more, and its details are at the focus of the studies in this section. We drive the actuator to varying reversal points within the actuator range at different frequencies to characterize the hystereses. For sinusoidal loads, the frequencies are 0.1Hz, 1Hz, 10 Hz and 100Hz. For piecewise linear loads, the rates are 0.1kV/mm/s , 1kV/mm/s , 10kV/mm/s and 100kV/mm/s . Figure 2.7 and Figure 2.8 are two lowest-rate examples for the experiments in the actuator range, shown in the same way we presented the two full hysteresis examples in the previous section, one with a sinusoidal load and the other with a linear load.

In Figure 2.7, subfigure (a) shows the type of voltage signal applied. The electric field on the actuator starts from 1.5kV/mm , comes down to -0.4kV/mm , and then returns to 1.5kV/mm . This is the largest loop in the actuator range. From subfigure(g) we can see that the peak current also occurs when the polarization takes the sharpest change. In subfigure (d), we no longer see a symmetric $P - E$ loop, but a loop that looks like the profile of a wing. The descending branch of the loop is actually part of the full outer hysteresis loop, with more and more domains transforming when the electric field approaches -0.4kV/mm . The ascending branch, however, is only a small minor loop with respect to the full outer hysteresis loop. The phase transformation is most prominent at about 0.4kV/mm . The hysteresis in the strain vs. polarization curve is not as obvious as that of the full outer loop but still quite visible, as shown in subfigure (b). The linear example in Figure 2.8 is very similar to the sinusoidal one, except that the current signal is so low at this rate it is almost completely buried in the background noise.

In the following sections, we will show different types of minor hysteresis loops within this actuator range. For each experiment, a complete set of data as illustrated in the two examples have been recorded, but in the sequel we will concentrate on the $P - E$ loops and $S - E$ loops and only show their shape characteristics under low and high frequencies.

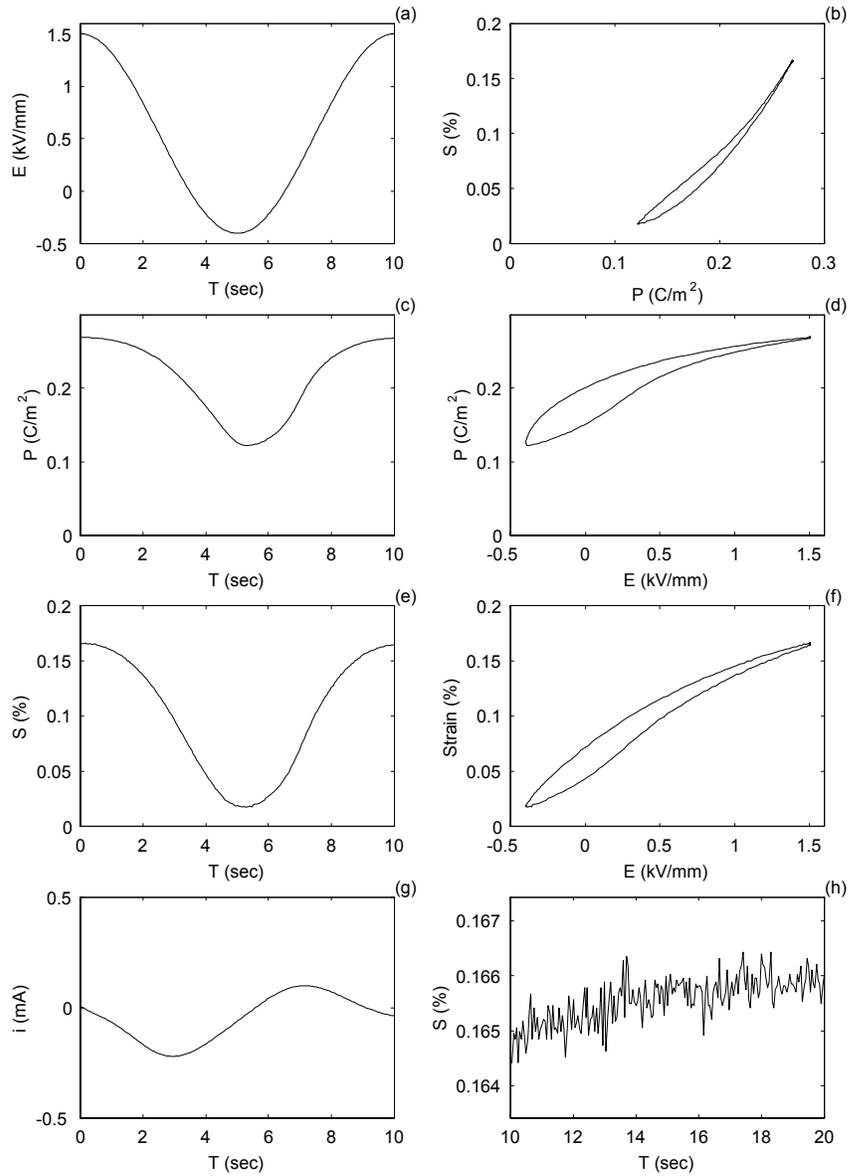


Figure 2.7: Actuator loop experiment with sinusoidal loading at 0.1Hz.

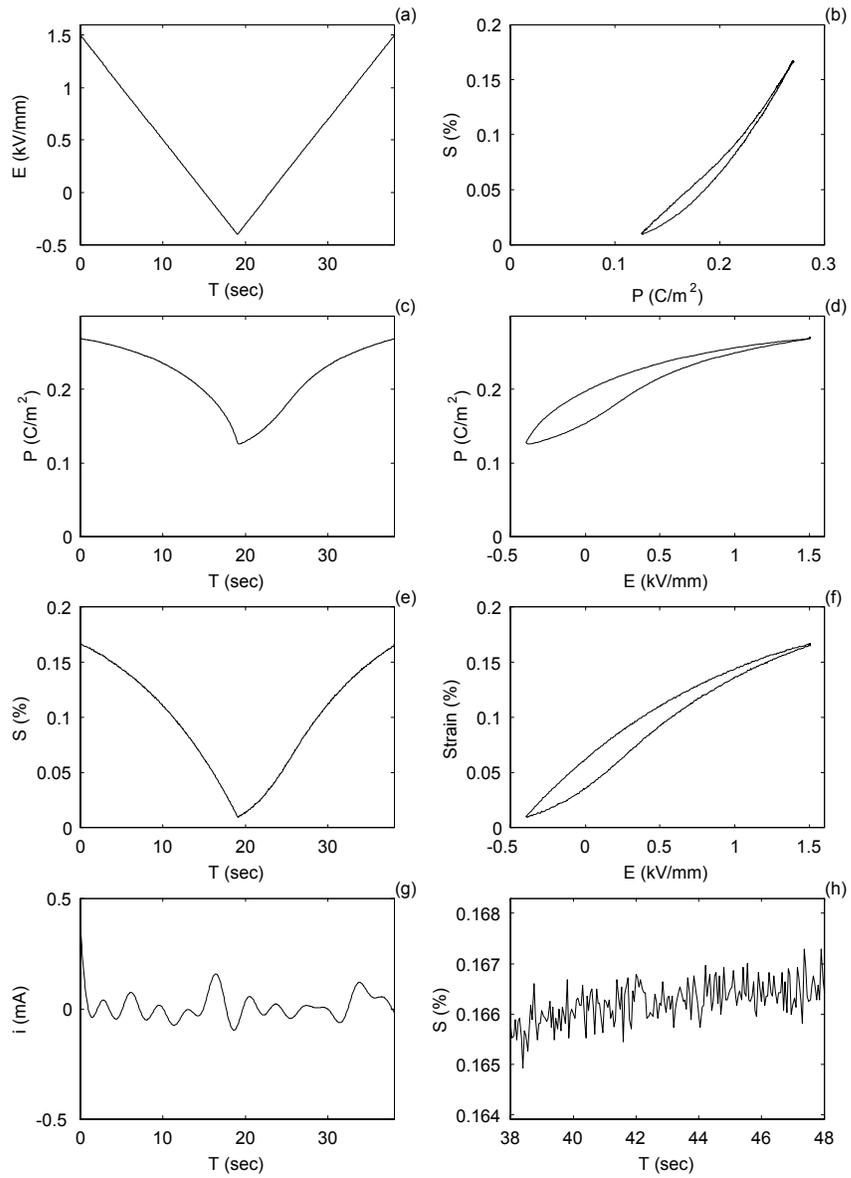


Figure 2.8: Actuator loop experiment with linear loading at 0.1kV/mm/s.

Minor Loops Type 1

The first type of experiments are shown in Figure 2.9 and Figure 2.10. A series of curves are plotted in the same figure. They all reverse from a point on the descending outer loop and return to the starting point at 1.5kV/mm. The return points are chosen to better demonstrate the shape of the minor loops within the range. The electric field is then held at 1.5kV/mm for 10 seconds. The sinusoidal load experiments are conducted at 4 magnitudes of frequencies from 0.1Hz to 100Hz, and the linear load experiments are conducted at 4 magnitudes of rates from 0.1kV/mm/s to 100kV/mm/s, but only experiments at the lowest and the highest rates are plotted in the figures, because experiments at neighboring frequencies are too similar to be discernible on the same plot. Subfigures (a) and (c) show sinusoidal load data at 0.1Hz and 100Hz, while subfigures (b) and (d) show linear load data at 0.1kV/mm and 100kV/mm. The same plotting arrangement will also be used for all other types of experiments hereinafter.

Rate dependency can then be observed from plots on the same column. Take the largest loop for example, at 0.1Hz, polarization has come down to about 0.12 C/m² and strain has reached about 0.02% at the reversal point. At 100Hz, the polarization has only reached 0.16 C/m², a 0.04 C/m² difference from the low frequency result at the same reversal point. The strain for the 100Hz loop also stops at a higher value of 0.035% for the same reversal point as the lower frequency loop. They all show that fewer domains in the PZT are able to transform at higher frequencies.

The linear load experiments show almost the same phenomenon and very similar hysteresis loops. They mostly differ from the sinusoidal load curves at the reversal points. The sinusoidal experiments have a more gradual change in electric field at the reversal points so the loops look more rounded at the reversal points. The linear loops have a sudden rate change at the reversal points and therefore have sharper reversal shapes. This is best seen in the magnified view of the reversal point.

For the same frequency or rate, the ascending loops from different reversal points are similar in that their fastest transformation almost all occurs at 0.4kV/mm. But the higher the reversal point is, the less non-linearity is seen in the curve. In the ten second holding period, very little relaxation is observed at the end point for this type of experiments due to the fact that the phase transformation is nearly complete at 1.5kV/mm so the polarization

can increase only by a very small amount.

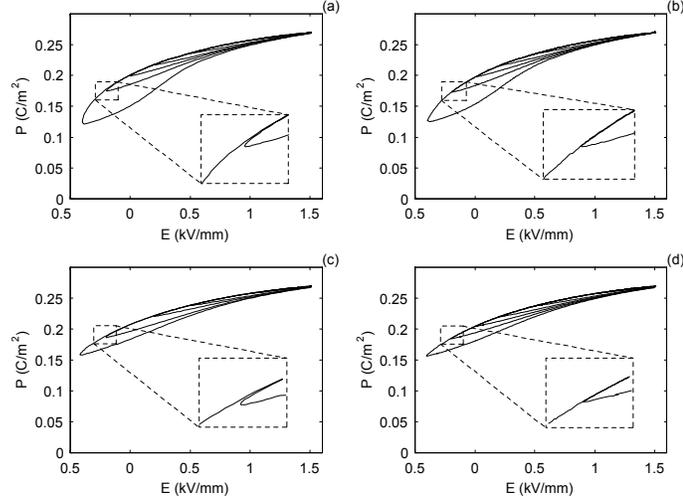


Figure 2.9: Minor loops type 1: Polarization. (a) Sinusoidal 0.1Hz; (b) Linear 0.1kV/mm/s; (c) Sinusoidal 100Hz; (d) Linear 100kV/mm/s.

Minor Loops Type 2

The second type of experiments are shown in Figure 2.11 and Figure 2.12. These curves all reverse from a point on the ascending loop and return to the lowest electric field at -0.4kV/mm . The electric field is then held at -0.4kV/mm for 10 seconds. Just as the first type in section 2.2.3 is intended to show the similarity and difference between the ascending minor loops, the second type is intended to show the features of the descending minor loops from different reversal points. This family of descending loops, as we can see in the figures, are very similar in shape.

The relaxation period at the end of these experiments exhibits a noticeable drop in polarization and strain, even for the curves at the lowest frequencies and rates. Therefore the lowest frequency of 0.1Hz is still not slow enough for a real quasistatic case, but it is quite close. We also observe that, although the low frequency curves start relaxing from a lower point and the high frequency curves start relaxing from a higher point, they all relax to almost the same level, indicating a fixed equilibrium point for each electric field.

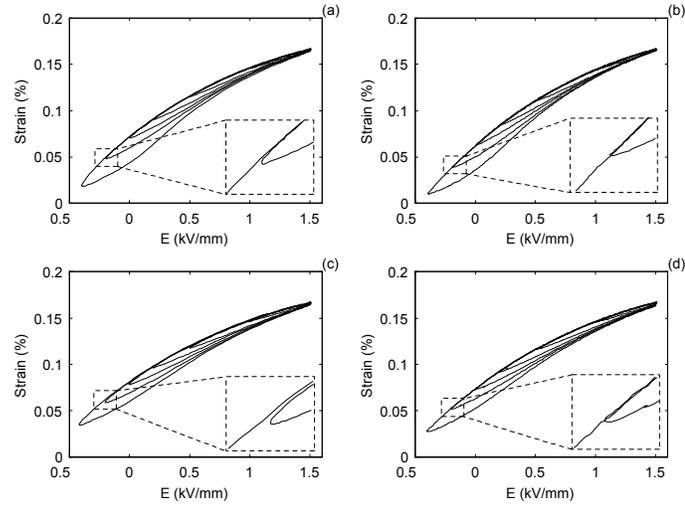


Figure 2.10: Minor loops type 1: Strain. (a) Sinusoidal 0.1Hz; (b) Linear 0.1kV/mm/s; (c) Sinusoidal 100Hz; (d) Linear 100kV/mm/s.

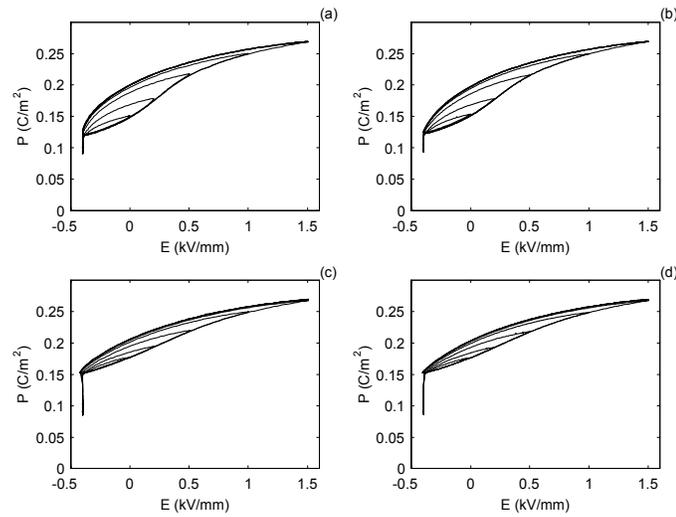


Figure 2.11: Minor loops type 2: Polarization. (a) Sinusoidal 0.1Hz; (b) Linear 0.1kV/mm/s; (c) Sinusoidal 100Hz; (d) Linear 100kV/mm/s.

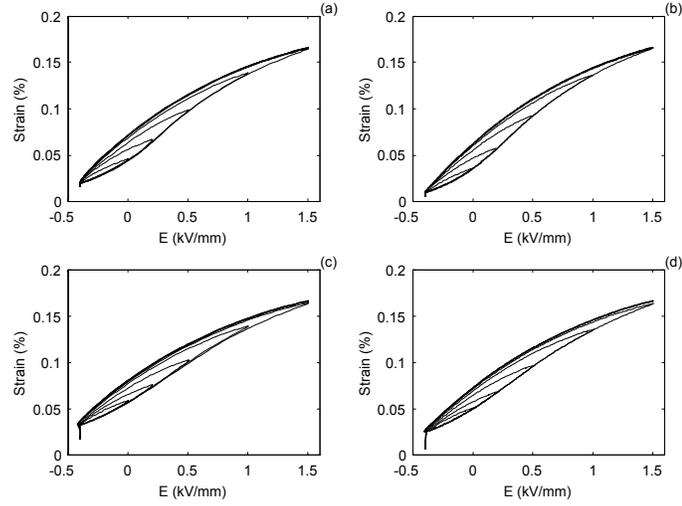


Figure 2.12: Minor loops type 2: Strain. (a) Sinusoidal 0.1Hz; (b) Linear 0.1kV/mm/s; (c) Sinusoidal 100Hz; (d) Linear 100kV/mm/s.

Minor Loops Type 3

The third type of loops are a bit more complex than the first two types. They first reverse from a point on the descending outer loop, then close a small minor loop and return to the first reversal point, as shown in Figure 2.13 and Figure 2.14. The field change for the small minor loops is 0.5kV/mm peak to peak. The field is finally held at the reversal point for 10 seconds.

The minor loops all take the shape of a weaving shuttle, although minor loops at higher fields are narrower. The relaxation behavior is similar to what has been observed in the previous type of experiments, but in this type of experiments, the constant electric fields are at different reversal points, allowing us to observe the relationship between the relaxation and the electric field. At the same loading frequency or rate, more relaxation is observed at lower reversal points, especially the -0.4kV/mm one, which is close to the negative coercive field of the material. For the same reversal point, more relaxation is observed after a faster loading cycle, because the domains of the actuator have less time to transform and will stop at a higher polarization level before the relaxation process. For the same reversal point, the actuator always relaxes to the same level, regardless of the frequency used for previous

loading before the 10-second holding period. This behavior will be discussed in more detail later with figures showing the relaxation process over time.

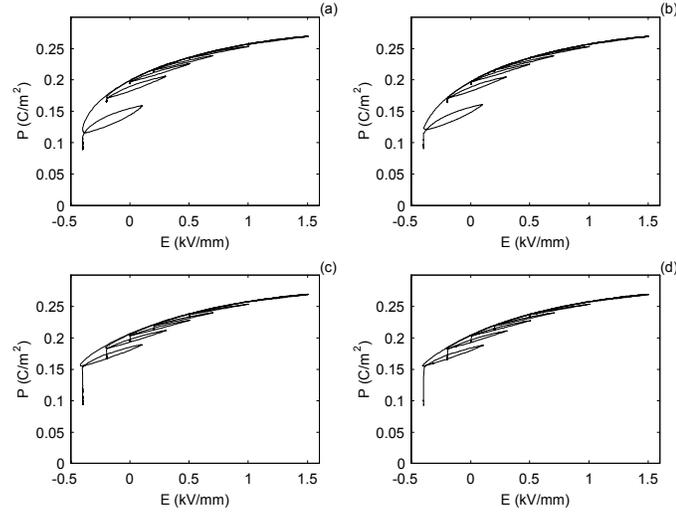


Figure 2.13: Minor loops type 3: Polarization. (a) Sinusoidal 0.1Hz; (b) Linear 0.1kV/mm/s; (c) Sinusoidal 100Hz; (d) Linear 100kV/mm/s.

Minor Loops Type 4

The fourth and last type of experiments is intended to show the opposite of the third type. The actuator reverses from a point at the ascending outer loop, completes a 0.5kV/mm peak to peak minor loop, and then returns to the reversal point on the ascending outer loop. The electric field is then held constant at the reversal point. Curves for different reversal points and loading frequencies and rates are shown in Figure 2.15 and Figure 2.16. Again, we can see that the shape of these minor loops look like a weaving shuttle and becomes narrower as the field of the reversal point increases. Unlike the previous type of experiments, the relaxation behavior is not very prominent on these ascending loops. The increase in polarization and strain after the 10-second holding period is hardly noticeable.

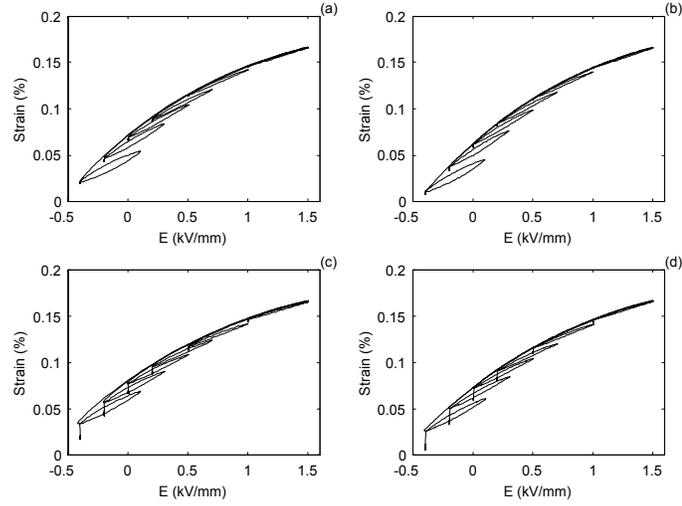


Figure 2.14: Minor loops type 3: Strain. (a) Sinusoidal 0.1Hz; (b) Linear 0.1kV/mm/s; (c) Sinusoidal 100Hz; (d) Linear 100kV/mm/s.

Relaxation Behavior

In Figures 2.13 and 2.14 of section 2.2.3 (Minor Loop Type 3), we have seen very noticeable relaxation during the 10-second constant field period at the end of each experiment. We have also seen that at the same reversal point, the curves almost always relax to the same final strain level. To better demonstrate this behavior, an example of the relaxation in strain is plotted over time in Figure 2.17. The constant field for these curves is fixed at -0.2 kV/mm. Subfigure (a) is the relaxation after sinusoidal loads at 0.1Hz and 100Hz; Subfigure (b) is the relaxation after linear loads at 0.1kV/mm/s and 100kV/mm/s. They show the same type of relaxation behavior. The experiment at a higher loading rate will have less phase transformation and thus start from a higher strain point. Then during the constant field period, the strain drops asymptotically towards an equilibrium and overlaps with the relaxation curve for the experiment at a lower rate.

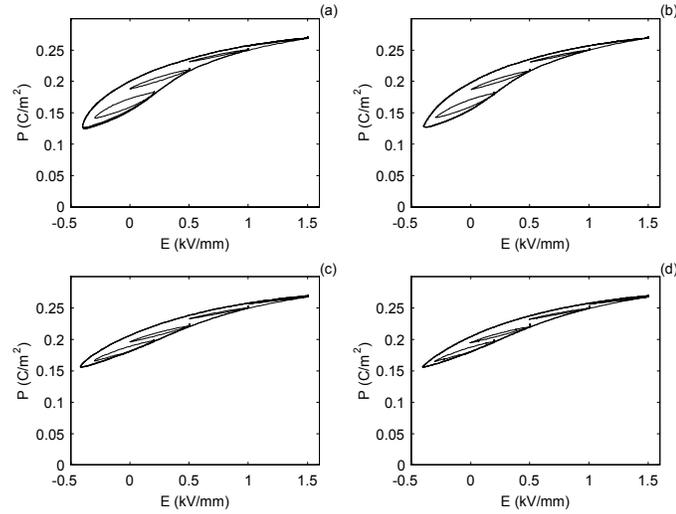


Figure 2.15: Minor loops type 4: Polarization. (a) Sinusoidal 0.1Hz; (b) Linear 0.1kV/mm/s; (c) Sinusoidal 100Hz; (d) Linear 100kV/mm/s.

2.3 Conclusion

In this chapter we have presented experimental results at different rates for the full outer hysteresis loop and four types of minor loops in the actuator range. From these experiments, we have observed that

1. The coercive field increases with the loading rate. This can also be expressed alternatively that at the same electric field, less phase transformation occurs at higher loading rates, indicating the existence of a relaxation time for the material;
2. Minor loops in the actuator range have very similar shapes, regardless of the location of reversal;
3. For a positively poled actuator, more relaxation is observed in a descending loop than an ascending loop.
4. At the same constant field level, regardless of the previous loading rate, the actuator relaxes asymptotically toward the same final strain level.

