Cansmart 2007

International Workshop

SMART MATERIALS AND STRUCTURES

10 - 11 October 2007, Montreal, Quebec, Canada

FPGA-BASED MODEL IMPLEMENTATION FOR REAL-TIME CONTROL OF SMART MATERIAL SYSTEMS OPERATING IN HYSTERETIC REGIMES

Thomas R. Braun and Ralph C. Smith

Center for Research in Scientific Computation
Department of Mathematics
North Carolina State University
Raleigh, NC 27695
trbraun@ncsu.edu rsmith@ncsu.edu

ABSTRACT

Ferroelectric (e.g., PZT) and ferromagnetic materials exhibit varying degrees of hysteresis and constitutive nonlinearities at all drive levels due to their inherent domain structures. At low drive levels, these nonlinear effects can be mitigated through feedback mechanisms or certain amplifier architectures (e.g., charge or current control for PZT) so that linear models and control designs provide sufficient accuracy. However, at the moderate to high drive levels where actuators and sensors utilizing these compounds often prove advantageous, hysteresis and constitutive nonlinearities must be incorporated into models and control designs to achieve high accuracy, high speed control specifications. In this paper, we employ a homogenized energy framework to characterize hysteresis in this combined class of ferroic compounds, and we show that embedded microprocessors or desktop computers may not capable of computing this framework in real-time. This motivates the development of a field programmable gate array (FPGA) implementation of the homogenized energy model. We introduce one such implementation, and show that this implementation is capable or achieving a reasonable accuracy over 60 times faster than an Intel Pentium IV 3.8 GHz processor.

Keywords: Hysteresis, Homogenized energy model, Field programmable gate array, FPGA.

INTRODUCTION

Ferroelectric and ferromagnetic materials are employed in a wide variety of applications as both actuators and sensors, including fluid pumps, nanopositioning stages, sonar, vibration control, ultrasonic sources, and high-speed milling. They are attractive because the transducers are solid-state and often very compact. However, the coupling of nonmechanical field to mechanical deformation, which makes these materials effective transducers, also introduces hysteresis and time-dependent behaviour that must be accommodated by controllers. Classically, this has been attained by operating the transducers in only a small portion of their theoretical operating range and/or operating it at a low frequency. This allows controllers to accommodate nonlinear dynamics but reduces the performance of the transducer. More sophisticated approaches incorporate these dynamics through electromechanical or magnetomechanical models. In theory, this permits high-speed, high-accuracy control, but in practice introduces significant computational overhead which limits effectiveness.

Classically, accuracy and computational efficiency are inversely related. However, parallelization may permit the computational efficiency to be improved without trading off accuracy. To achieve the desired speed improvements, this dictates either a large network of microprocessors or a custom-designed field programmable gate array (FPGA) or application specific integrated circuit (ASIC). These require more development effort than microprocessors, but can deliver the needed parallelization within either a single physical chip or a small set of chips. FGPAs combine tens to hundreds of thousands of look-up-tables, flip-flops, and switches that can be programmed by a user to achieve a custom design. This allows single arithmetic operations to be done in parallel, if desired, and then recombined. However, since device resources must be allocated to each operation, integer arithmetic is preferable, where possible, to floating point arithmetic; the use of integer arithmetic allows more operations to be computed in parallel since each operation requires fewer resources. ASICs typically can allocate the resources to allow for floating point operations and yet be more compact than FPGAs; however, the design is etched into silicon once and cannot be changed. They are thus less-flexible and have a much higher up-front development cost. As such, we focus on FPGA implementations at present.

HOMOGENIZED ENERGY MODEL DEVELOPMENT

Several different models exist for ferroelectric and ferromagnetic materials. These include domain wall models [2, 4, 5, 12], Preisach models [3, 6-8, 15, 16], and homogenized energy models [1, 9-11, 13, 14]. We employ the homogenized energy model due to its solid basis in the underlying physics of the material and its applicability to both ferroelectric and ferromagnetic materials. Although not discussed here, the model can also be naturally extended to include thermal activation effects such as creep.

The homogenized energy model for ferroelectric materials is

$$P(E;x_+) = \int_0^\infty \int_{-\infty}^\infty \nu_c(E_c)\nu_I(E_I)\overline{P}(E+E_I;E_c,x_+)dE_I dE_c \tag{1}$$

where P is the polarization, E is the electric field, E_c denotes the coercive field value at which a dipole changes its orientations, and E_I quantifies dipole interactions; see [9] for details. Note that this is a multiscale approach; the mesoscopic polarization P is extended to the macroscopic scale by assuming two parameters (E_c and E_I) are manifestations of underlying stochastic distributions v_c and v_I. The model is obtained by integrating over these distributions, where both distributions are subject to the constitutive laws:

1. both v_c and v_I are bounded by decaying exponentials,
2. v_c is strictly positive,
3. v_I is symmetric about 0, and
4. both densities integrate to unity.