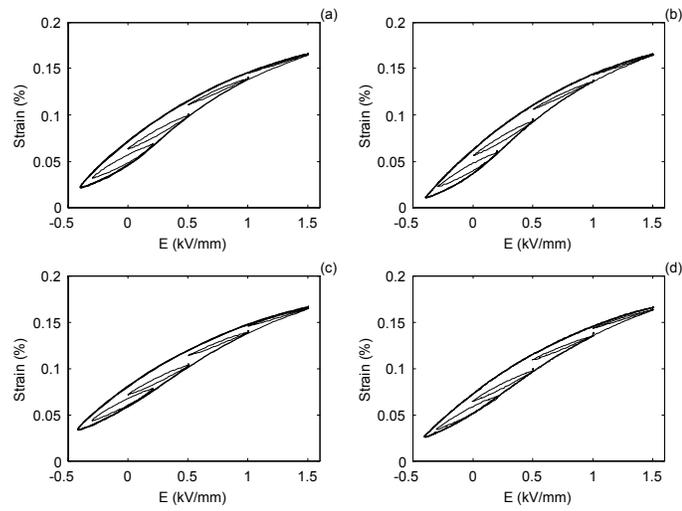


Figure 2.16: Minor loops type 4: Strain. (a) Sinusoidal 0.1Hz; (b) Linear 0.1kV/mm/s; (c) Sinusoidal 100Hz; (d) Linear 100kV/mm/s.

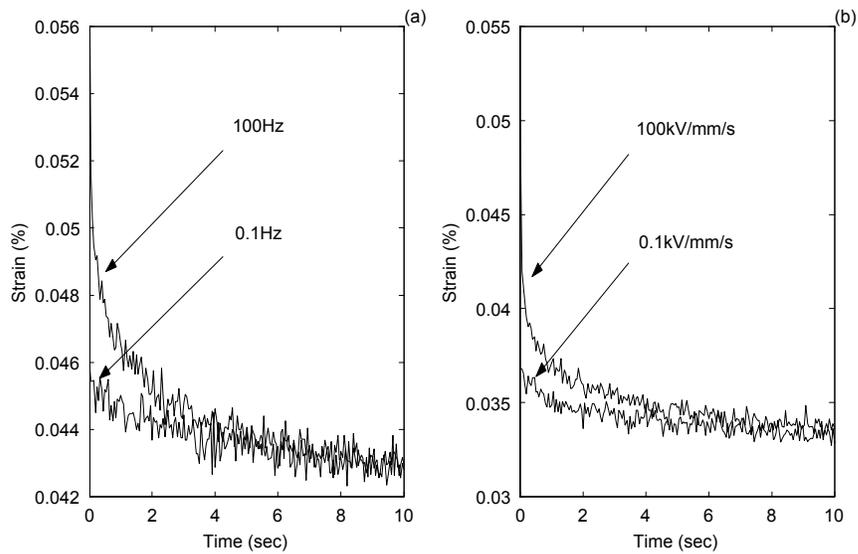


Figure 2.17: Relaxation: (a) After sinusoidal loads; (b) After linear loads.

Chapter 3

Modeling of Hysteresis

In this chapter, we try to model the hysteretic behavior of the piezoceramics presented in the previous chapters. We will begin with an introduction to a basic single crystal model for piezoceramics. With the physical principles established, we will extend the basic model to polycrystalline materials. Then a modified parameterization method will be proposed to improve the computational efficiency of the model.

3.1 Energy Model for Single Crystal PZTs

One of the central goals of this thesis is to model the rate-dependent behavior of piezoceramics. Therefore, the model must be based on physics as closely as possible, while being computationally simple enough for control applications. In this thesis, we start with a mesoscopic one-dimensional energy model proposed by Smith and Seelecke [38]. It is developed with the assumption that dipoles in a single crystal with uniform lattice spacing have two orientations, which are denoted by (+) and (-), depending on whether the Ti^{4+} ion is displaced upward or downward with respect to the center of symmetry (see Figure 3.1).

For a typical mesoscopic domain consisting of a number of equally poled lattice cells, the model assumes the following multiparabolic representation for the Helmholtz free energy density ψ as a function of the polarization P :

$$\psi(P) = \begin{cases} \frac{1}{2}E_1(P + P_T)^2 & , P \leq -P_I; \\ \frac{1}{2}E_1(P - P_T)^2 & , P \geq P_I; \\ \frac{1}{2}E_1(P_I - P_T) \left[\frac{P^2}{P_I} - P_T \right] & , |P| < P_I. \end{cases} \quad (3.1)$$

In Equation 3.1, P_T , P_I and E_1 are related to physical properties, which can be extracted from an electric field vs. polarization diagram. P_T denotes the remnant polarization, P_I the onset of the switching process, and E_1 is the reciprocal of the slope of the (E, P) curve (see the bottom row of Figure 3.2).

The switching processes in the material can be interpreted as phase transformations with the polarization as the order parameter. The kinetics of these phase transformations are determined by the electric-field-dependent barriers and minima in the Gibbs free energy, which is related to the Helmholtz free energy in Equation 3.1 through the Legendre transformation $G = \psi - EP$. This dependence is illustrated in Figure 3.2 for increasing field values (from left to right) together with the corresponding polarization vs. electric-field behavior.

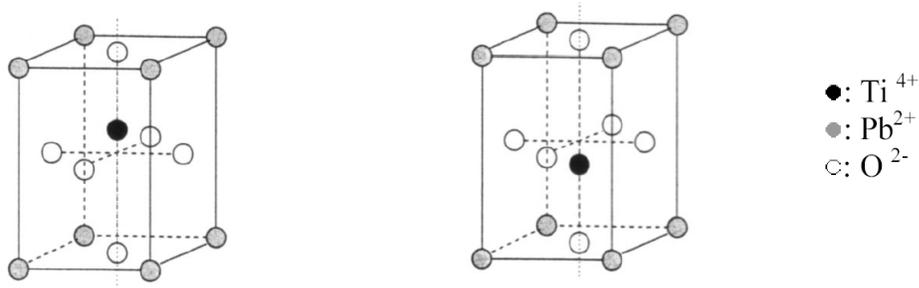


Figure 3.1: PbTiO_3 crystal lattice cells with positive and negative polarization

Due to the ever-present thermal activation, all domains fluctuate about their equilibrium polarization values. From Boltzmann principles, the probability of attaining an energy level G is

$$\mu(G) = Ce^{-G/kT}, \quad (3.2)$$

where k denotes Boltzmann's constant. The expected average polarizations due to domains with positive and negative polarizations are then given by

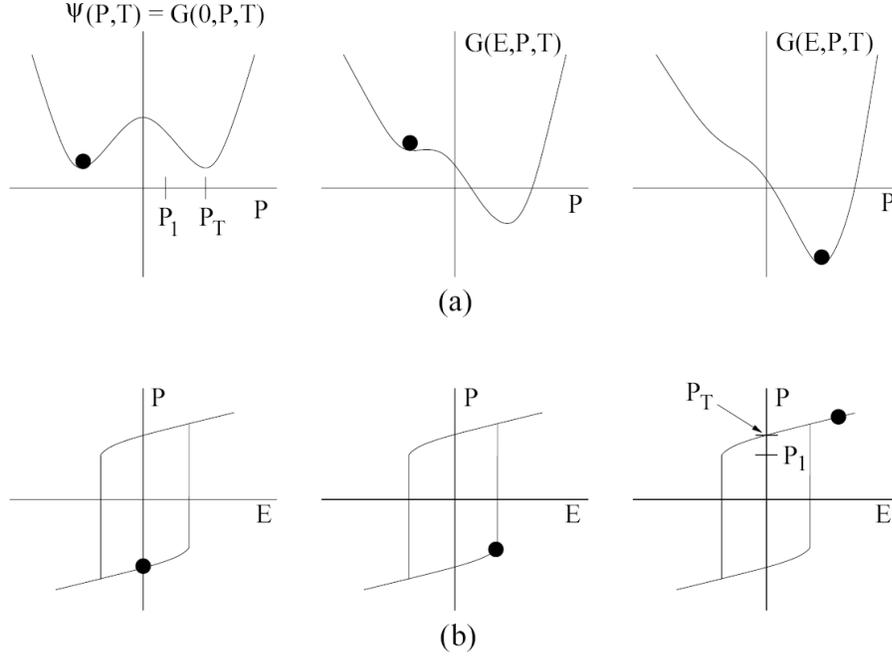


Figure 3.2: (a) Helmholtz energy ψ and Gibbs energy G for increasing field E ; (b) Polarization P as a function of E for a single crystal with uniform lattice.

$$\langle P_+ \rangle = \frac{\int_{P_1}^{\infty} P e^{-G(E, P, T)V_D/kT} dP}{\int_{P_1}^{\infty} e^{-G(E, P, T)V_D/kT} dP} \quad (3.3)$$

$$\langle P_- \rangle = \frac{\int_{-\infty}^{-P_1} P e^{-G(E, P, T)V_D/kT} dP}{\int_{-\infty}^{-P_1} e^{-G(E, P, T)V_D/kT} dP}, \quad (3.4)$$

and the average polarization for the whole actuator is

$$\bar{P} = x_+ \langle P_+ \rangle + x_- \langle P_- \rangle, \quad (3.5)$$

where x_+ and x_- denote the fractions of domains with positive or negative polarization, respectively. These phase fractions evolve in time according to the differential equations

$$\dot{x}_+ = -p_{+-}x_+ + p_{-+}x_- \quad (3.6)$$

$$\dot{x}_- = -p_{-+}x_- + p_{+-}x_+, \quad (3.7)$$

which can be simplified to

$$\dot{x}_+ = -p_{+-}x_+ + p_{-+}(1 - x_+) \quad (3.8)$$

through the identity $x_+ + x_- = 1$. Here p_{+-} and p_{-+} denote the transition probabilities of a domain switching from one polarization to the other. They are computed by

$$p_{+-} = \frac{1}{\tau_x} \frac{e^{-G(E, P_I(T), T)V_D/kT}}{\int_{P_I}^{\infty} e^{-G(E, P, T)V_D/kT} dP} \quad (3.9)$$

$$p_{-+} = \frac{1}{\tau_x} \frac{e^{-G(E, -P_I(T), T)V_D/kT}}{\int_{-\infty}^{-P_I} e^{-G(E, P, T)V_D/kT} dP}, \quad (3.10)$$

where τ_x is the relaxation time of the material and V_D denotes the activation volume of the domain. The ratio V_D/kT determines the fluctuation of polarization about the equilibrium values due to thermal activation, and together with τ_x , this ratio dictates the frequency dependence of the hysteresis loops. The solution of Equation (3.5) specifies the hysteretic relation between E and P , as is depicted in Figure 3.2.

3.2 Extension of the Model to Polycrystalline Materials

As the basic model describes perfect single crystal behavior, realistic actuators are neither perfect nor are they single crystals. Rather, they exhibit inhomogeneities due to lattice imperfections, impurities, grain boundaries, etc., which lead to variations in the energy barriers, depending on the location of the domain under consideration. In addition, the applied field in the material may vary as a result of the interaction between neighboring dipoles. Following the ideas outlined in [37] and [38], we take these variations into account by introducing distributions in the energy barriers and effective electric fields.

3.2.1 Distribution of Energy Barriers and Effective Electric Fields

A distribution is first introduced into the local coercive field $E_c = E_1(P_T - P_I)$. E_c is the measure of the energy barrier that needs to be overcome for a lattice to jump from one dipole configuration into the other. This barrier is equal for phase transformations in both directions and must take a positive value, so that the lattice will remain physically

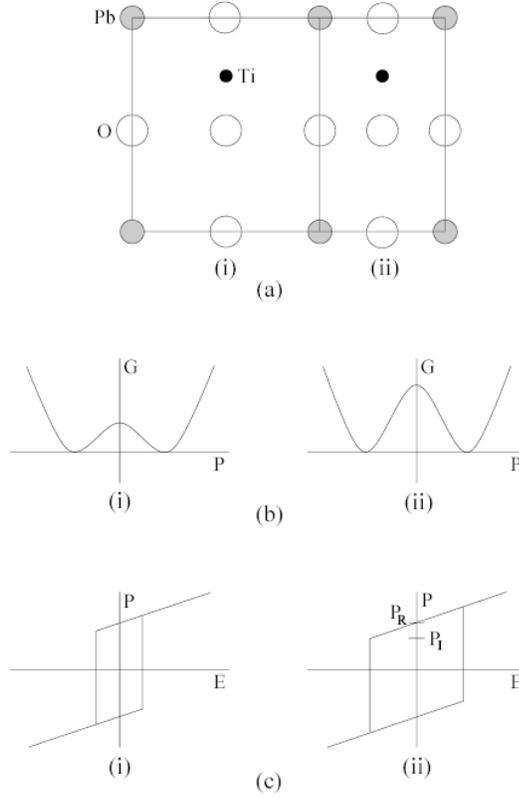


Figure 3.3: (a) Nonuniform PZT lattice structure; (b) Free energies associated with lattice widths (i) and (ii); (c) Variations in hysteresis kernel due to different free energies.

bistable. As seen in Figure 3.3 [38], the barrier dictates the width of a hysteresis kernel. A non-negative barrier guarantees that no kernels will have a negative width.

In particular, we consider E_c to be normally distributed with the mean value as \overline{E}_c . The total polarization is then given by

$$P(E) = \int_0^{\infty} \overline{P}(E, E_c) f(E_c) dE_c \quad (3.11)$$

with the probability

$$f_1(E_c) = C_1 e^{-(E_c - \overline{E}_c)^2 / b_1}, \quad (3.12)$$

where C_1 is a constant to ensure that

$$\int_0^{\infty} f_1(E_c) dE_c = 1.$$

The integration interval starts from zero, which corresponds to the requirement of non-negative barriers. Because of the clipping at $E_c = 0$, the center of the normal distribution \bar{E}_c is no longer the mean value of E_c , but it remains the point with the highest transformation probability, where the sharpest slope occurs on a hysteresis curve. Therefore an initial estimation of \bar{E}_c can be obtained by reading this value from the experimental data of a complete outerloop.

Although the distribution in E_c allows the modeled material to transform gradually near E_c , it does not allow the material to start transforming before the applied electric field reaches zero, which is not consistent with experimental observations. The distribution of E_c alone is not sufficient for modeling real materials. Besides the coercive field E_c , the second most important quantity in the model is the applied field E . When an external electric field is applied to the material, the fields generated by neighboring dipoles will vary the effective field applied to a local domain inside the material. This variation in effective field can be incorporated by adding a second distribution as we did for the coercive field.

Again, we consider the effective field to be normally distributed about the externally applied field \bar{E} . If we only consider this distribution and take E_c as a fixed value, the total polarization becomes

$$P(E) = \int_{-\infty}^{\infty} \bar{P}(E + E_e, E_c) f_2(E_e) dE_e \quad (3.13)$$

with the probability

$$f_2(E_e) = C_2 e^{-(E_e)^2/b_2}, \quad (3.14)$$

where C_2 is a constant to ensure that

$$\int_{-\infty}^{\infty} f_2(E_e) dE_e = 1.$$

This second distribution in the applied field causes some domains to have an effective field beyond zero and start their switching process before the external field reaches zero. If we look at the corresponding hysteresis kernels, the effective field applied on a domain will

offset the center of its hysteresis kernel from zero. Together with the first distribution in E_c , it allows the model to capture the aforementioned experimental observations. Combining two distributions in the model, the complete polarization for polycrystalline materials with varying effective fields becomes

$$P(E) = C \int_0^\infty \int_{-\infty}^\infty \bar{P}(E + E_e, E_c) e^{-(E_c - \bar{E}_c)^2/b_1} e^{-(E_e)^2/b_2} dE_e dE_c \quad (3.15)$$

where C is a constant to ensure that the integration of distributions equals one.

3.2.2 Implementation of Distributions

In order to implement the model described by Equation 3.15, the integrals can be approximated with quadrature methods. A Gaussian quadrature takes the form

$$S = \sum_{i=1}^n w_i f(x_i)$$

where w_i are the weights and x_i are the abscissae. The total polarization is then approximated as

$$P(E) = C \sum_{i=1}^{n_i} \sum_{j=1}^{n_j} \bar{P}(E + E_{e_j}, E_{c_i}) e^{-(E_{c_i} - \bar{E}_c)^2/b_1} e^{-(E_{e_j})^2/b_2} w_j E_{e_j} v_i E_{c_i}. \quad (3.16)$$

To evaluate the integrals, we may use the Gauss-Hermite quadrature on an infinite domain, the Gauss-Laguerre quadrature on a semi-infinite domain, or the Gauss-Legendre quadrature on a finite domain. Here we chose the Gauss-Legendre quadrature over other quadrature methods because of two major reasons. First, the abscissae calculated from Gauss-Legendre are more evenly distributed on the domain of our interest and thus produce smoother curves and better inner-loop behavior than other quadratures with the same number of abscissae. Secondly, Gauss-Legendre makes it possible for us to divide the integration range into subintervals and have greater flexibility on the number and location of abscissae within a specific interval, e.g. the peak of the distribution. Although the Gauss-Legendre quadrature is appropriate only on a finite domain, the decaying nature of the distributions allows truncation of their domains to finite intervals with tolerable error.

In the calculations, the integration domains are truncated at their 3σ point (only 0.3% of the unclipped integral over the infinite domain will be outside this point for a normal distribution) and then divided into two subintervals which are separated at the peak of the distribution. Figure 3.4 and Figure 3.5 shows examples of the distribution density function and the abscissae by Gauss-Legendre quadrature.

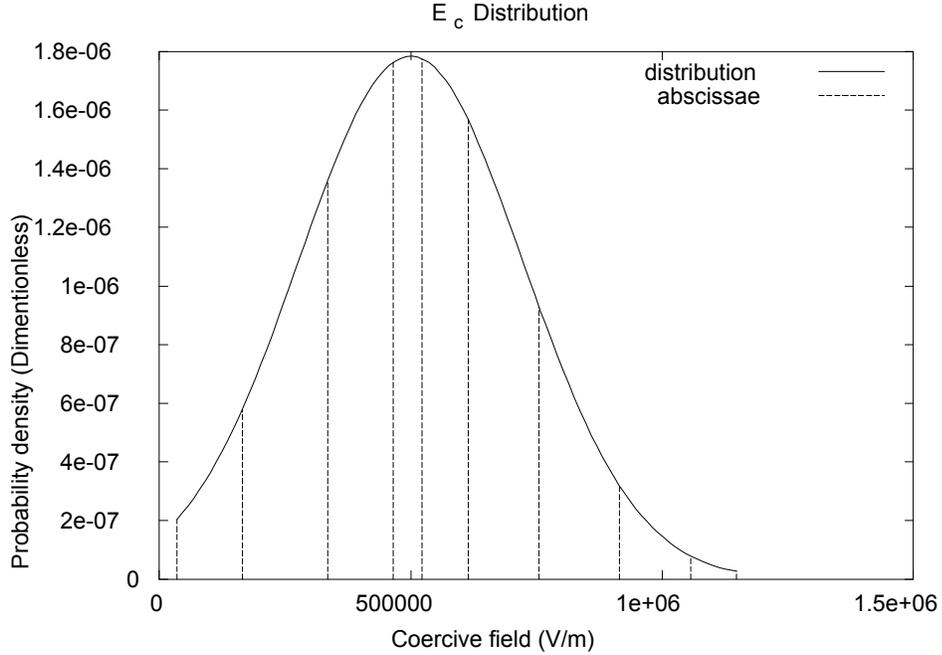


Figure 3.4: E_c Distribution, $\bar{E}_c = 0.5 \times 10^6$, $b_1 = 1.0 \times 10^{11}$, 12 total abscissae points.

For each combination of abscissae E_{e_j} and E_{c_i} , the term $\bar{P}(E + E_{e_j}, E_{c_i})$ in Equation 3.16 must be evaluated with an ODE. If we take n_i abscissae for E_c and n_j abscissae for E_e , a total of $n_i \times n_j$ ODEs have to be solved simultaneously, which is very computationally expensive when $n_i \times n_j$ becomes large. Therefore a smaller number of abscissae is preferred when the precision of the overall integration is within an acceptable range.

In the Fortran implementation, the weights and abscissae are calculated using the NAG library routine D01BBF[24]. A more detailed introduction to the Legendre polynomials used in the calculation can be found in related documentations of [24]; the Fortran program used for the computation is included in the appendix of the thesis.

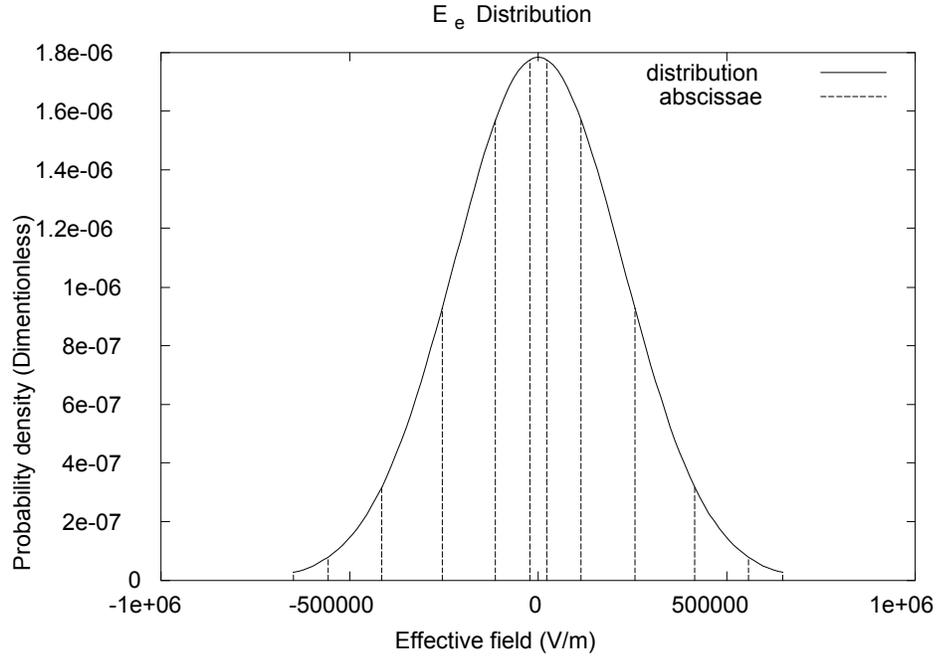


Figure 3.5: E_e Distribution, $b_2 = 1.0 \times 10^{11}$, 12 total abscissae points.

3.2.3 Simulation and discussion

Using the quadrature method in the previous section, we apply different combinations of distribution parameters and verify the flexibility of the model. Figure 3.6 shows a representative example of the model simulation. The model parameters are $V_D = 1.0 \times 10^{-22} \text{ m}^3$, $\tau_x = 1.0 \times 10^{-3} \text{ s}^{-1}$, $P_T = 0.26 \text{ C/m}^2$, $E_1 = 6.0 \times 10^7 \text{ Vm/C}$, $\overline{E}_c = 0.5 \times 10^6 \text{ V/m}$, $b_1 = 5.0 \times 10^{11}$, $b_2 = 5.0 \times 10^{11}$. Piecewise linear electric loads are applied at a rate of 10 MV/m/s. These loads are chosen to illustrate the ability of the model to reproduce symmetric unbiased minor loops, biased asymmetric minor loops and closure of minor loops.

Although the simulation shown in Figure 3.6 demonstrates the capability of the model, it is not easy to see how the choice of distribution parameters would change the shape of the major and minor loops. To show the influence of the two distributions, a series of simulations are done with combinations of different distribution parameters. For the coercive field, a fixed mean value is taken to be $\overline{E}_c = 0.5 \times 10^6 \text{ V/m}$, while b_1 changes from

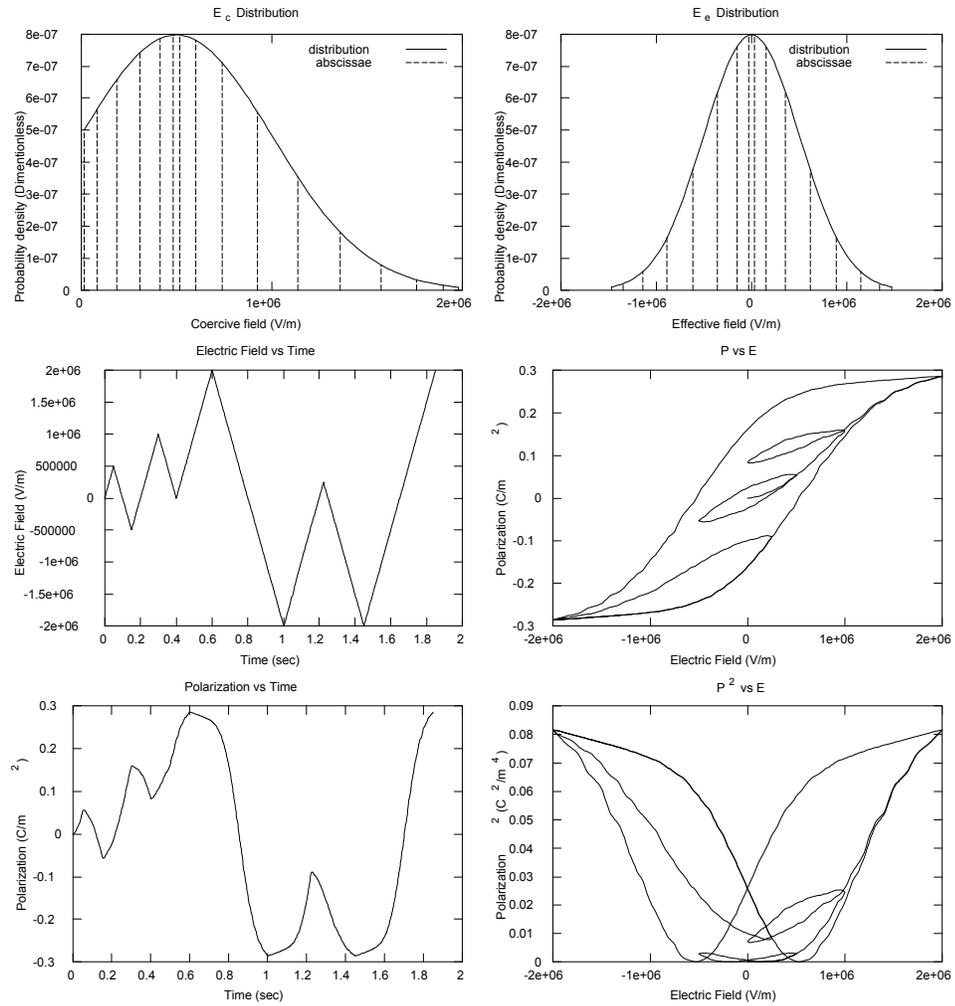


Figure 3.6: Simulation of PZT polarization using the quadrature method.

1.0×10^{10} to 1.0×10^{11} to 1.0×10^{12} . For the effective field, b_2 changes from 1.0×10^{10} to 1.0×10^{11} to 1.0×10^{12} . Altogether $3 \times 3 = 9$ simulations are computed, as shown in Figure 3.7. All other physical parameters are the same as those used in the previous example. The simulations on the same column have the same distribution in E_c and the simulations on the same row take the same distribution in E_e .

From Figure 3.7, we can now clearly see the different influences of the two distributions. They can both transform the major loop into a more gradual process when b is sufficiently large and the distribution is flat, but they have completely different effects on the minor loops. With a fixed E_e , the distribution in E_c determines how the width of hysteresis kernels changes along with the phase fractions and dictates the shape of the innerloops after the first coercive field is passed. This is best shown in the top row of Figure 3.7, where the distribution in E_c is dominant. Very little transformation starts before the applied electric field passes zero.

The distribution in E_e , on the other hand, mainly determines the onset of transformation before the applied field passes zero. The effective field E_e , as we discussed in section 3.2.1, offsets the center of the hysteresis kernel for a domain and allows this domain to transform before the external field passes its coercive field. In the first column of Figure 3.7, E_c takes a fixed, narrow distribution and the shapes of the loops are dominated by the distribution in E_e . The switching process of the minor loops almost never starts before the applied field reaches the major loop.

With the two distributions, we now have not only greater control of the major loop, but also the flexibility to change the minor-loop behavior. The model is capable of reproducing minor-loop behavior commonly seen in real-world polycrystalline materials. As illustrated in [38], which uses an almost identical model (Log normal instead of clipped normal distribution for E_c) and a simplified implementation, the model can be used to fit many types of PZT materials. Fitting to materials with unusual majorloop shape will also be possible by using different distribution functions.

Although the model has all these advantages, there is a serious drawback that prevents it from being used in control applications. The requirement of solving an ODE for each combination of E_c and E_e in the quadrature equation makes it too computationally expensive. To obtain a reasonably smooth simulation curve, hundreds of ODEs must be solved

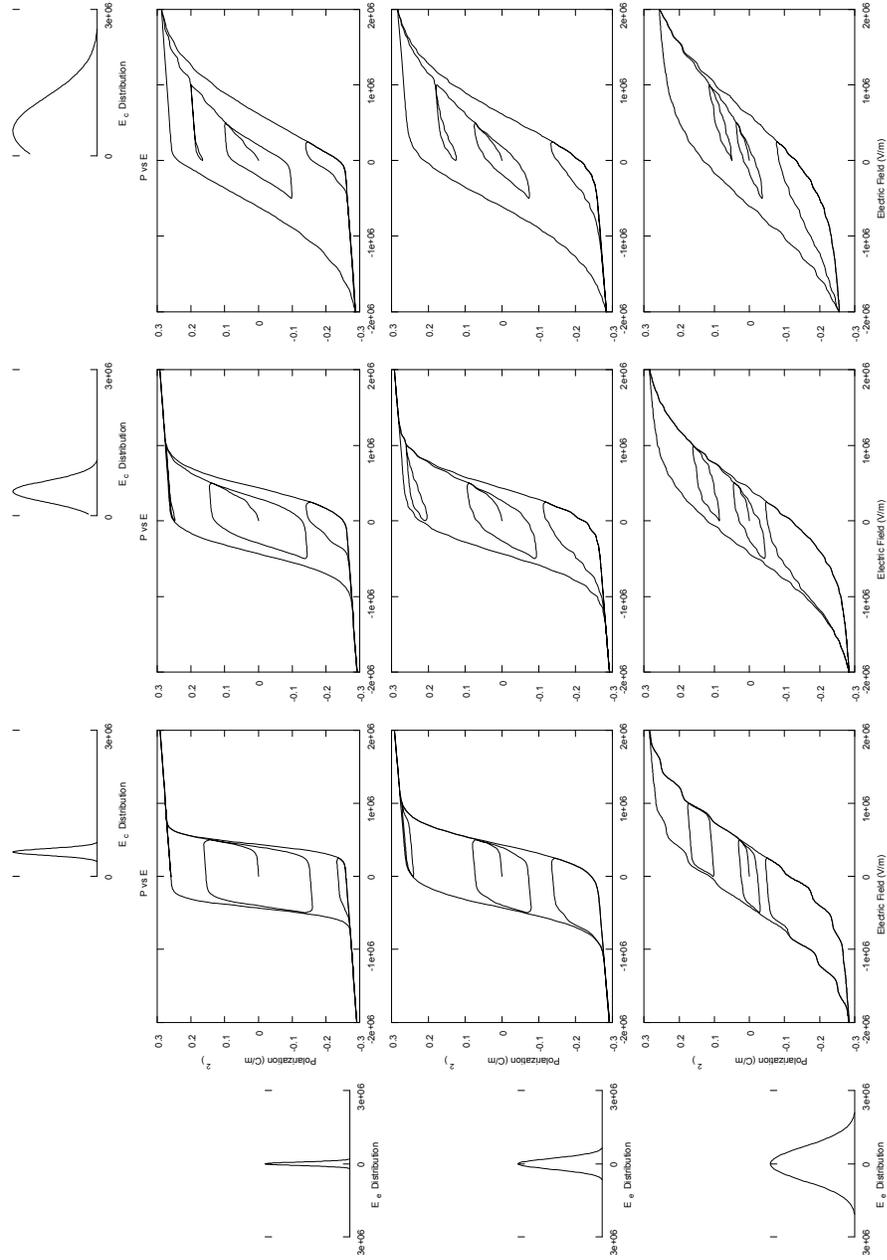


Figure 3.7: Polarization simulated with 9 different combinations of distribution parameters.

together and typically take tens of minutes. [38] uses a simplified implementation of the model, which is easier to evaluate and faster but loses the dynamic characteristics of the energy model, making it suitable only for low thermal activation and low frequencies. To make the model more efficient while retaining the dynamic capability of the energy model, an alternative to the quadrature method must be used. In the next section, we will propose a method to incorporate the distributions by parametrization.

3.3 Parametrization Method

As we have discussed in the previous section, the quadrature method discretizes the two distributions, and according to the number of integration points, it requires the simultaneous solution of many ODEs, each representing an element with distinct energy barrier and effective field. The associated computational cost prohibits its use in real-world control applications. To reduce the computation time, we will switch to an alternative method to implement the distributions into the model, namely the parametrization method.

For the quadrature method, we consider many domains of elements with each element having a fixed barrier and a fixed effective field. The parametrization method starts with a different perspective. Instead of looking at an ensemble of many physical elements, we now consider what we call a "representative" element. This is the element which is always next to undergo the phase transformation; and depending on how many elements have already transformed, this element will see a constantly changing barrier. With this in mind, we now parametrize the process of phase transformation and calculate the barrier from the current phase fraction, eliminating the need to solve many ODEs for different domains and reducing the implementation to only one ODE for the representative domain. Thus, in essence, we retain the level of complexity of the perfect single crystal version.

In the following sections, we will present the derivation of the parametrized model. To begin with a simpler problem, we first parametrize the version with only one distribution in the coercive field E_c , and subsequently the version with only one distribution in the effective field E_e . After explaining the differences between these two versions, as we have seen in the quadrature simulations, we will combine the coercive field and the effective field into a unified version.

3.3.1 Parametrization of the Distribution in E_c

Before we derive the relationship between the coercive field E_c and the phase fraction x_+ , we shall review the model with only one distribution in E_c . The average polarization of the material is

$$P(E) = \int_0^\infty \bar{P}(E, E_c) f_1(E_c) dE_c \quad (3.17)$$

with the probability

$$f_1(E_c) = C_1 e^{-(E_c - \bar{E}_c)^2 / b_1}, \quad (3.18)$$

where C_1 is a constant to ensure that

$$\int_0^\infty f_1(E_c) dE_c = 1.$$

In particular, if no truncation is applied to the above domain of integration,

$$C_1 = \frac{2}{\sqrt{b_1 \pi} \operatorname{erfc}(-\bar{E}_c / \sqrt{b_1})}, \quad (3.19)$$

where $\operatorname{erfc}()$ is the so-called error function.

If we only consider one element, the polarization can be rewritten as

$$P(E) = \frac{E}{E_1} + (2x_+ - 1) P_T. \quad (3.20)$$

The first term of the equation is only dependent on the applied field and is considered a reversible or spontaneous part of the polarization. The second term of the equation is dependent on the phase fraction and constitutes the irreversible part of the polarization. From now on, we focus on the phase fraction part and derive its relationship with the energy barrier.

We start by looking at the evolution of phase fraction versus energy barrier on an ascending loop. As a certain value of coercive field \bar{E}_c is passed and the domains with this particular \bar{E}_c are transformed, the increase in barrier dE_c will be accompanied by an increase in phase fraction, which is dictated by the probability function $f(E_c)$,

$$dx_+ = f_1(E_c)dE_c. \quad (3.21)$$

If we consider a general case in which the material has an initial phase fraction of x_+^0 and an associated initial barrier of E_c^0 , integrating Equation 3.21 from E_c^0 to $E_c > E_c^0$ yields the change in phase fraction

$$\begin{aligned} \Delta x_+(E_c) &= \int_{E_c^0}^{E_c} f_1(E'_c) dE'_c \\ &= \int_{E_c^0}^{E_c} \frac{2}{\sqrt{b_1}\pi \operatorname{erfc}(-\bar{E}_c/\sqrt{b_1})} e^{-(E'_c - \bar{E}_c)^2/b_1} dE'_c \\ &= \left[\operatorname{erfc}\left(\frac{\bar{E}_c - E_c}{\sqrt{b_1}}\right) - \operatorname{erfc}\left(\frac{\bar{E}_c - E_c^0}{\sqrt{b_1}}\right) \right] / \operatorname{erfc}\left(\frac{-\bar{E}_c}{\sqrt{b_1}}\right). \end{aligned} \quad (3.22)$$

Equation 3.22 describes how the phase fraction will change as a certain value of E_c is passed, as is illustrated in Figure 3.8. The darkest area is the initial phase fraction x_+^0 , and the area between E_c^0 and E_c is the change in phase fraction Δx_+ . Inverting Equation 3.22 yields

$$E_c(\Delta x_+) = \bar{E}_c - \sqrt{b_1} \operatorname{inverfc}\left(\Delta x_+ \operatorname{erfc}\left(\frac{-\bar{E}_c}{\sqrt{b_1}}\right) + \operatorname{erfc}\left(\frac{\bar{E}_c - E_c^0}{\sqrt{b_1}}\right)\right), \quad (3.23)$$

where $\operatorname{inverfc}()$ is the so-called inverse complementary error function.

Equation 3.23 enables us to parametrize the process of phase transition. Figure 3.9 shows an illustration of the curve defined by the equation. In the simplest case of an ascending half major loop, where $x_+^0 = 0$ and $E_c^0 = 0$, Δx_+ is the same as x_+ and Equation 3.23 simplifies to

$$E_c(x_+) = \bar{E}_c - \sqrt{b_1} \operatorname{inverfc}\left(x_+ \operatorname{erfc}\left(\frac{-\bar{E}_c}{\sqrt{b_1}}\right) + \operatorname{erfc}\left(\frac{\bar{E}_c}{\sqrt{b_1}}\right)\right), \quad (3.24)$$

which shows that the coercive field changes continuously from 0 to ∞ as the phase fraction x_+ increases from 0 to 1.

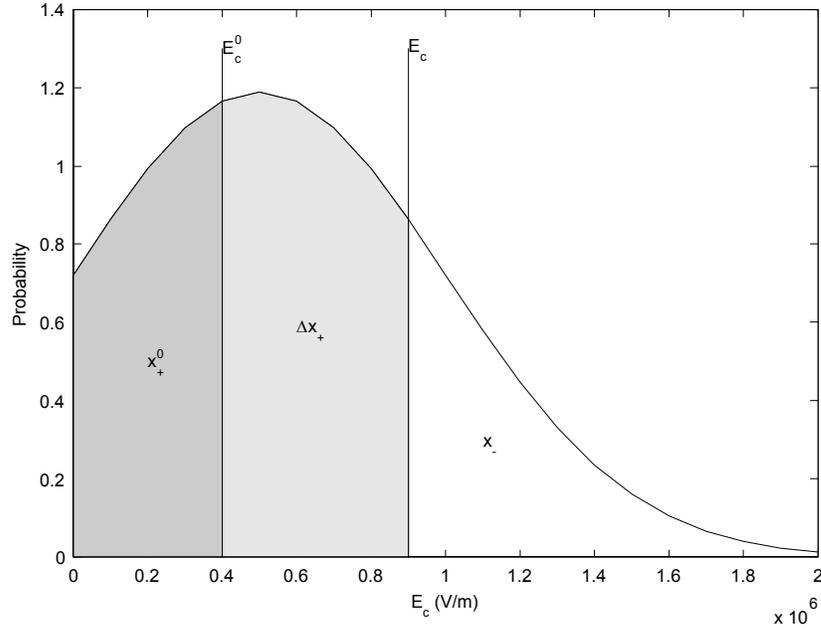


Figure 3.8: Evolution of E_c during a transformation process.

For the descending loop, the barrier can be obtained by substituting $x_- = 1 - x_+$ for x_+ and reversing the sign of Equation 3.24. By substituting $E_c(x_+)$ into the perfect single crystal ODE, we can now introduce the distribution into the model without using a quadrature and discretizing the process. Another important feature of a hysteresis model is its ability to predict proper minor loops. For the quadrature method, minor loops are automatically guaranteed because of its discretized nature and no minor-loop return points need to be tracked. For the parametrized method, different parts of minor loops take different initial conditions of x_+^0 , so the phase fraction at the minor-loop return points must be tracked in order to calculate the correct coercive field using Equation 3.23. In the Fortran implementation, this is done by using two data arrays storing respectively the phase fractions at the return points on ascending and descending loops. Detailed Fortran code can be found in the appendices of the thesis. The simulated polarization with major and minor loops using the complete implementation of Equation 3.23 is shown in Figure 3.10(left) against the same simulation using the quadrature method in Figure 3.10(right). Both versions produce

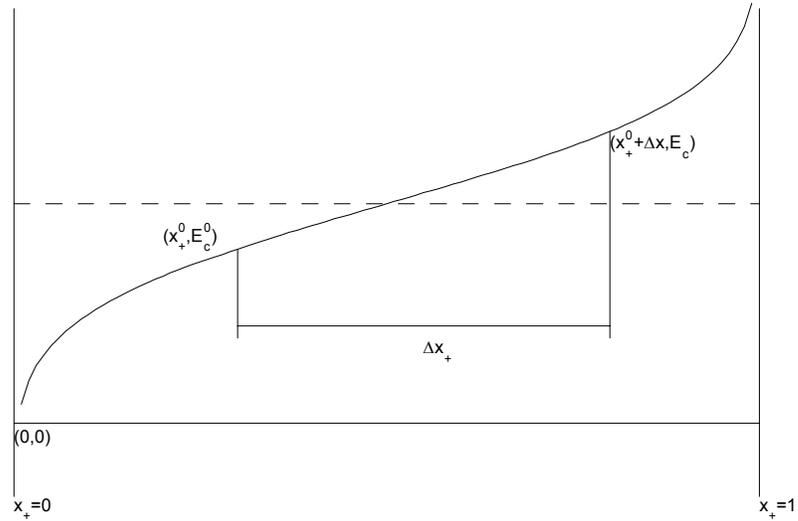


Figure 3.9: Schematic of $E_c(\Delta x_+)$. (Dashed line: $E = \overline{E_c}$)

almost the same hysteresis loops, which validates the feasibility of parametrization. The parametrized version takes only 0.04 second, while the quadrature version with 32 elements needs 9.4 seconds to compute the solution and yet produces less smooth results.

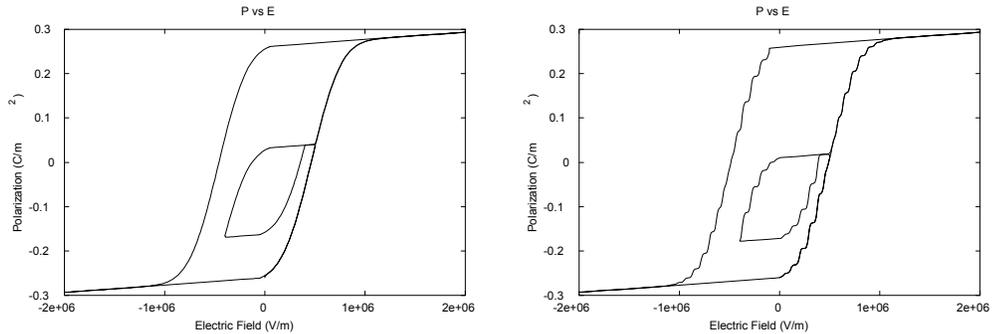


Figure 3.10: Left: Parametrized version; Right: Quadrature version.

3.3.2 Parametrization of the Distribution in E_e

Another distribution that we introduced in the model is the distribution in the effective field E_e . In addition to the externally applied field, a local domain will be imposed an effective

field from its neighboring dipoles. With a fixed E_c , we consider only the distribution in E_e . Again we start by looking at the evolution of phase fraction versus effective field on an ascending loop. A domain with a larger positive effective field will transform before the domains with smaller effective fields. As the domains with a certain effective field E_e are transformed, the decrease in barrier $-dE_e$ will be accompanied by an increase in phase fraction, which is dictated by the probability function $f(E_e)$,

$$dx_+ = -f_2(E_e)dE_e, \quad (3.25)$$

in which the probability function

$$f_2(E_e) = C_2 e^{-(E_e)^2/b_2}, \quad (3.26)$$

where C_2 is a constant to ensure that

$$\int_{-\infty}^{\infty} f_2(E_e)dE_e = 1.$$

In particular, if no truncation is applied to the above infinite domain of integration,

$$C_2 = \frac{1}{\sqrt{b_2\pi}}. \quad (3.27)$$

If we consider a general case in which the material has an initial phase fraction of x_+^0 and an associated initial barrier of E_e^0 , integrating Equation 3.25 from E_e^0 to $E_e < E_e^0$ yields the change in phase fraction

$$\Delta x_+(E_e) = - \int_{E_e^0}^{E_e} f_2(E_e') dE_e' \quad (3.28)$$

$$\begin{aligned} &= - \int_{E_e^0}^{E_e} \frac{1}{\sqrt{b_2\pi}} e^{-(E_e')^2/b_2} dE_e' \\ &= \int_{E_e}^{E_e^0} \frac{1}{\sqrt{b_2\pi}} e^{-(E_e')^2/b_2} dE_e' \quad (3.29) \\ &= \frac{1}{2} \left[\operatorname{erfc} \left(\frac{E_e}{\sqrt{b_2}} \right) - \operatorname{erfc} \left(\frac{E_e^0}{\sqrt{b_2}} \right) \right]. \end{aligned}$$

Equation 3.28 describes how the phase fraction will change along with the effective field for the representative domain, which is illustrated in Figure 3.11. The darkest area is the initial phase fraction x_+^0 , and the area between E_e^0 and E_e is the change in phase fraction Δx_+ . Inverting Equation 3.28 yields

$$E_e(\Delta x_+) = \sqrt{b_2} \operatorname{inverfc} \left(2\Delta x_+ + \operatorname{erfc} \left(\frac{E_e^0}{\sqrt{b_2}} \right) \right). \quad (3.30)$$

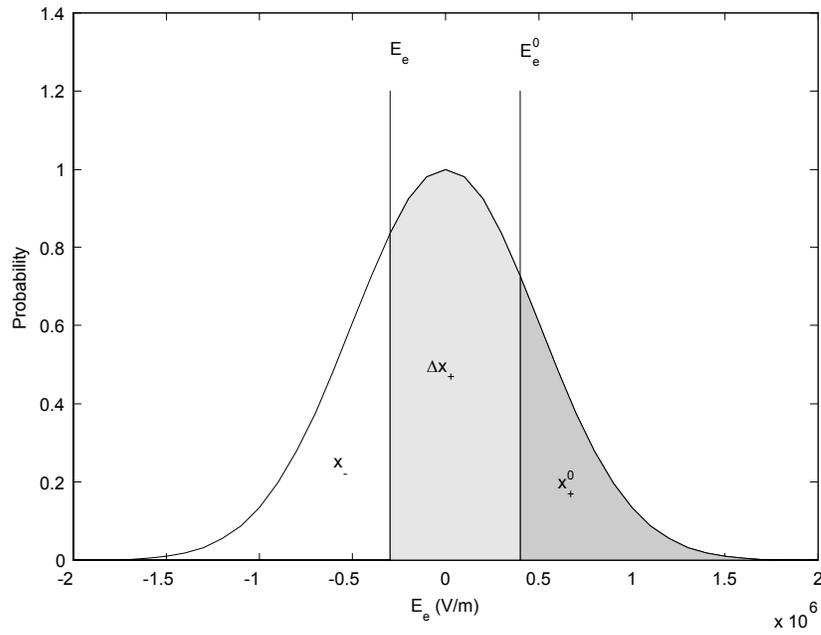


Figure 3.11: Evolution of E_e during a transformation process.

Equation 3.30 enables the parametrization of phase transformation process with respect to the effective field. Figure 3.12 shows an illustration of the curve defined by the equation. In the simplest case of an ascending half major loop, where $x_+^0 = 0$ and $E_e^0 = \infty$, Δx_+ is the same as x_+ and Equation 3.30 simplifies to

$$E_e(x_+) = \sqrt{b_2} \operatorname{inverfc}(2x_+), \quad (3.31)$$

which shows that the effective field changes from ∞ to $-\infty$ as the phase fraction x_+ increases from 0 to 1.

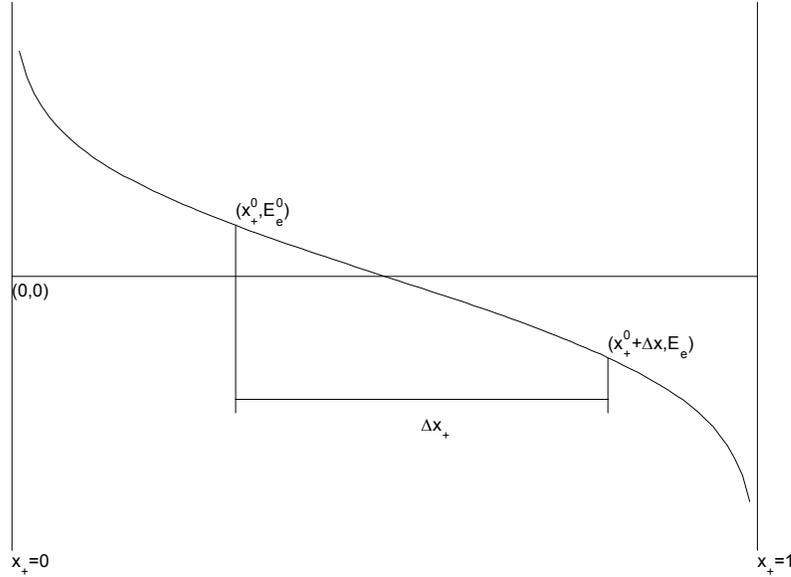


Figure 3.12: Schematic of $E_e(\Delta x_+)$.

For the descending loop, the barrier can be obtained by substituting $x_- = 1 - x_+$ for x_+ and reversing the sign of Equation 3.31. By substituting $E_e(x_+)$ into the perfect single crystal ODE, the distribution of effective field is introduced into the model without using a quadrature and discretizing the process.