The integrals are solved numerically via quadrature, i.e.,

$$P(E;x_+) = \sum_{i=1}^{N_c}\sum_{j=1}^{N_I} v_c(E_c[i])v_I(E_I[j])w_c[i]w_I[j]\overline{P}(E+E_I[j];E_c[i],x_+[i,j]) \tag{2}$$

where w_c and w_I give the quadrature weights. As detailed in [9], the model for magnetic materials is equivalent. Therefore, to simplify discussion, we formulate equations solely in terms of the electric field and polarization, with the understanding that everything applies equally to ferromagnetic materials.

The kernel $\overline{P}$ is modeled through energy principles. The mesoscopic Helmholtz energy is taken to be

$$\psi(P) = \begin{cases} \eta(P+P_R)^2/2, & P \le -P_I \\ \dfrac{\eta}{2}(P_I - P_R)\left(\dfrac{P^2}{P_I} - P_R\right), & |P| < P_I \\ \eta(P-P_R)^2/2, & P \ge P_I \end{cases} \tag{3}$$

where P_I denotes the positive inflection point at which the switch occurs, P_R is the local remanence polarization, and η is the reciprocal slope $\partial E/\partial P$. The Gibbs free energy

$$G = \psi - EP \tag{4}$$

balances this internal Helmholtz energy with the electrostatic energy; i.e., work performed by the applied external field. Direct minimization of the Gibbs energy yields the kernel

$$\overline{P}(E+E_I;x_+) = \frac{E+E_I}{\eta} + 2P_R x_+ - P_R \tag{5}$$

where $x_+ = 1$ for positively oriented dipoles and $x_+ = 0$ for negatively oriented dipoles. For implementation purposes, x_+ is set to 1 whenever $E + E_I > E_c$ and to 0 when $E + E_I < -E_c$.

It should be noted that the kernel given here is analogous to a Preisach kernel if the limit is taken as η goes to infinity. It thus provides increased physical motivation and an extra degree of freedom over the Preisach formulation. Like the Preisach kernel, it does not include thermal relaxation effects such as creep, although these can be added in a straightforward manner, see [1, 9] for more details. Whereas some applications require the increased accuracy thermal relaxation provides, others will benefit more from more frequent control updates which the faster speed of the negligible relaxation formulation provides. As detailed in [1], the formulation above runs about 2 – 4 times faster on Intel Pentium IV processors than a formulation with relaxation. FPGA implementations of the relaxation

model are still under development, but preliminary indications are that the relaxation model will be 8 – 16 times slower than the FPGA implementation described in this paper.

We also note that whereas the result is given in terms of polarization or magnetization, as shown in [1], the extension of this to strain is straightforward and does not add significant computational complexity. As such, we focus on the polarization/magnetization to simplify discussion.

COMPUTATIONAL COST AND PROCESSOR LIMITS

While the kernel (5) is relatively simple, the overall model is computationally intensive. Simplifying (2) yields

$$P(E;x_+) = \frac{E}{\eta} - P_R + 2P_R \sum_{i=0}^{N_c} \sum_{j=0}^{N_I} x_+ \nu_c(E_c[i]) \nu_I(E_I[j]) w_c[i] w_I[j] \tag{6}$$

which requires $N_c N_I$ + O(1) multiplies and additions per timestep, plus 2 $N_c N_I$ conditional statements per timestep to determine x_+. N_c and N_I can vary between roughly 30 and 100 depending on the material being modeled and accuracy requirements. Thus, a very large volume of simple mathematical operations need to be computed each second. The effect is a model that is too slow for many applications. For example, a C code implementation of the model with $N_c = N_I = 32$ was capable of computing almost 100,000 timesteps per second on a Pentium IV Xeon 3.8 GHz.

Depending on the control law being employed, a large number of timesteps per second may need to be evaluated in software to determine the correct control input. For example, [1] develops an approximate inverse model which runs in 1/10 to 1/20 of the speed of the forward model. Thus, with a resolution of 10 timesteps/cycle, one hundred thousand timesteps per second equates to a maximum cycle rate of 500 – 1000 Hz. Most embedded microprocessors will be slower than this. Thus, while a processor may be fast enough for some applications, others will require a faster alternative.

FPGA IMPLEMENTATION

To map the given homogenized energy model to an FPGA, we must determine how to parallelize the algorithm and determine the bit widths needed for the arithmetic. This must be done in the context of the board and design method we wish to target.

A. Design Hardware, Tools and Methods

Traditionally, FPGA design is performed with the hardware description languages VHDL and/or Verilog. More recently, numerous high-level tools have emerged on the market to simplify development. We chose to use the Xilinx System Generator application which allows design to be performed within Simulink and converted to VHDL or Verilog. We also chose a prototyping board from Lyrtech Signal Processing that provides the necessary board model in Simulink to compile, place, and route directly to the FPGA. The board contains two Xilinx Virtex 2 3000 FPGAs and provides a clock speed of 66 MHz. It also provides 4 Texas Instrument DSPs, although those have are not utilized in the current

design. A screenshot depicting a portion of the System Generator Design can be observed in Figure 1.