As we have observed in the quadrature simulations, the distribution in E_e only changes the onset of phase transformations. If we consider two domains with different effective fields, the one with a larger E_e will transform first in an ascending loop but will transform last in a descending loop. For a reversal at any overall phase fraction, the ascending barrier and the descending barrier of the "representative" domain retain a constant difference of $2E_c$, where E_c is the fixed coercive field. The same overall phase fraction always maps to the same ascending and descending barriers. There is no need to distinguish between major loops and minor loops or different initial conditions. Or in other words, special minor loops do not really exist if only the distribution in E_e is present. This is completely different from the case with only one distribution in E_c . For the E_c version, the reverse barrier after any reversal will always start from zero, which demands the tracking of history points.

Without storing reversal points, the version with a fixed E_c and a distribution in E_e can be easily implemented by substituting Equation 3.31 into the perfect single crystal ODE. The simulated polarization with major and minor loops using Equation 3.31 is shown in Figure 3.13(left) against the same simulation using the quadrature method in Figure 3.13(right). Both versions produce almost the same hysteresis loops. In fact, the two results should look exactly the same if sufficiently many elements are used in the quadrature to approximate the integral. The parametrized version takes only 0.04 second, while the quadrature version with 32 elements needs 82 seconds to compute the same solution curve with noticeable jaggedness.

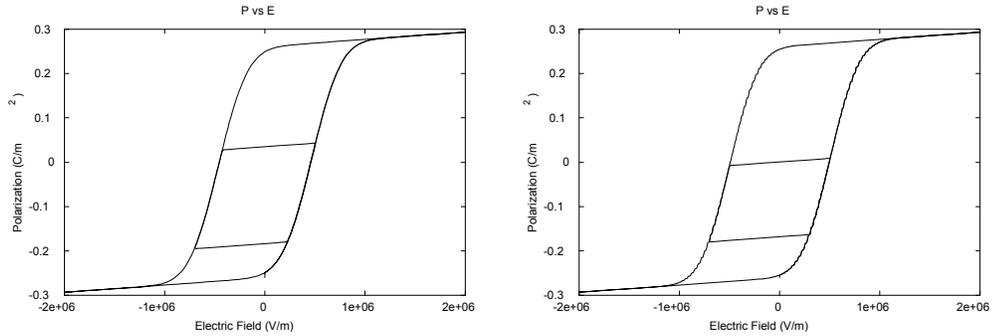


Figure 3.13: Left: Parametrized version; Right: Quadrature version.

3.3.3 Combination of Two Distributions

Following the idea that we used to parametrize each of the two distributions, we now apply the same procedure to a phase transformation process with both distributions present. When both distributions are combined, the switching of domains becomes more complicated. Before we establish the equations, we shall revisit the physical landscape behind the switching process. Figure 3.14 shows $3 \times 3 = 9$ domains with representative combinations of energy barriers and effective fields from all the possible domains. Domains shown on the same column have the same coercive field, while domains on the same row has the same effective field.

Domains shown in the middle row have zero effective field. They have equal barriers for transformations in both directions. Domains in the top row are the same domains

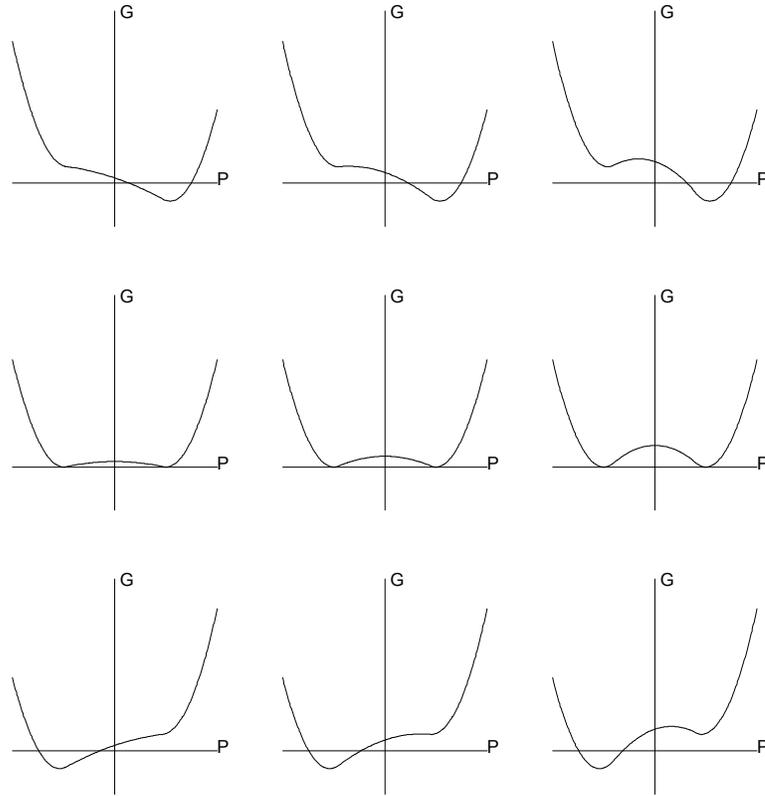


Figure 3.14: Domains with different combinations of energy barriers and effective fields.

with a positive effective field imposed, which practically alters the balance in their Gibbs energy even with the absence of an external field. The altered balance now favors the switching from minus to plus, and the particles in these domains are more prone to rest in a configuration of positive polarization. The domains have effectively a smaller barrier for a forward transformation (a transformation from minus to plus) and a larger barrier for a backward transformation (a transformation from plus to minus). Domains in the bottom row, on the other hand, are imposed a negative effective field and have their energy balance altered to favor the transformation from minus to plus.

From the Gibbs free energy of the domains in Figure 3.14, we can make several observations: First, the domain with the minimum energy barrier and maximum positive effective field (the upper left one) will transform from a negative phase into a positive phase before

all the other domains, while the domain with the maximum negative effective field (lower left) will transform first from a positive phase into a negative phase; Second, it is possible for a domain with a small barrier and a small effective field to transform together with a domain with a large barrier and large effective field, if their altered barriers happen to be the same; Third, because of the effective field, it is possible for a domain to lose one of its energy minima and have all its particles resting in the other minimum when no external field is applied yet. The third observation actually implies the existence of negative barriers, which were not allowed in earlier versions of the model.

With the energy landscapes illustrated, we now return to the notion of coercive field E_c and effect field E_e . Figure 3.15 shows the hysteresis kernels reflecting the energy landscapes in Figure 3.14. Although negative barriers are introduced for forward transformations, all the kernels still have a positive width (defined as the forward coercive field minus the backward coercive field). In Equation 3.9, the kinetics of domain switching from minus to plus is determined by the term $(E + E_e) - E_c$. We have shown in the previous paragraph that the effective field can be combined with the energy barrier, which we quantify as the coercive field E_c , so we now rewrite $(E + E_e) - E_c$ as $E - (E_c - E_e)$ and denote $(E_c - E_e)$ as what we call an effective barrier B_e . The effective barrier B_e is obviously not limited to be non-negative.

At a specific value of B_e , all the domains with a combination of E_c and E_e satisfying $B_e = (E_c - E_e)$ will transform together. In the special case of $E_c = 0$, $E_e = -B_e$. To derive the relationship between x_+ and B_e , we start with the simplest case of an ascending major loop. The material starts with an initial phase fraction of $x_+^0 = 0$ and $B_e^0 = -\infty$, then as the effective barrier increases to a certain value of B_e , its associated phase fraction becomes

$$\begin{aligned} x_+(B_e) &= \int_{-B_e}^{\infty} f_2(E'_e) \int_0^{E'_e+B_e} f_1(E'_c) dE'_c dE'_e \\ &= \int_{-B_e}^{\infty} \frac{1}{\sqrt{b_2\pi}} e^{-(E'_e)^2/b_2} \int_0^{E'_e+B_e} \frac{2}{\sqrt{b_1\pi} \operatorname{erfc}(-\bar{E}_c/\sqrt{b_1})} e^{-(E'_c-\bar{E}_c)^2/b_1} dE'_c dE'_e. \end{aligned} \quad (3.32)$$

The domain of integral in Equation 3.32 can be visualized as a triangular infinite area on the half infinite plane shown in Figure 3.16.

Equation 3.32 is the key to parametrizing the transformation process with respect to the

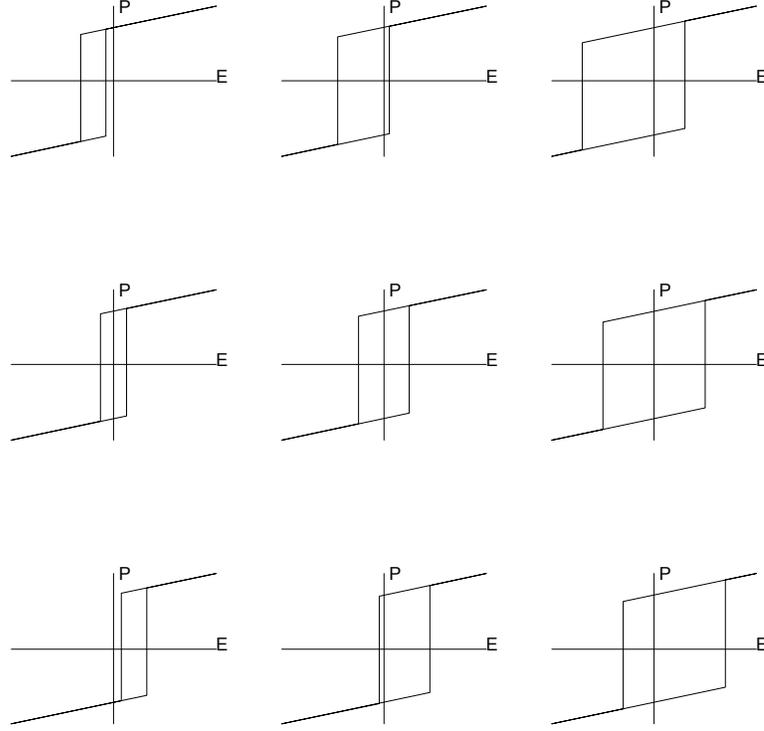


Figure 3.15: Hysteresis kernels with different combinations of coercive fields and effective fields.

effective barrier, but unfortunately we have not been able to obtain a closed-form solution for the double integral. Without a closed-form solution, Equation 3.32 can not be reversed and used in an ODE as we have done before. Moreover, this relationship becomes even more complicated for an inner loop. If the material has just reversed from a descending loop into an ascending inner loop, with a phase fraction of x_+^0 and effective barrier of B_e^0 at the reversal point, then as B_e increases, the increase in the phase fraction x_+ will be

$$\begin{aligned} \Delta x_+(B_e) = & \int_{B_e^0}^{\frac{1}{2}(B_e+B_e^0)} f(-E_e') \int_0^{E_e'+B_e} f(E_c') dE_c' dE_e' \\ & + \int_{\frac{1}{2}(B_e+B_e^0)}^{B_e} f(-E_e') \int_0^{-B_e^0-E_e'} f(E_c') dE_c' dE_e'. \end{aligned} \quad (3.33)$$

The domain of integral in Equation 3.33 can be visualized as a triangular area on the

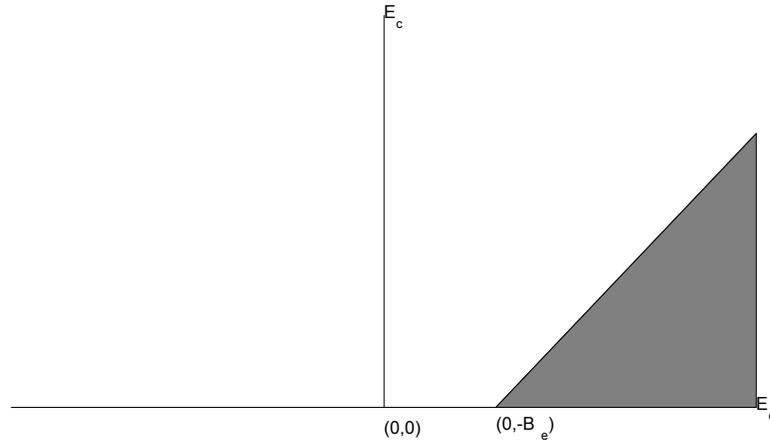


Figure 3.16: Illustration of the domain of integral in equation 3.32.

half infinite plane shown in Figure 3.17.

The relationship between the portion of transformed domains and their barriers might not be very obvious on the first glance. Figure 3.18 shows a simple minor loop along with the full major loop. To form this minor loop, the polarization starts from point (1), where $x_+ = 0$ and $B_e = -2.0 \times 10^6 \text{V/m}$. It is then loaded to zero field at point (2) and back through point (3) at $-1.0 \times 10^6 \text{V/m}$. To visualize the change of barrier with the phase fraction during the process, we show the domain states at the points (1)-(3) in Figures 3.19, 3.20 and 3.21. The domains with a positive phase are shown on the left and the domains with a negative phase are shown on the right. The domains are plotted in 3-D with their combined barrier and effective field distributions. The "volume" of the domains on the left represent the phase fraction x_+ , while the other one on the right represents the phase fraction x_- .

As we have discussed, at a specific value of B_e , all the domains with a combination of E_c and E_e satisfying $B_e = (E_c - E_e)$ will transform together. Therefore the phase transformation process from point (1) to point (2) is analogous to scanning through the distribution surface in Figure 3.20 with a vertical plane defined by $B_e = (E_c - E_e)$. The position of this scanning plane is indicated by a line at the bottom of the surface in Figure 3.20. The same

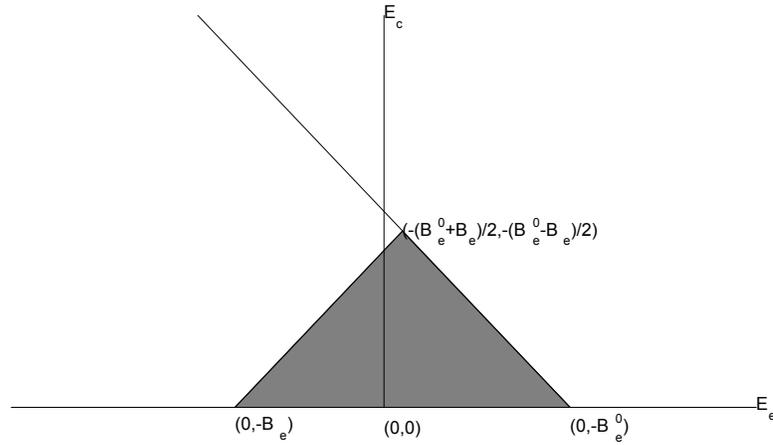


Figure 3.17: Illustration of the domain of integral in equation 3.33.

analogy also holds for the unloading process, except that the vertical scanning plane is perpendicular to the previous one and scans also in a direction perpendicular to the previous direction, because in a reverse transformation, the coercive field E_c takes a negative value in the equation $B_e = (E_c - E_e)$. This plane is also indicated in Figure 3.21.

3.3.4 An Alternative for the Combined Effective Barrier.

Given the difficulty in reversing Equation 3.32 and Equation 3.33, an alternative was used to prescribe the function $B_e(x_+)$ for both the major loop and the minor loops. We adopt the concept of "effective barriers", which differ from the original barriers only through the fact that they are allowed to be negative. Figure 3.22 shows a simulation with both distributions computed with the quadrature method. A family of ascending minor loops are also shown to compare with the major loop. We can see from the plot that the shape of the major loop still resembles the shape of an inverse complementary function, and the minor loops take very similar shapes to the major loop. Thus, for the construction of the minor loops, we use the same function as for the major outer loop and scale it such as to fit between reversal points.

Motivated by Equation 3.23, the shape function $B_e(x_+)$ is based on an inverse complementary error function. To keep the equation simple yet flexible enough, we define $B_e(x_+)$

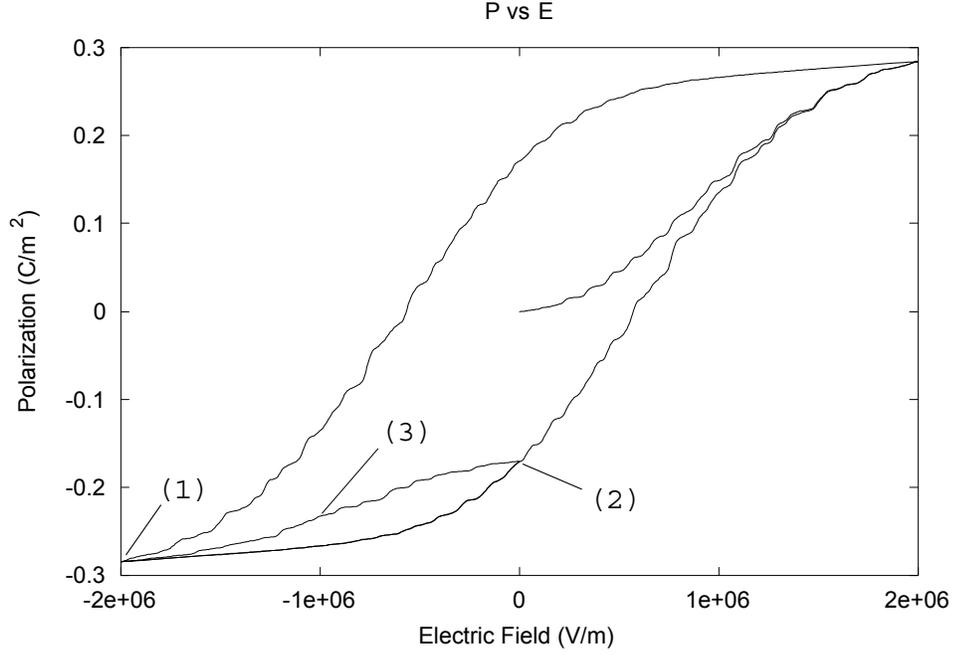


Figure 3.18: Example of a simple minor loop.

as

$$B_e(x_+) = \overline{B_e} (1 - s \cdot \operatorname{inverfc}(mx_+ + p)), \quad (3.34)$$

where $\overline{B_e}$ is the effective barrier at which the most domains transform, s is a scaling factor to adjust the shape of the loops. A smaller s will make the loops look steeper around $\overline{B_e}$. These two parameters are the key parameters of the equation and are fixed for the minor loops and the major loop. The parameters m and p determine which parts of the shape function are used to construct the minor loops and so are different for different loops. In a general case where the material reverses from a descending loop into an ascending loop, if $x_+ = x_+^1$ and $B_e = B_e^1$ at the lower reversal point and $x_+ = x_+^2$ and $B_e = B_e^2$ at the upper reversal point, m and p for this ascending minor loop can be solved from the conditions

$$\begin{cases} B_e^1 = B_e(x_+^1) = \overline{B_e} (1 - s \cdot \operatorname{inverfc}(mx_+^1 + p)) \\ B_e^2 = B_e(x_+^2) = \overline{B_e} (1 - s \cdot \operatorname{inverfc}(mx_+^2 + p)) \end{cases} \quad (3.35)$$

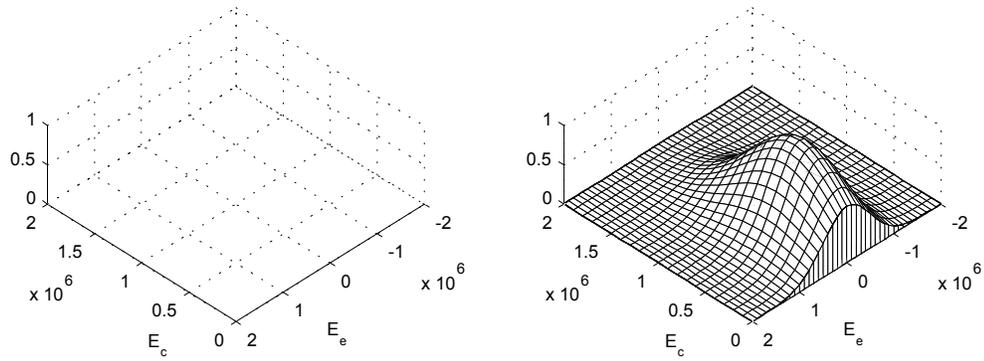


Figure 3.19: Domain states at point (1) - Left: x_+ ; Right: x_- .

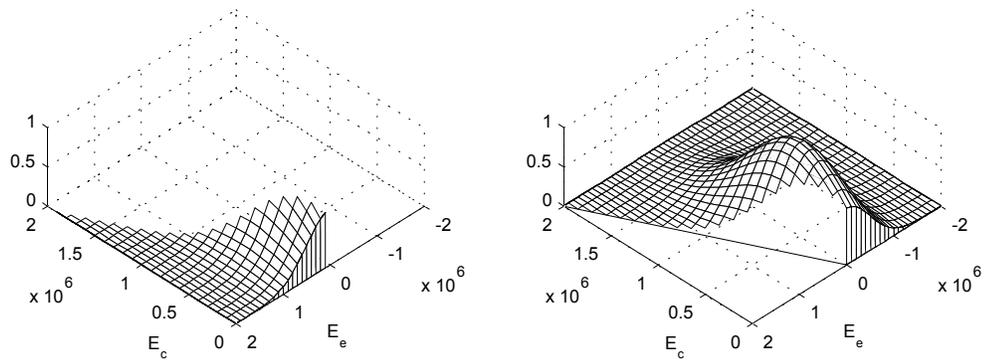


Figure 3.20: Domain states at point (2) - Left: x_+ ; Right: x_- .

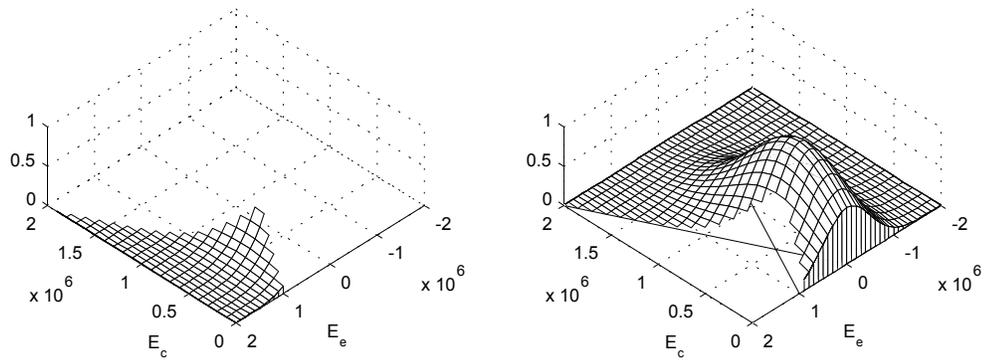


Figure 3.21: Domain states at point (3) - Left: x_+ ; Right: x_- .

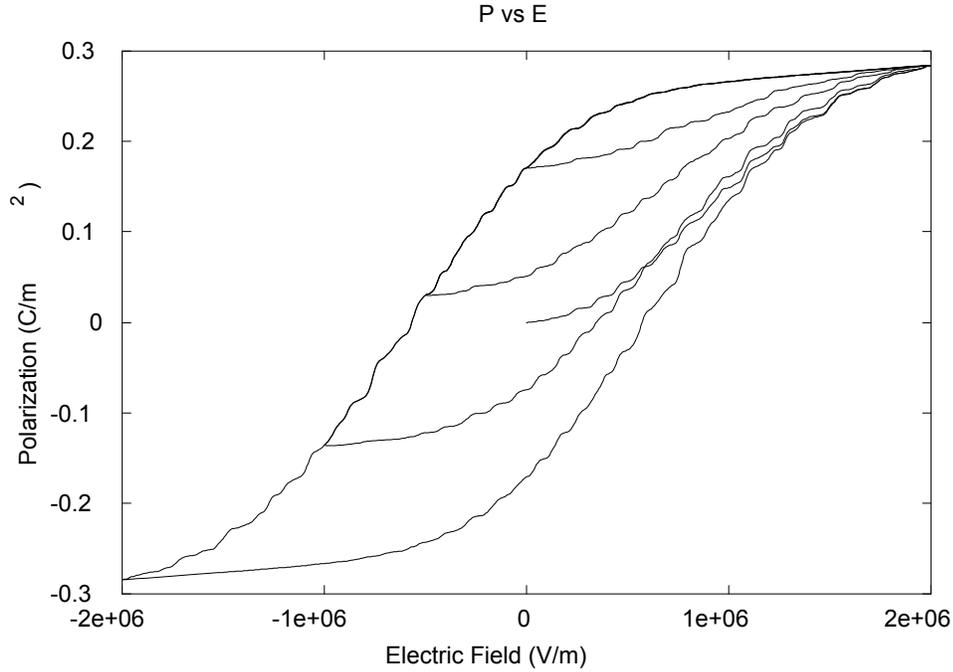


Figure 3.22: Major loop and a family of minor loops simulated with two distributions.

Equations 3.34 and 3.35 also hold for the major loop. In this case, $x_+ = 0$ at the lower reversal point and $x_+ = 1$ at the upper reversal point. Their associated effective barriers are the maximum barriers of the material. Barriers for the descending loops can be obtained by substituting $x_- = 1 - x_+$ for x_+ in Equation 3.34 and reversing its sign.

Using Equation 3.34 and the construction method above has several advantages: First, the minor loops and the major loop are calculated from the same equation and require no special treatment; Second, a minor loop will always be "inside" a major loop, e.g., for ascending loops, at the same phase fraction, the barrier for the minor loop will always be smaller than that of the major loop; Third, the return point memory is guaranteed for all the minor loops, which is a required feature of almost all hysteresis models.

By incorporating Equation 3.34 into the perfect single crystal ODE, we now have an energy model with varying barriers. In Section 3.2.3, we have shown a simulation with two distributions using the quadrature method. Here we show again the polarization vs electric behavior of that simulation in Figure 3.23(right) and try to produce the same type

of hysteresis with the parametrized version. The simulated polarization with major and minor loops using Equation 3.34 is shown in Figure 3.23(left). The parameters for the shape function are identified as $\overline{B}_e = 0.5 \times 10^6 \text{ kV/mm}$ and $s = 1.3$. The parametrized version used only 0.8 second to compute the solution, while the quadrature method needs at least $16 \times 16 = 256$ elements to resolve the fine details of minor loops and produce a reasonably smooth result, requiring 600 seconds for the whole calculation. The improvement in computational efficiency using parametrization is obviously demonstrated in this example. In a simulation from the ODE, we may not observe perfect return point memory due to the kinetics of the switching process, e.g., relaxation, but the barrier used in the model always guarantees perfect return point memory. The implication of this property will be discussed later along with more simulation results.

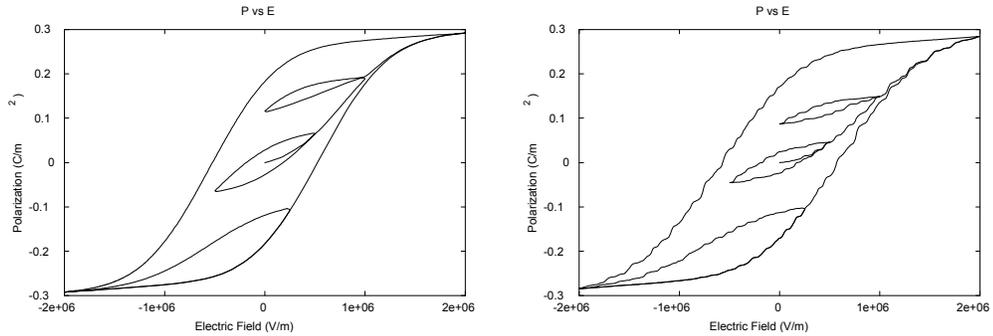


Figure 3.23: Left: Parametrized version; Right: Quadrature version.

In the Fortran implementation, we know from Equation 3.35 that the phase fraction and effective barriers at each reversal point have to be stored. Two arrays are used to store the phase fractions and effective barriers at the lower reversal points, and another two arrays for the upper reversal points. As a matter of fact, reversal points do not need to be distinguished as lower and upper in the code, but doing so makes the implementation cleaner and easier to understand. Detailed Fortran code is included in the appendices of the thesis.

3.3.5 Model Fit to Experiments

As we have demonstrated that the parametrized model can produce almost the same characteristics of the quadrature method, we would now fit the model with the experimental data presented in chapter 2. The simulations in this section are all computed under linear loads. Figure 3.24 shows the model fit to the experimental data from the low rate linear loading full hysteresis experiment. The identified parameters are: $s = 0.9$, $\overline{B}_e = 0.5\text{kV/mm}$, $T = 293\text{K}$, $V_D = 1.0 \times 10^{-22}\text{m}^3$, $\tau_x = 5.0 \times 10^{-2}\text{s}^{-1}$, $P_T = 0.215\text{C/m}^2$, $E_1 = 2.8 \times 10^7\text{Vm/C}$. The strain is calculated from the polarization through the approximation $S = \zeta P^2[1]$, where ζ is a constant. This approximation is motivated by electrostrictive or perfect single crystal behavior, and here $\zeta = 2.293\text{m}^4/\text{C}^2$. In Figure 3.24, although the model qualitatively captures the characteristics of the hysteresis, it does not make a perfect match to the experiment, especially in the $S - E$ hysteresis. This discrepancy is expected due to two reasons: First, the shape function (the inverse complementary error function) used for the parametrized method was originally chosen to reproduce the similar curves from the quadrature method, while real-world materials might have a distribution of barriers that take a very different shape from the analytical function. Second, the strain-polarization hysteresis we observed from the experiments in chapter 2 indicates that the approximation $S = \zeta P^2$ needs further development to capture the relationship between polarization and strain.

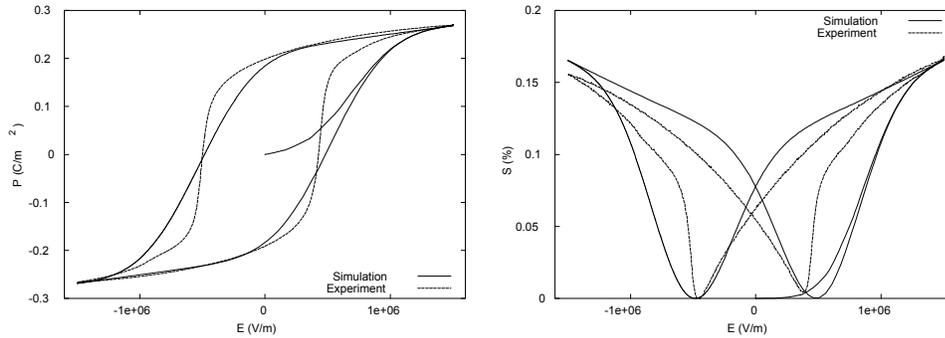


Figure 3.24: Left: Polarization; Right: Strain.

Besides the full hysteresis loop, the model has also been fitted to the actuator loops. Given the known limits of the model, we now focus more on the fitting of the strain curves. Much less hysteresis is present between strain and polarization in this range. The identified

parameters are: $T = 293\text{K}$, $V_D = 1.0 \times 10^{-22}\text{m}^3$, $\tau_x = 8.0 \times 10^{-2}\text{s}^{-1}$, $P_T = 0.210\text{C/m}^2$, $E_1 = 2.5 \times 10^7\text{Vm/C}$. In addition, we used different shape function parameters for the ascending loop and the descending loop to obtain a better fit in the actuator range. For the ascending loop, $s = 2.5$, $\overline{B}_e = 0.25\text{kV/mm}$. For the descending loop, $s = 1.1$, $\overline{B}_e = 1.0\text{kV/mm}$. The simulated $S - E$ curves for all four types of minor loops are shown in Figures 3.25, 3.26, 3.27 and 3.28, along with their corresponding low rate experimental curves. In this range, the simulated curves match the experimental results much better than in the full hysteresis range. The trends of the different minor loops are reflected in the simulations, though the simulated small minor loops are narrower than what we observe in the experimental curves, especially for the descending loops. From the curves for both the full loops and the minor loops, we can see that a better fit may be possible by choosing a better shape function and improving the algorithm for calculating minor loops.

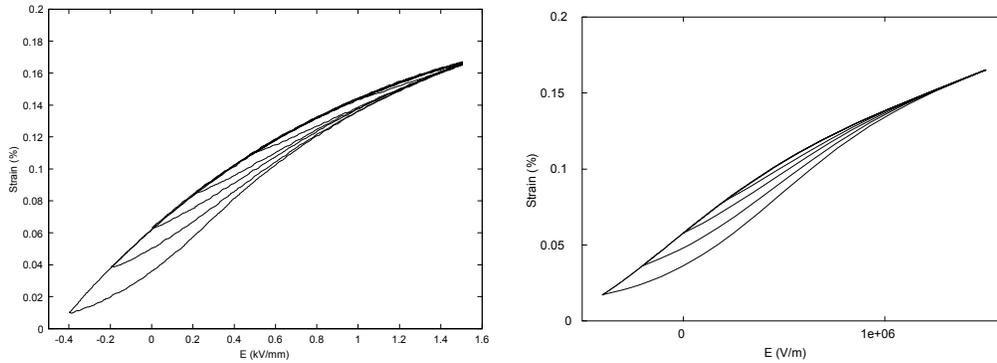


Figure 3.25: Type 1: Left: Experiment; Right: Simulation.

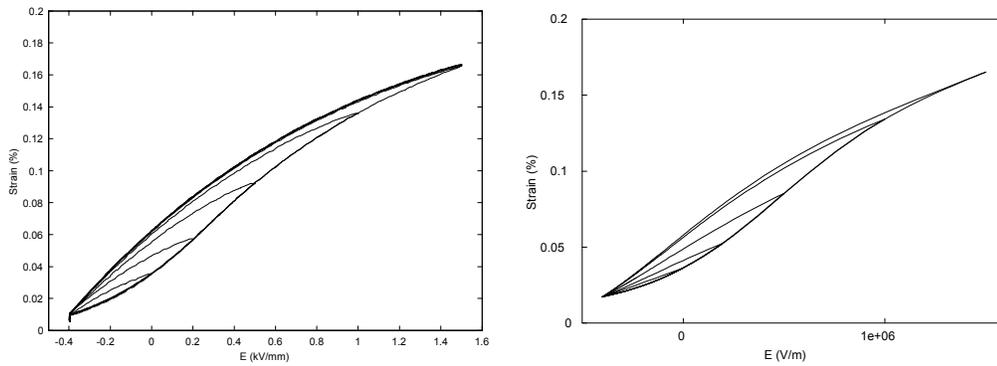


Figure 3.26: Type 2: Left: Experiment; Right: Simulation.

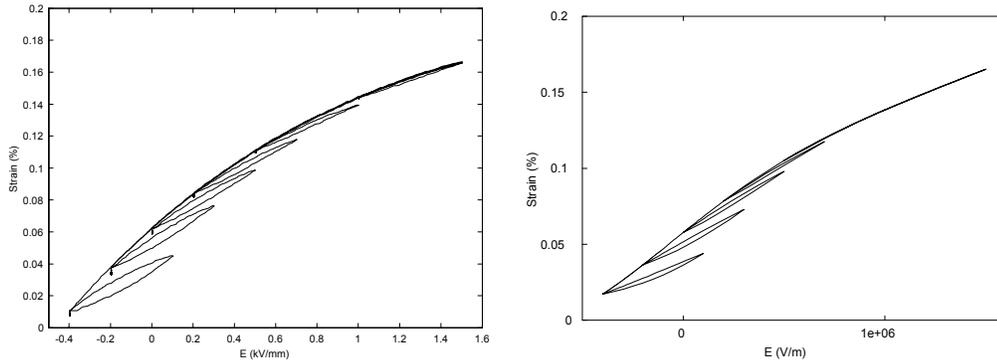


Figure 3.27: Type 3: Left: Experiment; Right: Simulation.

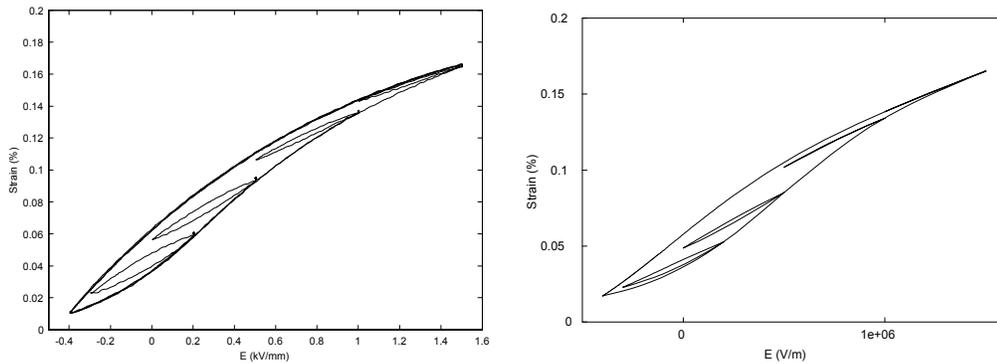


Figure 3.28: Type 4: Left: Experiment; Right: Simulation.

Another important property of the PZT model is its rate dependence and the resulting relaxation behavior. In Figure 3.29, we show again the relaxation plot we have seen at the end of chapter 2. During the 10-second constant field period at the end, the strain relaxes slowly and asymptotically towards a final equilibrium. The two curves start from different strain levels, but they relax toward the same level after the relaxation period. Simulations with the parametrized method under the same loads are plotted in Figure 3.30. The simulations qualitatively show the rate dependence and relaxation behavior in the $S - E$ plot, but in the strain vs. time plot, we see that the model doesn't produce similar relaxation on the same time scale. The drop of the strain takes place almost immediately at the beginning of the 10-second relaxation period. This behavior looks like the one observed in a perfect crystal with just one barrier.

For comparison, another set of simulations using the quadrature method is shown in

Figure 3.31. Due to the discretized nature of the quadrature method, we are unable to obtain a close fit for the simulation within a reasonable number of abscissae points, so the numerical values of the simulation are not directly comparable to the experiments. But on the strain vs. time plot, we observe the same type of gradual relaxation as the experiments. The higher-rate curve starts from a higher strain level, and the two curves both relax slowly and asymptotically toward a final equilibrium.

In the model, the parameters V_D (domain volume) and τ_x (relaxation time) decide the extent and rate of relaxation. For the simulations in Figure 3.30 and Figure 3.31, the same set of V_D and τ_x are used. While the quadrature method retains the correct relaxation behavior, the parametrized method does not. Further investigation is needed to find out the reason for this behavior and improve the parametrized model. This shall be addressed in future research.

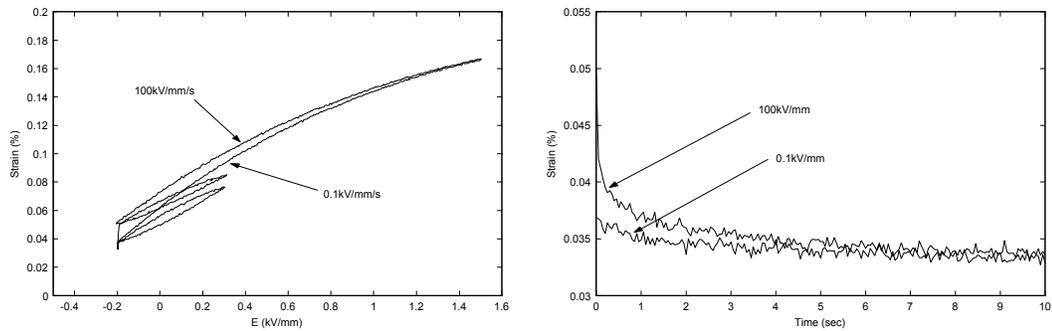


Figure 3.29: Relaxation behavior (Experimental data) - Left: S-E Hysteresis; Right: Strain relaxation over time.

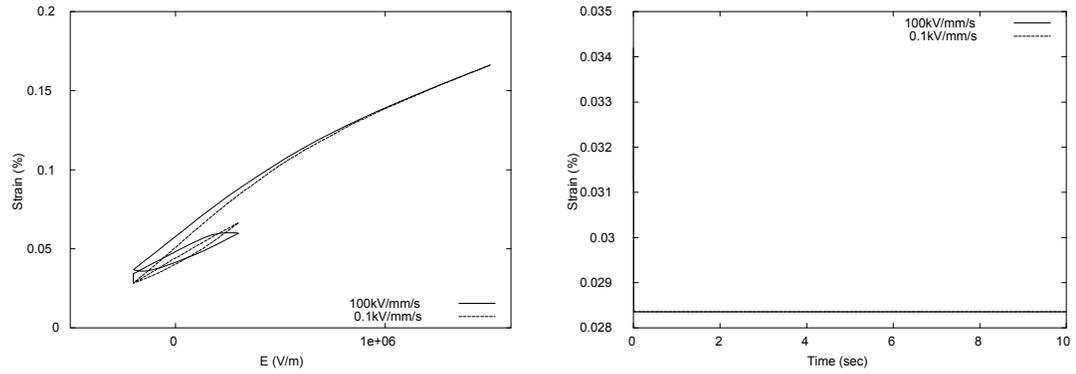


Figure 3.30: Relaxation behavior (Parametrized method) - Left: S-E Hysteresis; Right: Strain relaxation over time.

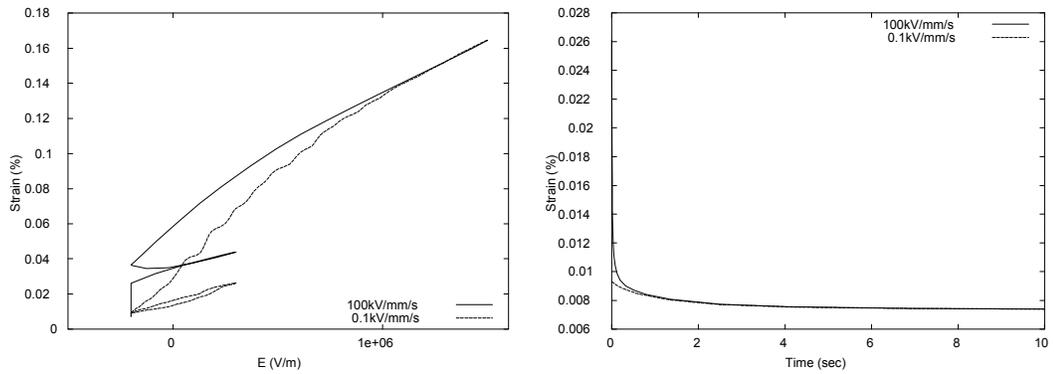


Figure 3.31: Relaxation behavior (Quadrature method) - Left: S-E Hysteresis; Right: Strain relaxation over time.

Chapter 4

Optimal Control of PZT

This chapter presents the first results of an optimal control approach to piezoceramic actuators. A preliminary version of the model introduced in the previous chapter is implemented into an optimal control package. The implementation does not address real-time aspects yet, but as a first step, focuses on basic off-line control problems. Two non-trivial control problems are simulated to illustrate the features and potential of the method.

4.1 Optimal Control Problem

Optimal control is an attractive approach in modern control applications, as it aims to minimize some cost functional like energy consumption or final time. It is especially powerful when the accurate model of the physical process is available [14]. An optimal control problem is generally stated as follows [39]: Given

1. The state equations, for a continuous system usually a set of differential equations $\dot{x}(t) = f(x(t), u(t))$, where $x(t)$ is the system state vector and $u(t)$ is the control input vector;
2. A set of boundary conditions on the state variables at the initial and terminal time;
3. A set of constraints on the state variables and control variables,

determine the control input $u(t)$ so that a cost functional is minimized.

A major problem for such an optimal control problem is the huge computational cost due to excessive memory requirement for discretized states and the related matrix operations during the solution process. Depending on the complexity of the state equations and the choice of dynamic programming method, the time required to find the solution could be prohibitive to any real-world application. To make the optimal control problem viable, the physical model must first be simple and efficient enough, and the optimization method should be carefully chosen. As we have demonstrated in the previous chapter, the parametrized implementation of the PZT model is very computationally efficient and could potentially serve as the basis for calculating the optimal control.

In the following sections, we will study the solution of optimal control problems with our model. For simplicity, we consider a one-dimensional piezoceramic stack actuator in the absence of mechanical loads, and, instead of tracking the displacement, we consider the polarization as the quantity that can be prescribed.

Thus we prescribe the polarization set-point $P_{set}(t)$, and solve for the electric field function $E(t)$, which minimizes the following cost functional

$$J = \int_0^T [P(t) - P_{set}(t)]^2 dt. \quad (4.1)$$

This ensures that the set point is reached in the shortest time possible and subsequently maintained. The minimization is subject to the following constraints

$$\dot{x} = f(x(t), E(t)), \quad (4.2a)$$

$$x(0) = x_0, \text{ and} \quad (4.2b)$$

$$|E(t)| \leq E_{\max}. \quad (4.2c)$$

Equation 4.2a represents the model equation, which describes the switching processes in the actuator. Equation 4.2b are the initial conditions of the process. And Equation 4.2c are box constraints for the control, representing the fact that only a finite electric field can be applied to the actuator.

For the numerical solution of the problem, we implement the model into the optimal control code NUDOCSS developed by Büskens [2]. Based on a direct approach, NUDOCSS discretizes the control function and transforms the original control problem into a nonlinear optimization problem (NLP). The resulting NLP problem is then solved with a

sequential quadratic programming (SQP) method. In the solution of NUDOCCCS, the control function (in this case the electric field $E(t)$) can be either assumed piecewise linear or interpolated by higher order splines. We will use the piecewise linear option for simplicity and faster calculation. The differential equations for the state variables are integrated by an implicit Runge-Kutta scheme (RADAU IIa) using the efficient and robust RADAU5 routine from the book of Hairer and Wanner [15].

4.2 Implementation of the Model Used for Optimal Control

Although the final model developed in chapter 3 is capable of simulating complex major and minor loops within a much shorter time, more development is needed to further improve its stability for use in optimal control simulations. For the optimal control in this chapter, we will use a simplified implementation of the parametrized model. Only the distribution in coercive field E_c is implemented and the history tracking for the minor loops has been disabled to allow maximum computational speed and stability. Even though it is not a full implementation of the model in chapter 3, it is still capable of predicting the outer hysteresis loop and illustrate the rate dependence for inner loops. Figure 4.1 shows the model's capability of capturing the frequency dependence for two typical values. Note the dramatic impact on the shape of outer and inner hysteresis loops. The parameter values used for the simulations throughout this chapter are: $T = 273\text{K}$, $V_D = 1.0 \times 10^{-22}\text{m}^3$, $\tau_x = 1.0 \times 10^{-2}\text{s}^{-1}$, $P_T = 0.26\text{C}/\text{m}^2$, $E_1 = 6.67 \times 10^7\text{V}\cdot\text{m}/\text{C}$, $b_1 = 2.0 \times 10^{22}$, and $\overline{E}_c = 1.0 \times 10^6\text{V}/\text{m}$.

4.3 Optimal Control Simulation

4.3.1 Case 1: Single-level Set-point Function

We start with a simple case, which allows illustrating the rate-dependent behavior of the actuator. From an initial polarization $P(0) = 0\text{C}/\text{m}^2$, the set point rises linearly to a new level $P_{set} = 0.2\text{C}/\text{m}^2$ at a rate of $500\text{C}/\text{m}^2/\text{s}$ within 4 ms. The constraint E_{\max} has been set to 6 MV/m.

The four plots in Figure 4.2 show the time evolutions of the phase fraction and polarization along with the value of the cost functional and the optimal electric field (from upper

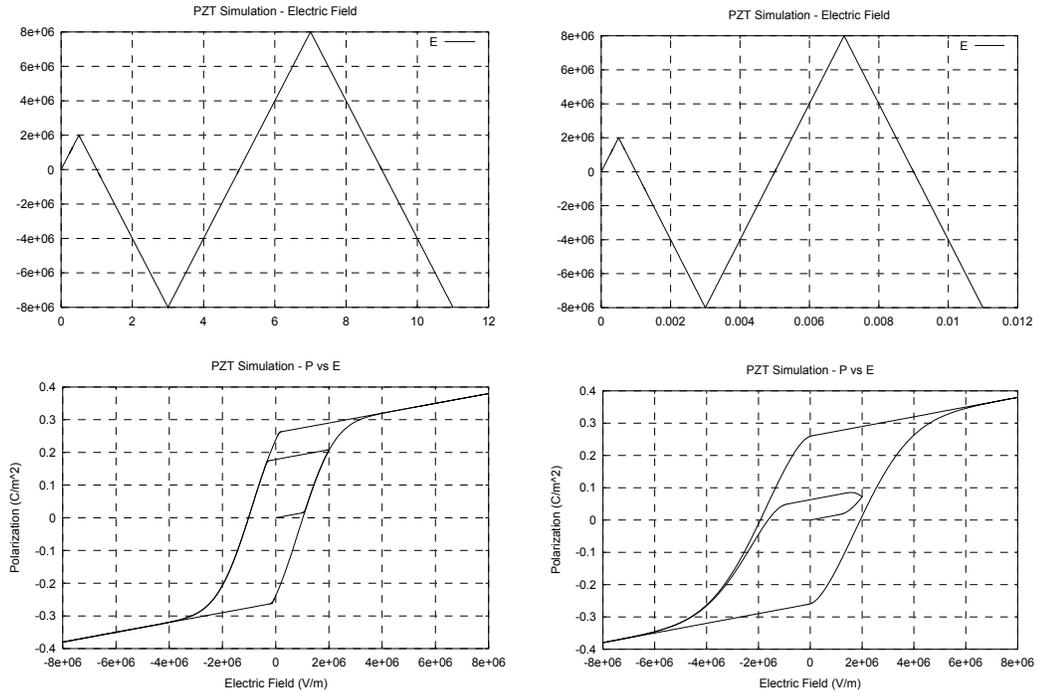


Figure 4.1: Left: hysteresis curve at 0.125Hz; Right: hysteresis curve at 125Hz.

left to lower right). Figure 4.3 shows the hysteretic behavior of the actuator in a P-E plot.

The polarization calculated from the optimal solution follows the set-point function closely, especially in the middle of the rising process, which can be seen from the cost functional increasing only very little during this phase of the process. Most of the error occurs near the beginning and the end of the transition. In the mean time, the phase fraction increases monotonously from 0.5 to 0.83. The electric field, however, exhibits some interesting features due to the nonlinear and hysteretic behavior of piezoceramics at different loading rates.

When a positive electric field is applied to the piezoceramic actuator, the increase in polarization contains two different terms. The first term is proportional to the applied field and time-independent. The second term is governed by the evolution of the phase fraction and as such is time-dependent. The model automatically accounts for this behavior through a suitable choice of relaxation time and activation volume, and consequently, the piezoceramic actuator exhibits different shapes of hysteresis at different loading frequencies.

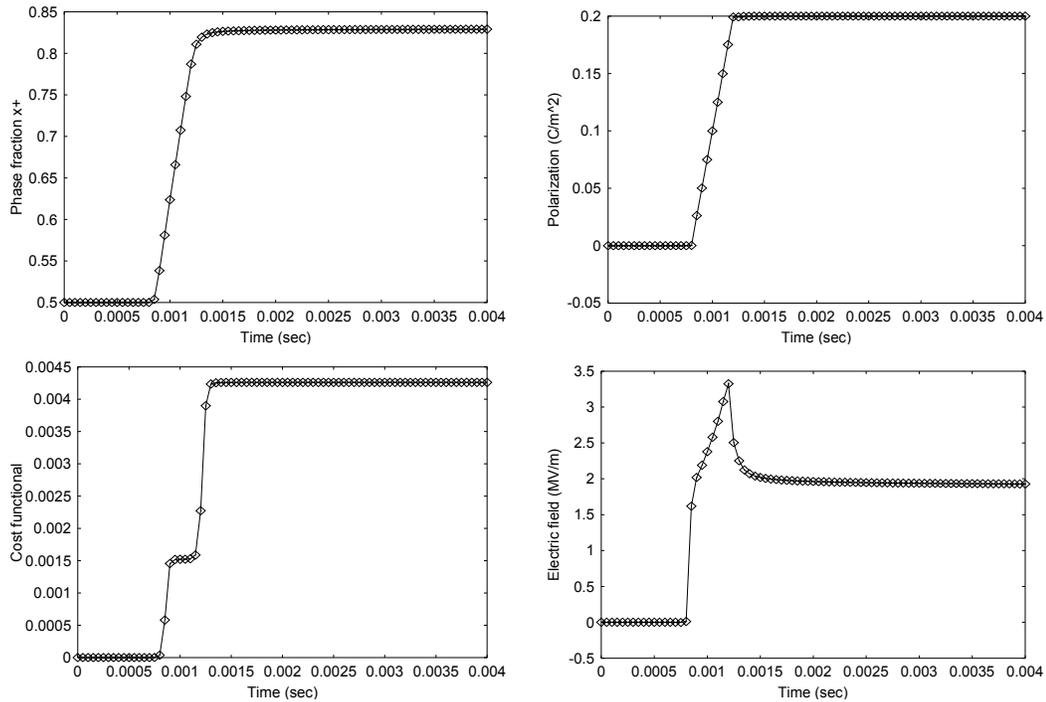


Figure 4.2: Optimal control of piezoceramic polarization (Case 1).

As we can see from Figure 4.2, the electric field starts with a sharp increase, which is equivalent to a loading frequency of 1250 Hz, then continues to increase at a roughly 10 times slower rate, until it reaches the peak of 3.3 MV/m. At this electric field value, the polarization has almost reached the required set point, but the phase transition has not yet completed. Some domains of negative phase continue to change into positive phase, which will increase the overall polarization of the actuator. To compensate for this increase, the electric field then decreases asymptotically to about 1.9 MV/m. This process is better illustrated in Figure 4.3, which shows the optimal solution along with hysteresis loops at three different loading frequencies.

In Figure 4.3, the actuator first follows the hysteresis curve at the highest frequency (the widest loop), then as the rate of the electric field decreases, it overlaps with the curve at a lower loading frequency, and finally relaxes to a point on the quasistatic outer loop (0.125 Hz). At a frequency of 0.125 Hz, the effect from the relaxation time of the domains is negligible because the domains now have enough time to complete the phase transition

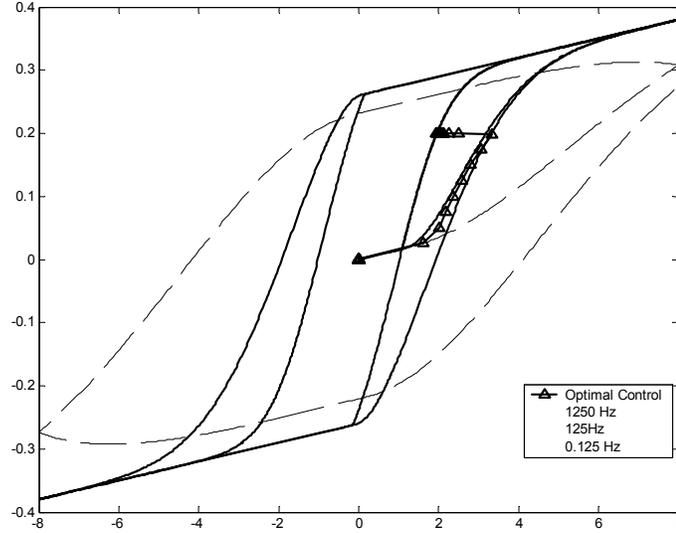


Figure 4.3: P-E behavior of the optimal solution (Case 1).

as the field increases. If no optimal control is applied, the same polarization set point can be achieved by slowly increasing the electric field. The actuator would then follow the hysteresis curve of 0.125 Hz.

4.3.2 Case 2: A More Complex Set-point Function

Due to hysteresis, the same polarization may require a different electric field if the loading history is different. We proceed with the prescription of a more complex set-point function to verify the model's ability to predict the polarity of the electric field needed for a positive polarization with different history. The set-point polarization first increases from 0.0 C/m^2 to 0.1 C/m^2 linearly at a rate of $500 \text{ C/m}^2/\text{s}$, holds for a period of time, increases again to 0.2 C/m^2 at the same rate, holds the value, and then decreases back to 0.1 C/m^2 at a rate of $-500 \text{ C/m}^2/\text{s}$. In the simulation, the control process has a total duration of 4 ms, and is discretized using 81 points (NDISKRET=81 in NUDOCCS). E_{\max} is set to 6 MV/m. The four plots in Figure 4.4 show the time evolutions of the phase fraction and polarization

along with the value of the cost functional and the necessary electric field (from upper left to lower right). Figure 4.5 shows the hysteretic behavior of the actuator in a P-E plot.

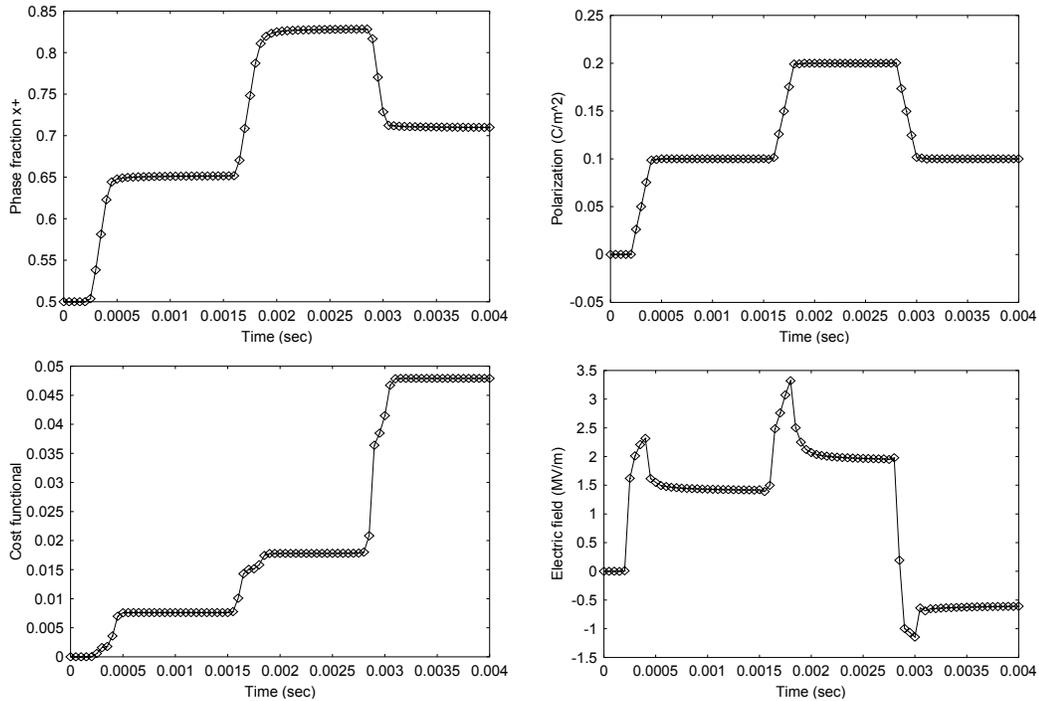


Figure 4.4: Optimal control of piezoceramic polarization (Case 2).

In this case, not surprisingly, the model predicts similar shapes of optimal electric fields for the actuator. The electric field increases first to a high value, and then drops back to a lower value as the piezoceramic material relaxes, forcing the polarization to follow the set point. The maximum electric field needed for the second set point of 0.2 C/m^2 is almost the same as that of the previous section, and the field also decreases at the same rate to the same final value to maintain the polarization. As Figure 4.5 shows, in following the three set points, the actuator first follows the hysteresis loop of a higher frequency but always comes back to a point on a loop of low frequency.

For the first and third set point, although they have the same polarization of 0.1 C/m^2 , the model predicts completely different phase fractions and electric fields. Obviously, this is due to the difference in the history. For the first set point, a positive electric field is applied, because the initial positive phase fraction needs to be increased. To reach the third

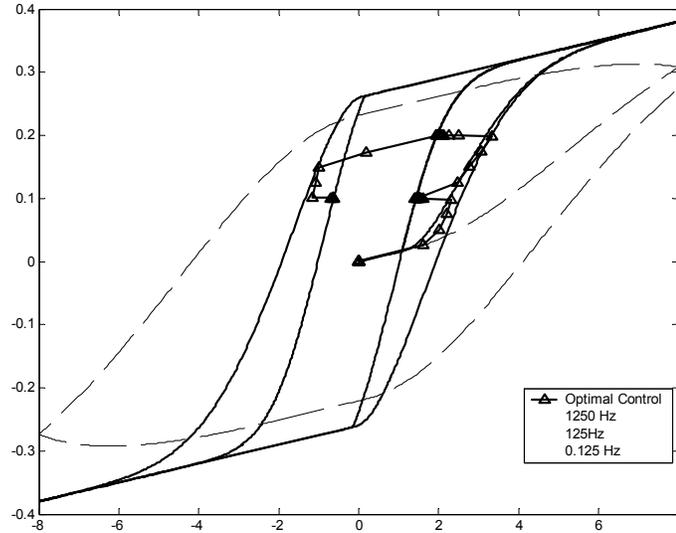


Figure 4.5: P-E behavior of the optimal solution (Case 2).

set point, the hysteresis has to be traversed, and a negative electric field has to be applied in order to reach the same polarization value as for the first set point.

4.3.3 Conclusion

This chapter introduced the implementation of the parametrized energy-based model for the simulation of polycrystalline piezoceramic materials. The model reproduces the outer hysteresis loops typically observed in these materials correctly, although a better simulation of minor loops will require further development. It was also shown, that the effect of varying loading rate is captured, and both these effects were illustrated to play an important role in typical control problems. The model has been implemented into an optimal control code, which not only compensates for the non-linear and hysteretic material behavior, but allows optimality criteria like speed of adjustment to be taken into account. This first implementation demonstrated the feasibility of the concept by successfully computing offline solutions. Future work shall focus on: 1. Implementing the complete parametrized model

into the optimal control code for more versatile and accurate simulation; 2. Implementing real-time control concepts with sensitivity analysis developed by the authors of [29] and [33].

Chapter 5

Conclusions

This thesis has presented work in three aspects: experimental investigation on the hysteretic behavior of a typical piezoceramic actuator, modeling of hysteresis, and feasibility studies of the model in an optimal control problem.

First, four types of representative actuation experiments on a typical commercially available piezoceramic stack actuator are conducted and analyzed, with a focus on the rate-dependency of the hysteretic behavior over a long travel range under different loading rates. Besides confirming material behavior documented in previous literature, the set of collected data covers the driving range usually unavailable in previous research of the area. In the constant field relaxation experiments, we have found the strain and polarization of the material to relax asymptotically toward a fixed equilibrium level under the same constant electric field.

Secondly, an energy model for polycrystalline piezoceramic materials proposed by Smith and Seelecke has been implemented with two different methods, namely the quadrature method and the parametrization method. In the past, the model has only been partially implemented in Matlab by Smith, whose implementation achieves efficiency by omitting domain switching kinetics and utilizing algebraic matrix operations. Therefore it was valid only for the limiting case of zero thermal activation and quasistatic loading rate. The quadrature method in this thesis is a full implementation of the model, computing the singular crystal energy model ODE for each quadrature element to reflect the kinetics of domain switching. It is useful in verifying the validity of the model under all loading conditions. Because of the huge computational cost associated with the quadrature method,

it is not suited for control applications, so the parametrization method has been proposed to implement the distributions in the model without using quadratures. The parameterization method reduces the implementation to only one ODE and drastically improves the computational efficiency while retaining the energy basis of the model, typically 3 orders of magnitudes faster than the quadrature method when comparable results are simulated. This makes the model useful in control applications and is one of the major contributions of the thesis. The model has been found to qualitatively match most of the material behavior measured in the experiments.

And finally, with the model established, feasibility studies have been done on applying the energy model for the piezoceramic material in an optimal control problem. To the best of our knowledge this has been done for the first time, and the results demonstrate that the model has a good potential for use in optimal control applications. The program is able to minimize tracking error by predicting the relaxation of the material and the different polarities of electric field for the same set-point due to hysteresis. This is where traditional PID control schemes usually fail because of these undesirable material properties.

Future work of this research naturally also contains three aspects. First of all, additional experiments focusing on the relaxation behavior may provide more insight into the physical principles behind the phenomenon. Secondly, the parameterization method needs to be further developed. Despite the advantage of the parameterization method, it is not yet able to produce the same results as the quadrature method under all conditions, especially for the slow relaxation behavior. With a more thorough understanding of the physical background, improvement shall be made to the parameterized energy model to enable the correct prediction of slow relaxation over time. In computing the strain of the material, the hysteretic strain-polarization relationship must be accounted for, and furthermore, the distributions must be identified more exactly in order to provide better quantitative agreement with the experiments. Finally for optimal control, a more complete implementation of the model into the optimal control code is needed to achieve more flexible and accurate simulations. This will provide the basis for a final real-time implementation as has been developed for shape memory alloys.

List of References

- [1] a. J. Moulson and M. Herbert J. *Electroceramics*. Chapman and Hall, 1990.
- [2] C. Büskens. *NUDOCCCS - User's Manual (In German)*. Universität Münster, 1996.
- [3] W. Chen and C. S. Lynch. A model for simulating polarization switching and af-f phase changes in ferroelectric ceramics. *Journal of Intelligent Material Systems and Structures*, 9:427–431, 1998.
- [4] R. L. Clark, W. R. Saunders, and G. P. Gibbs. *Adaptive Structures: Dynamics and Control*. John Wiley & Sons, 1998.
- [5] B. Clephas. *Adaptronics and Smart Structures*, chapter Piezoelectric Actuators. Springer Verlag, 1999.
- [6] J. M. Cruz-Hernandez and V. Hayward. An approach to reduction of hysteresis in smart materials. In *Proceedings of the 1998 IEEE International Conference on Robotics & Automation*, pages 1510–1515, Leuven, Belgium, 1998.
- [7] A. Daniele, S. Salapaka, M. V. Salapaka, and M. Dahleh. Piezoelectric scanners for atomic force microscopes: Design of lateral sensors, identification and control. In *Proceedings of the American Control Conference*, pages 253–257, San Diego, California, June 1999. AACC.
- [8] L. A. L. de Almeida, G. S. Deep, A. M. N. Lima, and H. Neff. Limiting loop proximity hysteresis model. *IEEE Transactions on Magnetics*, 39(1), January 2003.
- [9] Klaus Dragosits. *Modeling and Simulation of Ferroelectric Devices*. PhD thesis, Technischen Universität Wien, 2000.

- [10] W. S. Galinaitis and R. C. Rogers. Compensation for hysteresis using bivariate Preisach models. In *Proc. 4th Ann. Int. Symp. Smart Struct. Mat., Mathematics and Control in Smart Structures*, San Diego, CA, U.S.A., 1997.
- [11] P. Ge and M. Jouaneh. Modeling hysteresis in piezoceramic actuators. *Precision Engineering*, 17:211–221, 1995.
- [12] P. Ge and M. Jouaneh. Tracking control of a piezoceramic actuator. *IEEE Transactions on Control System Technology*, 4:209–216, 1996.
- [13] D. Grant and V. Hayward. Variable structure control of shape memory alloy actuators. *IEEE Contr Syst Mag*, 17(3):80–88, 1997.
- [14] M. Grötschel, S. O. Krumke, and J. Rambau (Eds.). *Online Optimization of Large Scale Systems*. Springer, 2001.
- [15] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II - Stiff and Differential-Algebraic Problems*. Springer Verlag, 1991.
- [16] L. Huang and H. F. Tiersten. An analytic description of slow hysteresis in polarized ferroelectric ceramic actuators. *Journal of Intelligent Material Systems and Structures*, 9:417–426, 1998.
- [17] Physik Instrumente. <http://www.physikinstrumente.com/>.
- [18] P. Ioannou and J. Sun. *Robust Adaptive Control*. Prentice Hall, Englewood Cliffs, NJ, 1996.
- [19] H. Janocha. *Adaptronics and Smart Structures*, chapter Actuators in Adaptronics. Springer Verlag, 1999.
- [20] S. Jung and S. Kim. Improvement of scanning accuracy of pzt piezoelectric actuators by feedforward model-reference control. *Precision Engineering*, 16:49–55, 1994.
- [21] M. Kamlah. Ferroelectric and ferroelastic piezoceramics - modeling of electromechanical hysteresis phenomena. *Continuum Mechanics and Thermodynamics*, 13:219–268, August 2001.

- [22] S. M. Kuo and D. R. Morgan. *Active Noise Control Systems*. John Wiley & Sons, 1996.
- [23] F. Liorzou, B. Phelps, and D. L. Atherton. Macroscopic models of magnetization. *IEEE Transactions on Magnetics*, 36(2), March 2000.
- [24] The Numerical Algorithms Group Ltd, Oxford, UK. *The NAG Fortran Library Manual, Mark 20*, 2002.
- [25] N. Papenfuß and S. Seelecke. Simulation and control of sma actuators. In *Proc. 6th Ann. Int. Symp. Smart Strut. Mat. vol. 3667*, pages 586–595, Newport Beach, CA, U.S.A., March 1999. SPIE.
- [26] F. Preisach. Über die magnetische nachwirkung (on the magnetic aftereffect). *Zeitschrift für Physik*, 94:277–302, 1935.
- [27] Queensgate. <http://www.nanopositioning.com/>.
- [28] S. Ruckmich. *Elektrokeramik lecture notestu berlin*, 1995.
- [29] S. Seelecke. *Adaptive Structures with SMA Actuators - Modeling and Simulation (In German)*. TU Berlin, 1999.
- [30] S. Seelecke and C. Büskens. optimal control of beam structures by shape memory wires. In Hernandez and C. A. Brebbia, editors, *OPTI 97, Computer Aided Optimum Design of Structures*, pages 457–466, Rome, Italy, September 1997. Comp. Mech. Press.
- [31] S. Seelecke, C. Büskens, I. Müller, and J. Stryk. *Online Optimization of Large Systems: State of the Art*, chapter Real-Time Optimal Control of Shape Memory Alloy Actuators in Smart Structures. Springer Verlag, 2001.
- [32] S. Seelecke and I. Müller. Shape memory alloy actuators in smart structures - modeling and simulation. To appear in *Appl. Mech. Reviews*, 2003.
- [33] R. C. Smith. A nonlinear optimal control method for magnetostrictive actuators. *Journal of Intelligent Material Systems and Structures*, 9:468–486, 1998.

- [34] R. C. Smith. Inverse compensation for hysteresis in magnetostrictive transducers. *Mathematical and Computer Modelling*, 33:285–298, 2001.
- [35] R. C. Smith and S. Seelecke. An energy formulation for preisach models. In *Proc. 9th Ann. Int. Symp. Smart Struct. Mat., Modeling, Signal Processing and Control*, San Diego, CA, U.S.A., March 2002. SPIE.
- [36] R. C. Smith, S. Seelecke, M. J. Dapino, and Z. Ounaies. Unified model for hysteresis in ferroic materials. In *Proc. 10th Ann. Int. Symp. Smart Struct. Mat., Modeling, Signal Processing and Control*, San Diego, CA, U.S.A., March 2003. SPIE.
- [37] R. C. Smith, S. Seelecke, and Z. Ounaies. A free energy model for piezoceramic materials. In *Proc. 9th Ann. Int. Symp. Smart Struct. Mat., Modeling, Signal Processing and Control*, San Diego, CA, U.S.A., March 2002. SPIE.
- [38] R. C. Smith, S. Seelecke, Z. Ounaies, and J. Smith. A free energy model for hysteresis in ferroelectric materials. Submitted to *Journal of Intelligent Material Systems and Structures*, 2003.
- [39] D. Tabak and B. C. Kuo. *Optimal Control by Mathematical Programming*. Prentice Hall, 1971.
- [40] S. Tan and H. Xu. private communication tu berlin, 1995.
- [41] X. Tan. *Control of Smart Actuators*. PhD thesis, University of Maryland, College Park, MD, 2002.
- [42] X. Tan and J. Baras. Optimal control of hysteresis in smart structures: A viscosity solutions approach. In *Proceedings of the 5th International Workshop on Hybrid Systems: Computation and Control*, pages 451–464. Springer series LNCS 2289, 2002.
- [43] G. Tao and P. V. Kokotovich. *Adaptive Control of Systems with Actuator and Sensor Nonlinearities*. John Wiley & Sons, 1996.
- [44] G. V. Webb and D. C. Lagoudas. Control of sma actuators under dynamic environments. In *Proc. 6th Ann. Int. Symp. Smart Struct. Mat.*, page vol. 3667, Newport Beach, CA, U.S.A., March 1999. SPIE.

- [45] M.-A. Weber, M. Kamlah, and D. Munz. Experiments about time effects in piezoceramics(in german). Technical Report FZKA 6465, Forschungszentrum Karlsruhe GmbH, Karlsruhe, Germany, 2000.
- [46] B. Widrow and S.D. Stearns. *Adaptive Signal Processing*. Prentice Hall, Englewood Cliffs, NJ, 1985.
- [47] D. Zhou. *Experimental Investigation of Non-linear Constitutive Behavior of PZT Piezoceramics*. PhD thesis, Forschungszentrum Karlsruhe GmbH, Karlsruhe, Germany, 2003.
- [48] D. Zhou, M. Kamlah, and D. Munz. Rate dependence of soft pzt ceramics under electric field loading. In Christopher S. Lynch, editor, *Proc. 8th Ann. Int. Symp. Smart Struct. Mat., Active Materials: Behavior and Mechanics*, pages 64–70, Newport Beach, USA, March 2001. SPIE, SPIE.
- [49] S. E. Zirka and Y. I. Moroz. Hysteresis modeling based on similarity. *IEEE Transactions on Magnetics*, 35(4), July 1999.

Appendix A

Fortran Programs

A.1 Quadrature Method

Main program:

```
c PZT Simulation Program
c
c Language: F77
c
c This program comes with two include files:
c 'quad.for' passes gaussian quadrature parameters
c 'load.for' passes linear loading information
c
c Revision: 20030714
c
c 20030714
c
c The program now outputs additional plot data of the two distributions
c in a better way.
c
c 20021028
c
c Restructured include files
c to keep only pzt related functions in the main file
c
c 20021022
c
c Improvement:
c Divided the integration range into two intervals to place more itegration
c points at the peak of the probability curve.
c
c 20021021
```

```
c
c A regular backup version.
c
c 20021016
c
c Change:
c Program now outputs plot data for the distribution curve
c
c 20021014
c
c Optimization:
c Combined some of the routines and took out some constants for efficiency.
c
c Improvement:
c CALERF from SPECFUN (Author W.J.Cody) is now used for the calculation of
c probabilities. More specifically, DERFCX is used in probabilities().
c
c 20021006
c
c Optimization:
c Improved efficiency for electricfield().
c
c 20021004
c
c Improvement:
c Added linear loading. More output during computation.
c
c 20020921
c
c Improvement:
c changed the integration from Hermite to Legendre. Improved smoothness.
c
c 20020916
c
c New Version:
c Distribution function added in both fields.
c
c 20020911
c
c Improvement:
c  $\exp(x^2) \operatorname{erfc}(x)$  is now approximated as  $2\sqrt{\pi}x^3/(2x^2-1)$ .
c
```

```

c 20020910
c
c Minor change:
c The program now shows the time used to finish the simulation.
c Uses X05BAF in NAGLib.
c
c 20020909
c
c Improvement:
c PI is now calculated using X01AAF from NAG Library.
c
c 20020908
c
c Improvement:
c Better handling in the calculation of transition probabilities.
c This is the last stable version before adding distribution functions.

      include 'calerf.for'

      program pztdisp

      implicit none

*****//variables to track the time consumed.
      real*8 time_begin,time_end,cputime,simtime

      include 'quad.for'
      include 'load.for'
      integer n,lwork,liwork
      parameter (n=2*(nia+nib)*ni2,lwork=4*n*n+12*n+20,liwork=3*n+20)

      integer ijac,mljac,imas,iout,itol,mlmas,mumas,idid,i,mujac,ipar
      real*8 y(n),work(lwork)
      integer iwork(liwork)
      real*8 t,tend,h,rpar,rtol(n),atol(n),abstol,reltol,amp,freq

      common /load/  amp,freq,tend

      external fcn,jac,mas,solout

*****//open input and data output files
      call opfile

```

```
*****//initialize all variables
      call init(y,n,abstol,reltol)

*****//Vorbereitung fuer RADAU5
*****//Preparation for RADAU5

*****//initial time
      t = 0.0d0
*****//initial timestep size, 0.00001 by default
      h = 1.0d-5
*****//Please refer to the RADAU5 documentation
*****//for the following parameters
      rpar = 1.0d-6
      ijac = 0
      mljac = n
      imas = 0
      iout = 1
      itol = 1
      imas = 0
      mlmas = 0
      mumas = 0
      idid = 0
      do 10 i=1,20
          iwork(i) = 0
          work (i) = 0.0d0
      10  continue
*****//maximum number of allowed steps
iwork(2)= 500000
*****//absolute tolerances and relative tolerances
do 50 i = 1,n
atol(i) = abstol
rtol(i) = reltol
50 continue

*****//the real simulation starts here
*****//mark the start time
      call cpu_time(time_begin)

* sinewave load:
* call the solver for every 1/4 cycle of the sine wave so that
* the solver can estimate the step sizes more easily.
* do 40 i=1,int(4.0d0*tend*freq)
```

```

*      call radau5(n,fcn,(i-1)/freq/4.0d0,y,i/freq/4.0d0,h,

* Unknown load:
* call radau5(n,fcn,t,y,tend,h,

*****//Apply a series of linear load
      if (loadtype.eq.2) then
*****//basefield and basetime are defined in load.for
*****//They are for multiple linear loads
      basefield = 0.0d0
      basetime = 0.0d0
      do i = 1,loadcount
        currentload = i
        print*,''
        print '(A,I2,A,F9.2,A,F9.2,A)', 'Computing load #',i,
&          ',t=[',t,',',basetime+period(i),']'
        call radau5(n,fcn,t,y,basetime+period(i),h,
&          rtol,atol,itol,
&          jac,ijac,mljac,mujac,
&          mas,imas,mlmas,mumas,
&          solout,iout,
&          work,lwork,iwork,liwork,rpar,ipar,idid)
      h = 1.0d-5
      basetime = basetime + period(i)
      basefield = basefield + rate(i)*period(i)
      end do
      tend = t
    endif

*****//Mark the end time and calculate the CPU time
call cpu_time(time_end)
cputime = time_end-time_begin
print *,''
print *,''
print '(A)', 'Solver ended successfully.'
print '(A,f9.3,A)', 'CPU time: ',cputime,' second(s)'
print '(A,f9.3)', 'CPU time/Sim time: ',cputime/tend

*****//close all data files
      call clfile
      stop
      end

```

```

c *****
      subroutine init(y,n,atol,rtol)

      implicit none

      integer n,iodat

      include 'quad.for'
      include 'load.for'
      integer itype,ifail,i,j
      real*8 d0lbbf
      external d0lbbf,d01baw,x01aaf,d01baz

      *****//vd: domain volume
      *****//tx(abbr. for TauX): relaxation time
      real*8 temp0,pistar,pi(nia+nib),pt,ei(nia+nib)
      real*8 s,vd,tx,gaussian(nia+nib)
      real*8 amp,freq,xplus0,tend,y(n),atol,rtol
      real*8 mathpi,boltzmann,x01aaf,nag_erf
      real*8 sigma,sigma2,x

      common /param/ temp0,ei,pi,pt,s,vd,tx
      common /load/  amp,freq,tend
      common /const/ mathpi,boltzmann

      read (20,1001)
      read (20,1001)
      read (20,1001)
      read (20,1001)
      read (20,1001) temp0,vd,tx,b,a
      read (20,1001)
      read (20,1001)
      read (20,1001)
      read (20,1001) pistar,pt,s,b2,a2
      read (20,1001)
      read (20,1001)
      read (20,1001)
      read (20,1001) amp,freq,xplus0,c
      read (20,1001)
      read (20,1001)

```

```
        read (20,1001)
        read (20,1001) tend,atol,rtol

1001 format (5d14.6)
1002 format (i14,4d14.6)

*****//Open the loading data file for reading
*****//"load.in": load parameter input file
        open (40,file = './load.in',
&           form = 'formatted',
&           status = 'unknown',
&           iostat = iostat)
        if (iostat.ne.0) then
print *, '...error opening loading input file...'
        stop
        endif
        rewind (40)

*****//get the type and number of loading
*****//skip the first 2 lines of the input file
read (40,1003)
read (40,1003)
read (40,1003) loadtype,loadcount
if (loadtype .eq. 2) then
*****//check for maximum indices
        if (loadcount .gt. maxcount) then
            print 1005,'ERROR: Too many input records. Only ',
&                maxcount,'records are allowed.'
                stop
        endif
        read (40,1003)
        read (40,1003)
        read (40,1003)
        read (40,1004) (rate(i),period(i),i=1,loadcount)
endif
1003 format (2i14)
1004 format (2d14.6)
1005 format (a,i3,a)
close (40)

*****//Mathematical & Physical constants:
boltzmann = 1.3806503d-23
```

```

mathpi = x01aaf(mathpi)

*****//Anfangswerte
*****//Parameters and initial values:

b = 1/b
b2 = 1/b2

sigma = 1/sqrt(2*b)
sigma2 = 1/sqrt(2*b2)

*****//The normalizing constant C
      c = sqrt(b/mathpi)*sqrt(b2/mathpi)
      &      /((1+nag_erf(-a*sqrt(b)))/2.0d0
      &      -(1+nag_erf(-3*sigma*sqrt(b)))/2.0d0)

*****//Weights and abscissae for both distributions
itype = 1
ifail = 0

      call d01bbf(d01baz,a-3*sigma,a,itype,nia,weight,ei,ifail)

if (a+3*sigma.gt.0.0d0) then
  call d01bbf(d01baz,a,0.0d0,itype,nib,
    &          weight(nia+1),ei(nia+1),ifail)
else
  call d01bbf(d01baz,a,a+3*sigma,itype,nib,
    &          weight(nia+1),ei(nia+1),ifail)
endif

call d01bbf(d01baz,a2-3*sigma2,a2,
  &          itype,ni2,weight2,abscis2,ifail)

call d01bbf(d01baz,a2,a2+3*sigma2,
  &          itype,ni2,weight2(ni2+1),abscis2(ni2+1),ifail)

*****//generate the distributions plot data
  do i = 0,100
    x = ei(nia)+real(i)*(ei(nia+1)-ei(nia))/100.0d0
    write (60,2002) -x,exp(-b*(x-a)**2)*sqrt(b/mathpi)
  end do

```

```

        do i = 0,100
        x = abscis2(ni2)+real(i)*(abscis2(ni2+1)-abscis2(ni2))/100.0d0
        write (70,2002) x,exp(-b2*(x-a2)**2)*sqrt(b2/mathpi)
        end do

*****//The format string for distributions plot data file
2002 format (2e15.6e3)

*****//Initialize all the phase fractions
        do i = 1,nia+nib
        write (50,2001) i,-ei(i),
        &      exp(-b*(ei(i)-a)**2)*sqrt(b/mathpi),weight(i)
        do j = 1,2*ni2
        weightTable(i,j) = weight(i)*weight2(j)
        &      *exp(-b*(ei(i)-a)**2-b2*abscis2(j)**2)
        if (abscis2(j).lt.0.0d0) then
        y((j-1)*(nia+nib)+i) = 0.0d0
        else
        y((j-1)*(nia+nib)+i) = 1.0d0
        endif
        end do
        end do

        do 103 j = 1,2*ni2
        write (80,2001) j,abscis2(j),
        &      exp(-b2*abscis2(j)**2)*sqrt(b2/mathpi),weight2(j)
103 continue

*****//The format string for integration points plot data file
2001 format (i14,3e15.6e3)

return
end

c *****
subroutine mas(n,am,lmas)
return
end

```

```

c *****
      subroutine jac(n,t,y,dfy,ldfy)
      return
      end

c *****

      subroutine fcn (n,t,y,f)

      implicit none

include 'quad.for'
      integer n,i
      real*8 t,y(n),f(n)
      real*8 xplus,ppm(2*(nia+nib)*ni2),pmp(2*(nia+nib)*ni2),efield
common /probab/ ppm,pmp

      call electricfield(efield,t)
      call probabilities(ppm,pmp,efield)

do 201 i = 1,n
      f(i) = pmp(i)-y(i)*(ppm(i)+pmp(i))
201 continue

      return
      end

c *****

      subroutine solout (nr,told,t,y,cont,lrc,n,rpar,ipar,irtrn)

      implicit none

include 'quad.for'
include 'load.for'
      integer n,nr,irtrn,lrc,ipar,i
      real*8 told,t,y(n),cont(lrc),rpar
real*8 efield,pol,xplus,xminus
real*8 ppm(2*(nia+nib)*ni2),pmp(2*(nia+nib)*ni2)
common /probab/ ppm,pmp

```

```

call electricfield(efield,t)
call polarization(pol,efield,y)

*****// note: xplus is now the phase fraction of the largest distribution
xplus = y((n+1)/2)
xminus = 1.0d0 - xplus

print '(a,$)', '.'
if(currentload.gt.1) then
    write (30,3001) t,xplus,xminus,efield,pol,2.293d0*pol**2
endif
3001 format(6e15.6e3)

    return
    end

c *****
*****//this routine returns the electric field at time t

subroutine electricfield (efield,t)

implicit none

include 'quad.for'
include 'load.for'

real*8 efield,t,field
real*8 amp,freq,tend
real*8 mathpi,boltzmann
common /load/ amp,freq,tend
common /const/ mathpi,boltzmann

c efield = amp*sin(2.0d0*mathpi*freq*t)

efield = basefield + rate(currentload)*(t-basetime)

return
end

c *****
*****//calculate polarization from electric field and phase fraction

```

```

subroutine polarization (pol,efield,xplus)

implicit none

include 'quad.for'

integer i,j
real*8 pol,efield,xplus(2*(nia+nib)*ni2)
real*8 temp0,ei(nia+nib),pi(nia+nib),pt,s,vd,tx
real*8 mathpi,boltzmann

common /param/ temp0,ei,pi,pt,s,vd,tx
common /const/ mathpi,boltzmann

*****//Sum the integral
*****//The weightTable has been initialized in init()
pol = 0.0d0
do 401 i = 1,nia+nib
  do 402 j = 1,2*ni2
    pol = pol+weightTable(i,j)
      & *((efield+abscis2(j))/s
      & +(xplus((j-1)*(nia+nib)+i)*2.0d0-1.0d0)*pt)
  402 continue
  401 continue

*****//Normalize
pol = pol*c

return
end

c *****
      subroutine probabilities (ppm,pmp,efield)

implicit none

include 'quad.for'

integer i,j
real*8 ppm(2*(nia+nib)*ni2),pmp(2*(nia+nib)*ni2),efield
real*8 temp0,ei(nia+nib),pi(nia+nib),pt,s,vd,tx

```

```

real*8 mathpi,boltzmann,derfcx
real*8 denom

common /param/ temp0,ei,pi,pt,s,vd,tx
common /const/ mathpi,boltzmann

*****//denom is the common denominator in the equation
denom = sqrt(2.0d0*s*boltzmann*temp0/vd)

do 701 i = 1,nia+nib
  do 702 j = 1,2*ni2
    ppm((j-1)*(nia+nib)+i)=1/derfcx((ei(i)-(efield+abscis2(j)))
    &                               /denom)/tx
    pmp((j-1)*(nia+nib)+i)=1/derfcx((ei(i)+(efield+abscis2(j)))
    &                               /denom)/tx
  702 continue
701 continue

      return
      end

c *****
      subroutine opfile

      implicit none

      integer ioplot,iodat

* "pzt.in": parameter input file

      open (20,file = './pzt.in',
&         form = 'formatted',
&         status = 'unknown',
&         iostat = iodat )

      if (iodat.ne.0) then
c -----> print*, '.....fehler beim oeffnen der Eingabedatei...'
print*, '.....error opening input file...'
          stop
        endif

```

```

        rewind (20)

* "pztdisp.plt": plot data output file

        open (30,file = './pztdisp.plt',
&          form = 'formatted',
&          status = 'unknown',
&          iostat = ioplot )

        if (ioplot.ne.0) then
c -----> print*, ' .....fehler beim oeffnen der Plotdatei...'
        print*, ' .....error opening plot file...'
        stop
        endif

        rewind (30)

* "abscis1.plt": gaussian distribution curve data ouput file

        open (50,file = './abscis1.plt',
&          form = 'formatted',
&          status = 'unknown',
&          iostat = ioplot )

        if (ioplot.ne.0) then
c -----> print*, ' .....fehler beim oeffnen der Plotdatei...'
        print*, ' .....error opening gaussian plot file...'
        stop
        endif

        rewind (50)

* "dist1.plt": coercive field distribution curve data ouput file

        open (60,file = './dist1.plt',
&          form = 'formatted',
&          status = 'unknown',
&          iostat = ioplot )

        if (ioplot.ne.0) then
c -----> print*, ' .....fehler beim oeffnen der Plotdatei...'
        print*, ' .....error opening gaussian plot file...'

```

```

        stop
        endif

        rewind (60)

* "dist2.plt": effective field distribution curve data ouput file

        open (70,file = './dist2.plt',
&           form = 'formatted',
&           status = 'unknown',
&           iostat = ioplot )

        if (ioplot.ne.0) then
c ----> print*, ' .....fehler beim oeffnen der Plotdatei...'
        print*, ' .....error opening gaussian plot file...'
        stop
        endif

        rewind (70)

* "abscis2.plt": gaussian distribution curve data ouput file

        open (80,file = './abscis2.plt',
&           form = 'formatted',
&           status = 'unknown',
&           iostat = ioplot )

        if (ioplot.ne.0) then
c ----> print*, ' .....fehler beim oeffnen der Plotdatei...'
        print*, ' .....error opening gaussian plot file...'
        stop
        endif

        rewind (80)

        return
        end

c *****

        subroutine clfile

```

```
        implicit none

        close (20)
        close (30)
        close (50)
close (60)
close (70)
close (80)

        return
        end

c *****
c error function wrapper
c function name changed to nag_erf, which is what this function is
c called in the NAG F90 Library.

        real*8 function nag_erf (x)

        implicit none

        integer ifail
real*8 x,s15aef

        nag_erf = s15aef(x,ifail)

        return
        end

c *****
c complementary error function wrapper
c function name changed to nag_erfc, which is what this function is
c called in the NAG F90 Library.

        real*8 function nag_erfc (x)

        implicit none

        integer ifail
real*8 x,s15adf
```

```
    ifail = 0
    nag_erfc = s15adf(x,ifail)

    return
end
```

Additional include files:

load.for

```
integer loadtype,loadcount,currentload,maxcount
  parameter (maxcount = 30)
  real*8 rate(maxcount), period(maxcount), basefield, basetime
  common /loading/ basefield,basetime,rate,period,
  &                loadtype,loadcount,currentload
```

quad.for

```
integer nia,nib,ni2
  parameter (nia = 10,nib = 6,ni2 = 8)
  real*8 weightTable(nia+nib,2*ni2)
  real*8 weight(nia+nib),b,a,weight2(2*ni2)
  real*8 abscis2(2*ni2),b2,a2,c
  common /gauss/ weightTable,weight,b,a,weight2,abscis2,b2,a2,c
```

A.2 Parametrization Method

Main program:

```

c PZT Simulation Program V.7.4
c
c Revision: 20030713
c
c CODE FREEZE
c

      include 'specfun.fi'

      program pztdisp

      implicit none
      include 'distrib.fi'
      include 'linload.fi'
include 'history.fi'

*****//variables to track the time consumed.
      real*8 time_begin,time_end,cputime,simtime

      integer n,lwork,liwork
      parameter (n=1,lwork=4*n*n+12*n+20,liwork=3*n+20)

      integer ijac,mljac,imas,iout,itol,mlmas,mumas,idid,i,mujac,ipar
      real*8 y(n),work(lwork)
      integer iwork(liwork)
      real*8 t,tend,h,rpar,rtol(n),atol(n),abstol,reltol,amp,freq
real*8 temp1,temp2,temp3,temp4

      common /load/ amp,freq,tend

      external fcn,jac,mas,solout,soloutdummy

*****//open input and data output files
      call openfile

*****//initialize all variables
      call init(y,n,abstol,reltol)

*****//Preparation for RADAU5

```

```
*****//initial time
      t = 0.0d0

*****//initial timestep size, 0.00001 by default
      h = 1.0d-5

*****//Please refer to the RADAU5 documentation
*****//for the following parameters
      rpar = 1.0d-6
      ijac = 0
      mljac = n
      imas = 0
      iout = 1
      itol = 1
      imas = 0
      mlmas = 0
      mumas = 0
      idid = 0

      do i=1,20
         iwork(i) = 0
         work (i) = 0.0d0
      end do

*****//maximum number of allowed steps
iwork(2)= 500000

*****//absolute tolerances and relative tolerances
do i = 1,n
   atol(i) = abstol
   rtol(i) = reltol
end do

*****//set virgin curve
      if (.true.) then

      temp1 = loadtype
      loadtype = 3
         call radau5(n,fcn,t,y,1.0d0,h,
&          rtol,atol,itol,
&          jac,ijac,mljac,mujac,
```

```

      &      mas,imas,mlmas,mumas,
      &      soloutdummy,iout,
      &      work,lwork,iwork,liwork,rpar,ipar,idid)
t = 0.0d0
      h = 1.0d-5
loadtype = temp1

      endif

*****//the real simulation starts here
*****//mark the start time
      call cpu_time(time_begin)

      if (loadtype.eq.1) then
*****//Apply a series of sinusoidal load
      call radau5(n,fcn,t,y,tend,h,
      &      rtol,atol,itol,
      &      jac,ijac,mljac,mujac,
      &      mas,imas,mlmas,mumas,
      &      solout,iout,
      &      work,lwork,iwork,liwork,rpar,ipar,idid)

      elseif (loadtype.eq.2) then
*****//Apply a series of linear load
*****//basefield and basetime are defined in linload.fi
*****//for multiple linear loads
      basefield = 0.0d0
      basetime = 0.0d0
      do i = 1,loadcount
      currentload = i
      print '(A)',
      &      '-----',
      print '(A,I2,A,F9.2,A,F9.2,A)', 'Computing linear load #',i,
      &      ',t=[',t,',',',basetime+period(i),']'
      call radau5(n,fcn,t,y,basetime+period(i),h,
      &      rtol,atol,itol,
      &      jac,ijac,mljac,mujac,
      &      mas,imas,mlmas,mumas,
      &      solout,iout,
      &      work,lwork,iwork,liwork,rpar,ipar,idid)
      basetime = basetime + period(i)
      basefield = basefield + rate(i)*period(i)

```

```

        end do
    tend = t

    end if

*****//Mark the end time and calculate the CPU time
call cpu_time(time_end)
cputime = time_end-time_begin

    print *,''
print '(A)', '==== End of computation ====='
    print '(A,E30.20)', 'xplus = ', y(1)
print '(A,f9.3,A)', 'CPU time: ', cputime, ' second(s)'
print '(A,f9.3)', 'CPU time/Sim time: ', cputime/tend

*****//close all data files
    call closefile
    stop
    end

C *****

    subroutine init(y,n,atol,rtol)

    implicit none

integer n,iodat

    include 'distrib.fi'
include 'linload.fi'
include 'history.fi'
integer itype,ifail,i,j
external x01aaf

*****//vd: domain volume
*****//tx(abbr. for TauX): relaxation time

real*8 temp0,pistar,pt,ei
real*8 s,vd,tx
real*8 amp,freq,tend,y(n),atol,rtol
real*8 ppm,ppm

```

```

real*8 mathpi,boltzmann,x01aaf,derfc
      real*8 sigma,sigma2,x
      real*8 ec_sat

common /param/ temp0,ei,pt,s,vd,tx
common /load/  amp,freq,tend
common /const/ mathpi,boltzmann
common /probab/ ppm,pmp

      read (20,1001)
      read (20,1001)
      read (20,1001)
      read (20,1001)
      read (20,1001) temp0,vd,tx,b,a
      read (20,1001)
      read (20,1001)
      read (20,1001)
      read (20,1001)
      read (20,1001) pistar,pt,s,b2,a2
      read (20,1001)
      read (20,1001)
      read (20,1001)
      read (20,1001) amp,freq,y(1),c,ec_sat
      read (20,1001)
      read (20,1001)
      read (20,1001)
      read (20,1001) tend,atol,rtol

1001 format (5d14.6)
1002 format (i14,4d14.6)

*****//get the type and number of loading
*****//skip the first 2 comment lines of the input file
read (40,1003)
read (40,1003)
read (40,1003) loadtype,loadcount
if (loadtype .eq. 2) then
*****//check for maximum number
      if (loadcount .gt. maxcount) then
          print '(a,i3,a)', 'ERROR: Too many input records. Only ',
            &                                maxcount, 'records are allowed.'
          stop

```

```

endif
read (40,1003)
read (40,1003)
read (40,1003)
read (40,1004) (rate(i),period(i),i=1,loadcount)
endif
1003 format (2i14)
1004 format (2d14.6)

*****//Mathematical & Physical constants:
boltzmann = 1.3806503d-23
mathpi    = x01aaf(0.0d0)

*****//Parameters and initial values:

sigma  = sqrt(0.5d0*b)
sigma2 = sqrt(0.5d0*b2)

*****//The normalizing constant C
c = 0.5d0*derfc(a/sqrt(b))

*****//Weights and abscissae for both distributions
itype = 1
ifail = 0

*****//generate the distributions plot data
do i = 0,int(-(a-3*sigma)/100.0d0)
  x = a-3*sigma+real(i)*100.0d0
  write (60,2002) x,exp(-(x-a)**2/b)/sqrt(b*mathpi)
end do

do i = 0,int(-(a-3*sigma2)/100.0d0)
  x = a2-3*sigma2+real(i)*100.0d0
  write (70,2002) x,exp(-(x-a2)**2/b2)/sqrt(b2*mathpi)
end do

*****//The format string for distribution plot data files
2002 format (2e15.6e3)

y(1) = 0.0d0
prev_xplus = 0.0d0
prev_ec = -ec_sat

```

```
prev_dir = INC

indexL = 1
indexH = 1

xplusL(1) = 0.0d0
bplusL(1) = -ec_sat

xplusH(1) = 1.0d0
bplusH(1) = ec_sat

return
end

C *****

      subroutine mas(n,am,lmas)
      return
      end

C *****

      subroutine jac(n,t,y,dfy,ldfy)
      return
      end

C *****
*****/phase fractions should never reach 0.0 or 1.0 so as not
*****/to crash the ODE or the inverse complementary error function
*****/or some other shape functions.

subroutine trimfraction(x)

implicit none

real*8 ZERO,ONE,x
parameter (ZERO = 0.0d0,ONE = 1.0d0)

if (x.lt.ZERO) then

  x = ZERO
```

```
elseif (x.gt.ONE) then

    x = ONE

endif

return
end

c *****

    subroutine fcn (n,t,y,f)

        implicit none

include 'distrib.fi'
include 'history.fi'
        integer n
        real*8 t,y(n),f(n),e,ec,emean
        real*8 xplus,ppm,pmp
common /probab/ ppm,pmp

call trimfraction(y(1))

xplus = y(1)

call electricfield(e,t)
call coercivefield(ec,xplus)
call probabilities(ppm,pmp,e,ec)

        f(1) = pmp - xplus*(ppm+pmp)

        return
    end

c *****

    subroutine soloutdummy (nr,told,t,y,cont,lrc,n,rpar,ipar,irtrn)

        implicit none

        include 'history.fi'
```

```

        integer n,nr,irtrn,lrc,ipar,i
        real*8 told,t,y(n),cont(lrc),rpar
real*8 e,ec,xplus,xminus,pol
real*8 ppm,pmp
common /probab/ ppm,pmp
integer curr_dir

call trimfraction(y(1))

xplus = y(1)
xminus = 1.0d0-xplus

call electricfield(e,t)
call coercivefield(ec,xplus)
call probabilities(ppm,pmp,e,ec)
call polarization(pol,e,xplus)

*****// Judge the direction of loading from the change in phase fraction.
*****// If there is no change, assume the previous direction.

        if (xplus.gt.prev_xplus) then

            curr_dir = INC

        elseif (xplus.lt.prev_xplus) then

            curr_dir = DEC

        else

            curr_dir = prev_dir

        endif

*****// If we are changing direction, update number of records
if (curr_dir.ne.prev_dir) then

    prev_dir = curr_dir

    if (curr_dir.eq.INC) then

        if (indexL.eq.HISTORYSIZE) stop 'History array overflow'

```

```

    indexL = indexL + 1
        xplusL(indexL) = prev_xplus
    bplusL(indexL) = prev_ec

    print '(A,E24.16)', 'DEC to INC at X+ = ', prev_xplus
c    print '(A,E24.16)', 'Current X+ =      ', xplus

elseif (curr_dir.eq.DEC) then

    if (indexH.eq.HISTORYSIZE) stop 'History array overflow'
    indexH = indexH + 1
        xplusH(indexH) = prev_xplus
    bplusH(indexH) = prev_ec

    print '(A,E24.16)', 'INC to DEC at X+ = ', prev_xplus
c    print '(A,E24.16)', 'Current X+ =      ', xplus

endif

endif

*****// Close minor loops

if (xplus.gt.xplusH(indexH)) then

*****// Underflow has rarely occurred, so this check is mostly
*****// redundant. It's needed only when xplusL(1) and xplusH(1)
*****// are not zeros.
        if (indexL.eq.1.or.indexH.eq.1) stop 'History array underflow'
        indexL = indexL - 1
        indexH = indexH - 1

elseif (xplus.lt.xplusL(indexL)) then

        if (indexL.eq.1.or.indexH.eq.1) stop 'History array underflow'
        indexL = indexL - 1
        indexH = indexH - 1

endif

        prev_xplus = xplus
prev_ec = ec

```

```
prev_pol = pol

      return
end

      subroutine solout (nr,told,t,y,cont,lrc,n,rpar,ipar,irtrn)

      implicit none

      include 'history.fi'
      integer n,nr,irtrn,lrc,ipar,i
      real*8 told,t,y(n),cont(lrc),rpar
real*8 e,ec,xplus,xminus,pol
real*8 ppm,pmp
common /probab/ ppm,pmp
integer curr_dir

      call trimfraction(y(1))

      xplus = y(1)
      xminus = 1.0d0-xplus

      call electricfield(e,t)
      call coercivefield(ec,xplus)
      call probabilities(ppm,pmp,e,ec)
      call polarization(pol,e,xplus)

      *****// Judge the direction of loading from the change in phase fraction.
      *****// If there is no change, assume the previous direction.

      if (xplus.gt.prev_xplus) then

         curr_dir = INC

      elseif (xplus.lt.prev_xplus) then

         curr_dir = DEC

      else

         curr_dir = prev_dir
```

```

endif

*****// If we are changing direction, update number of records
if (curr_dir.ne.prev_dir) then

    if (curr_dir.eq.INC) then

        if (indexL.eq.HISTORYSIZE) stop 'History array overflow'
        indexL = indexL + 1
            xplusL(indexL) = prev_xplus
        bplusL(indexL) = prev_ec

        print '(A,E24.16)', 'DEC to INC at X+ = ', prev_xplus

    elseif (curr_dir.eq.DEC) then

        if (indexH.eq.HISTORYSIZE) stop 'History array overflow'
        indexH = indexH + 1
            xplusH(indexH) = prev_xplus
        bplusH(indexH) = prev_ec

        print '(A,E24.16)', 'INC to DEC at X+ = ', prev_xplus

    endif

*****//Print out the state of the history array
print '(A,I3,e30.20)', 'xplusL: ', indexL, xplusL(indexL),
    & 'xplusH: ', indexH, xplusH(indexH)
print '(A,I3,e30.20)', 'bplusL: ', indexL, bplusL(indexL),
    & 'bplusH: ', indexH, bplusH(indexH)

endif

*****// Close minor loops

if (xplus.gt.xplusH(indexH)
    & .and.indexL.gt.1.and.indexH.gt.1) then

    print '(A)', 'Closing a minor loop when ascending'
    indexL = indexL - 1
    indexH = indexH - 1

```

```

elseif (xplus.lt.xplusL(indexL)
      &      .and.indexL.gt.1.and.indexH.gt.1) then

      print '(A)', 'Closing a minor loop when descending'
      indexL = indexL - 1
      indexH = indexH - 1

endif

      write (30,3001) t,xplus,xminus,e,pol,2.293*pol**2,ppm,pmp,
      &      ppm-pmp,ec,prev_ec

3001  format(14e30.20e3)

      prev_dir = curr_dir
      prev_xplus = xplus
prev_ec = ec
prev_pol = pol

      return
      end

c *****
*****//electric field at time t
subroutine electricfield (e,t)

implicit none

include 'linload.fi'

real*8 e,t

real*8 amp,freq,tend
common /load/ amp,freq,tend

real*8 mathpi,boltzmann
common /const/ mathpi,boltzmann

      if (loadtype.eq.1) then

          e = amp*sin(2.0d0*mathpi*freq*t)
      &      *(tend-t)/tend

```

```

c      &          +amp*t/tend

      elseif (loadtype.eq.2) then

          e = basefield + rate(currentload)*(t-basetime)

      elseif (loadtype.eq.3) then

          e = amp*sin(2.0d0*mathpi*10*t)
      &          *(1.0d0-t)/1.0d0

end if

return
end

c *****
*****//calculate polarization from electric field and phase fraction

subroutine polarization (pol,e,xplus)

implicit none

include 'distrib.fi'
real*8 pol,e,xplus
real*8 temp0,ei,pt,s,vd,tx
common /param/ temp0,ei,pt,s,vd,tx

      pol = e/s + (2.0d0*xplus-1.0d0)*pt

return
end

c *****

      subroutine probabilities (ppm,pmp,e,ec)

implicit none

real*8 ppm,pmp,e,ec
real*8 denom,derfcx

```

```

real*8 temp0,ei,pt,s,vd,tx
common /param/ temp0,ei,pt,s,vd,tx
real*8 mathpi,boltzmann
common /const/ mathpi,boltzmann

*****//denom is the common denominator of the term inside derfcx()
      denom = sqrt(2.0d0*s*boltzmann*temp0/vd)
      pmp = 1/derfcx((e-ec)/denom)/tx
      ppm = 1/derfcx((ec-e)/denom)/tx

      return
      end

C *****
subroutine coercivefield(ec,xplus)

implicit none
include 'history.fi'
integer direction,iL,iH
real*8 derfc,dierfc
real*8 ec,xplus,bmean,blower,bupper,basepoint
real*8 x,xlower,xupper,xtemp,prev_xminus,xminus
real*8 scale,s1,s2

      scale = 0.9d0
      bmean = 0.5d6

      iL = indexL
iH = indexH

      if (xplus.gt.xplusH(iH)) then

direction = INC
      do while (xplus.gt.xplusH(iH))
        iH = iH - 1
      end do
blower = bplusH(iH+1)
bupper = bplusH(iH)
      xtemp = (xplus-xplusH(iH+1))
&          /(xplusH(iH)-xplusH(iH+1))

```

```
        elseif (xplus.lt.xplusL(iL)) then

direction = DEC
        do while (xplus.lt.xplusL(iL))
            iL = iL - 1
        end do
        blower = bplusL(iL)
        bupper = bplusL(iL+1)
            xtemp = (xplus-xplusL(iL))
            &          /(xplusL(iL+1)-xplusL(iL))

        elseif (xplus.eq.xplusL(iL)) then

ec = bplusL(iL)
return

        elseif (xplus.eq.xplusH(iH)) then

ec = bplusH(iH)
return

        elseif (xplus.gt.prev_xplus) then

            direction = INC
            blower = prev_ec
            bupper = bplusH(iH)
            xtemp = (xplus-prev_xplus)
            &          /(xplusH(iH)-prev_xplus)

elseif (xplus.lt.prev_xplus) then

            direction = DEC
            bupper = prev_ec
            blower = bplusL(iL)
            xtemp = (xplus-xplusL(iL))
            &          /(prev_xplus-xplusL(iL))

        else

ec = prev_ec
return
```

```

endif

*****//translate coordinates

*****//shapefunction using inverse error function
c   if(.false.)then
      if (direction.eq.INC) then

cjh**//these are special parameters to fit the cigars
c   scale = 2.8d0 !2.4d0
c   bmean = 0.25e6
      xlower = derfc((1.0d0-blower/bmean)/scale)
      xupper = derfc((1.0d0-bupper/bmean)/scale)
      call trimfraction(xtemp)
      x = xlower + xtemp*(xupper-xlower)
      ec = bmean*(1.0d0-scale*dierfc(x))

else

c   scale = 1.1d0 !1.25d0
c   bmean = 0.9e6
      xlower = derfc((-1.0d0-blower/bmean)/scale)
      xupper = derfc((-1.0d0-bupper/bmean)/scale)
      call trimfraction(xtemp)
      x = xlower + xtemp*(xupper-xlower)
      ec = bmean*(-1.0d0-scale*dierfc(x))

endif
c   endif

*****//shapefunction using tangent function
      if(.false.)then

          s1 = 0.3
s2 = 1.57

          if (direction.eq.INC) then

c   bmean = 0.3e6
c   s1 = 1.5
c   s2 = 1.0
      xlower = 0.5d0*(atan((blower/bmean-1.0d0)/s1)/s2+1.0d0)

```

```

        xupper = 0.5d0*(atan((bupper/bmean-1.0d0)/s1)/s2+1.0d0)
    call trimfraction(xtemp)
        x = xlower + xtemp*(xupper-xlower)
        ec = bmean*(1.0d0+s1*tan(s2*(2.0d0*x-1)))

else

c       bmean = 0.5e6
c       s1 = 0.4
c       s2 = 1.2
        xlower = 0.5d0*(atan((blower/bmean+1.0d0)/s1)/s2+1.0d0)
        xupper = 0.5d0*(atan((bupper/bmean+1.0d0)/s1)/s2+1.0d0)
    call trimfraction(xtemp)
        x = xlower + xtemp*(xupper-xlower)
        ec = bmean*(-1.0d0+s1*tan(s2*(2.0d0*x-1)))

endif
endif

*****// This should not happen but if it does...
        if (xplus.lt.prev_xplus.and.ec.gt.prev_ec) stop 'grrr!'
        if (xplus.gt.prev_xplus.and.ec.lt.prev_ec) stop 'grrr!'

return
end

c *****
subroutine openfile

implicit none

integer ioplot,iodat

* "pzt.in": parameter input file

        open (20,file = './pzt.in',
&           form = 'formatted',
&           status = 'unknown',
&           iostat = iodat )

        if (iodat.ne.0) then

```

```
print *, '.....error opening input file...'
      stop
    endif

      rewind (20)

* "pztdisp.plt": plot data output file

      open (30,file = './pztdisp.plt',
&          form = 'formatted',
&          status = 'unknown',
&          iostat = ioplot )

      if (ioplot.ne.0) then
        print *, '.....error opening plot file...'
          stop
        endif

      rewind (30)

*****//Open the loading data file for reading
*****//"load.in": load parameter input file
      open (40,file = './load.in',
&          form = 'formatted',
&          status = 'unknown',
&          iostat = iodat)

      if (iodat.ne.0) then
        print *, '....error opening loading input file...'
          stop
        endif

      rewind (40)

* "gaussian.plt": gaussian distribution curve data ouput file

      open (50,file = './gaussian.plt',
&          form = 'formatted',
&          status = 'unknown',
&          iostat = ioplot )

      if (ioplot.ne.0) then
```

```
    print *, '.....error opening gaussian plot file...'
    stop
  endif

  rewind (50)

* "dist1.plt": coercive field distribution curve data ouput file

  open (60,file = './dist1.plt',
&      form = 'formatted',
&      status = 'unknown',
&      iostat = ioplot )

  if (ioplot.ne.0) then
    print *, '.....error opening gaussian plot file...'
    stop
  endif

  rewind (60)

* "dist2.plt": effective field distribution curve data ouput file

  open (70,file = './dist2.plt',
&      form = 'formatted',
&      status = 'unknown',
&      iostat = ioplot )

  if (ioplot.ne.0) then
    print *, '.....error opening gaussian plot file...'
    stop
  endif

  rewind (70)

* "abscis2.plt": gaussian distribution curve data ouput file

  open (80,file = './abscis2.plt',
&      form = 'formatted',
&      status = 'unknown',
&      iostat = ioplot )

  if (ioplot.ne.0) then
```

```

    print *, '.....error opening gaussian plot file...'
    stop
endif

    rewind (80)

    return
end

C *****

    subroutine closefile

    implicit none

    close (20)
    close (30)
close (40)
    close (50)
close (60)
close (70)
close (80)

    return
end

Additional include files:
distrib.fi

real*8 a,b,c,a2,b2,c2
    common /distribution/ a,b,c,a2,b2,c2

    history.fi

***** Variables for the history stack

integer HISTORYSIZE,INC,DEC,indexL,indexH
parameter (HISTORYSIZE = 100000,INC = 1,DEC = -1)
real*8 xplusL(HISTORYSIZE),xplusH(HISTORYSIZE)
real*8 eplusL(HISTORYSIZE),eplusH(HISTORYSIZE)
real*8 bplusL(HISTORYSIZE),bplusH(HISTORYSIZE)
real*8 prev_dir,prev_pol,prev_xplus,prev_ec
common /history/ xplusL,xplusH,eplusL,eplusH,bplusL,bplusH,
    &          indexL,indexH,prev_dir,prev_pol,prev_xplus,prev_ec

```

linload.fi

```
integer loadtype,loadcount,currentload,maxcount
  parameter (maxcount = 100)
  real*8 rate(maxcount), period(maxcount), basefield, basetime
  common /loading/ basefield,basetime,rate,period,
&                loadtype,loadcount,currentload
```

Appendix B

Matlab Model and Control Script

B.1 Simulink Model for the Sinusoidal Load Experiments

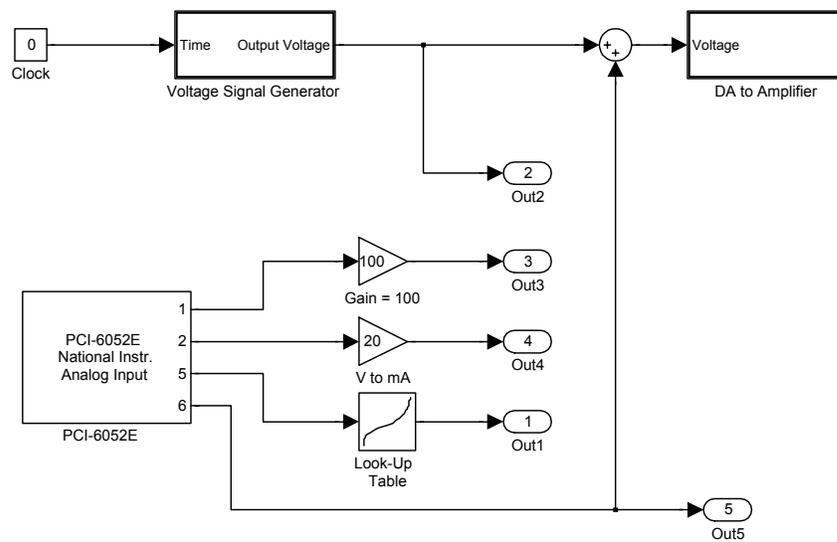


Figure B.1: Main model for sinusoidal load experiments

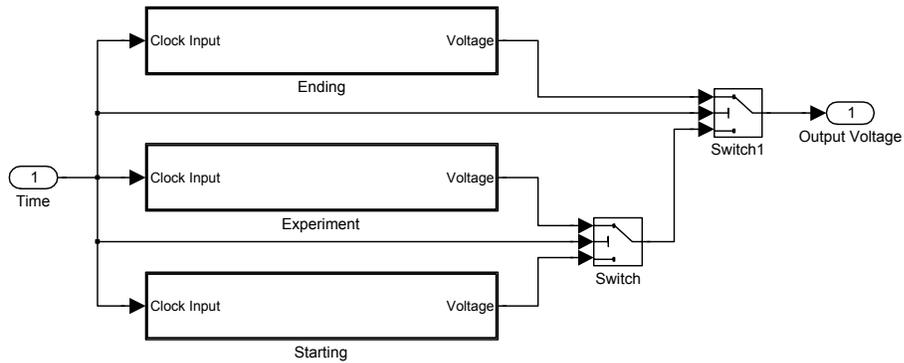


Figure B.2: Signal generation part of the model for sinusoidal loads.

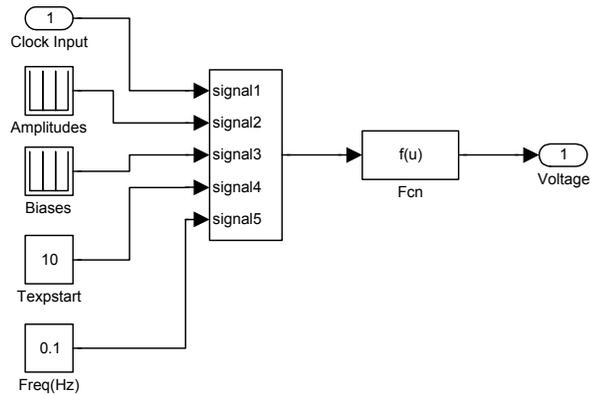


Figure B.3: Subsystem for generating the sinusoidal signal with changing amplitudes and offsets.

B.2 Matlab Control Script for the Model

```
disp('Sinusoidal experiments running.');
```

```
clear all;
```

```
dischargetime = 30;
```

```
xsin = 0;  
asin = 1;  
bsin = 2;  
csin = 3;  
dsin = 4;  
filename_prefix = 'raw';  
maxfreqhz = 100; % use 1 for xsin, 100 for abcd.
```

```
looptypes = [xsin];%[asin bsin csin dsin];
```

```
%butterflies:  
x_freqhzs = [0.1 1];  
x_rev_points = [-150];% actually unused.
```

```
%cigars:  
a_freqhzs = [0.1 1 10 100];  
a_rev_points = [-40 -20 0 20 50 100];
```

```
b_freqhzs = [0.1 1 10 100];  
b_rev_points = [-20 0 20 50 100 150];
```

```
c_freqhzs = [0.1 1 10 100];  
c_rev_points = [-40 -20 0 20 50 100];
```

```
d_freqhzs = [0.1 1 10 100];  
d_rev_points = [20 50 100 150];
```

```
for kr = 1:length(looptypes)
```

```
    looptypeid = looptypes(kr);  
    if (looptypeid == xsin)  
        looptype = 'xsin';  
        freqhzs = x_freqhzs;  
        rev_points = x_rev_points;
```

```

elseif (looptypeid == asin)
    looptype = 'asin';
    freqhzs = a_freqhzs;
    rev_points = a_rev_points;
elseif (looptypeid == bsin)
    looptype = 'bsin';
    freqhzs = b_freqhzs;
    rev_points = b_rev_points;
elseif (looptypeid == csin)
    looptype = 'csin';
    freqhzs = c_freqhzs;
    rev_points = c_rev_points;
elseif (looptypeid == dsin)
    looptype = 'dsin';
    freqhzs = d_freqhzs;
    rev_points = d_rev_points;
end

for ir = 1:length(freqhzs)
for jr = 1:length(rev_points)

% define target application parameters
appname = 'pzt_analog_sine';
sampletime = 0.00005;

% define experiment parameters
fs = 1/sampletime; % fs is the sampling frequency.
v_start = 150;

freqhz = freqhzs(ir);
rev_point = rev_points(jr);
if (looptypeid == xsin)
    v_setpoints = [-v_start v_start];
% no more than 10 setpoints.
% to have more, change the model block.
elseif (looptypeid == asin)
    v_setpoints = [rev_point v_start];
elseif (looptypeid == bsin)
    v_setpoints = [-40 rev_point -40];
elseif (looptypeid == csin)
    v_setpoints = [rev_point rev_point+50 rev_point];
elseif (looptypeid == dsin)

```

```

    v_setpoints = [-40 rev_point rev_point-50 rev_point];
end

inittime = 5.0; % time for the voltage to rise from zero to v_start.
inithold = 10.0; % time to hold at v_start before the experiment.
endhold = 10.0; % time to hold after reaching the last voltage.
returntime = 5.0; % time for the voltage to return to zero.
exptime = length(v_setpoints)*0.5/freqhz; % duration of the real experiment.

timearray = [0 (inittime+inithold) (inittime+inithold)
             (inittime+inithold+0.5/freqhz)];
amplitudes = [v_start/2 v_start/2 abs(v_setpoints(1)-v_start)/2
             abs(v_setpoints(1)-v_start)/2];
biases = [v_start/2 v_start/2 (v_setpoints(1)+v_start)/2
          (v_setpoints(1)+v_start)/2];

for i = 2:length(v_setpoints)
    timearray(2*i+1) = timearray(2*i);
    timearray(2*i+2) = timearray(2*i)+0.5/freqhz;
    amplitudes(2*i+1) = abs(v_setpoints(i)-v_setpoints(i-1))/2;
    amplitudes(2*i+2) = abs(v_setpoints(i)-v_setpoints(i-1))/2;
    biases(2*i+1) = (v_setpoints(i)+v_setpoints(i-1))/2;
    biases(2*i+2) = (v_setpoints(i)+v_setpoints(i-1))/2;
end

timearray = [timearray timearray(length(timearray))
            *ones(1,22-length(timearray))];
amplitudes = [amplitudes amplitudes(length(amplitudes))
            *ones(1,22-length(amplitudes))];
biases = [biases biases(length(biases))
          *ones(1,22-length(biases))];

dec_factor = maxfreqhz/freqhz;
fs = fs/dec_factor;

stoptime = inittime + inithold + exptime + endhold + returntime;

% initialize target;
tg = xpc;
if(strcmp(tg.Application,appname)~=1)
    tg.load(appname);
    pause(5.0);

```

```

end

tg.set('p10',inittime+inithold);
tg.set('p11',inittime+inithold+exptime+endhold);
tg.set('p12',returntime);
tg.set('p13',v_setpoints(length(v_setpoints)));
tg.set('p14',inittime+inithold+exptime+endhold);
tg.set('p15',timearray);
tg.set('p16',amplitudes);
tg.set('p17',timearray);
tg.set('p18',biases);
tg.set('p19',inittime+inithold);
tg.set('p20',freqhz);
tg.set('p21',v_start);
tg.set('p22',inittime);
tg.set('p23',inittime);
tg.set('p24',inittime+inithold+exptime);

% execute target application;
tg.stoptime = stoptime;
%tg.sampletime = sampletime;
tg.start;
pause(stoptime+5.0);

% download application results;
results = [tg.TimeLog,tg.OutputLog];
results = results(1:dec_factor:length(results),:);

% postprocess;
%time = results(:,1);
%displacement = results(:,2);
%voltage = results(:,3);
%ampvoltage = results(:,4);
%ampcurrent = results(:,5);
%refvoltage = results(:,6);

filename = sprintf('%s_type_%s_fs_%g_freqhz_%g_rev_%d.txt',
                  filename_prefix,looptype,fs,freqhz,rev_point);
save(filename,'results','-ASCII');

tg = xpc;

```

```
tg.load('discharger');  
pause(5.0);  
tg.stoptime = dischargetime;  
tg.start;  
pause(dischargetime+5.0);  
tg.reboot;  
pause(60.0);  
  
end  
end  
end  
  
disp('Experiments Finished.');
```