B. Integer Arithmetic Constraints

The Lyrtech board provides a 14 bit analog to digital converter. Based on this, the input field levels were set to be signed, 14 bit numbers. To accommodate this, the field E_s at which the material saturates was used to normalize the input. Additionally, polarization/magnetization was normalized so that remanence value is 1. This gives the model

$$P(E;x_+) = P_R\left(\hat{E}\hat{\eta} - 1 + 2\sum_{i=0}^{N_c}\sum_{j=0}^{N_I} x_+ \nu_c(E_c[i])\nu_I(E_I[j])w_c[i]w_I[j]\right) \tag{7}$$

where $\hat{E} = E/E_s$ and $\hat{\eta} = E_s/\eta P_R$. Only the portion within the parenthesis needs to be computed in the FPGA. This gives $\hat{E} \in (-1,1)$ while the nature of the materials ensures $\hat{\eta}$ does not grow large. Additionally, E_c and E_I can be divided by E_s a-priori, giving $\hat{E}_c = E_c/E_s \in (-1,1)$ and $\hat{E}_I = E_I/E_s \in (-1,1)$.

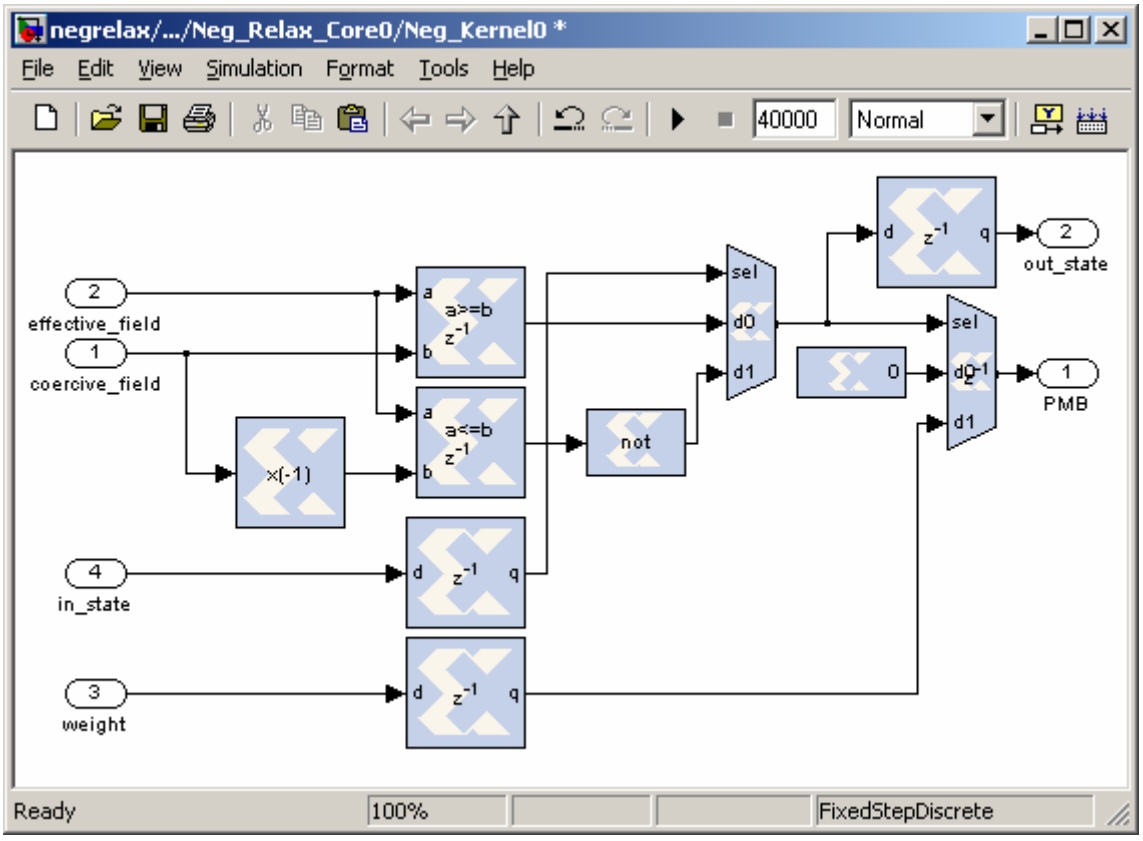

Figure 1: Screenshot of the Xilinx System Generator for DSP application, showing the kernel computation of the homogenized energy model.

Since E, E_c, and E_I are used solely for comparisons, there is no need to consider bit growth with these values. The state x_+ can be stored in memory and used to select a mux output, the result of which is 0 when x_+ is 0 and $\nu_c(E_c[i])\nu_I(E_I[j])w_c[i]w_I[j]$ when x_+ is 1. The term $\nu_c(E_c[i])\nu_I(E_I[j])w_c[i]w_I[j]$ can be computed and stored for each quadrature point in advance. These numbers will be less than one, but exactly how many bits are needed depends on the material-specific parameters. We chose to use a 22-bit unsigned fractional number to store these combined weights and density heights. Because the sum of all weights is less than or equal to one, no bit growth is needed on the summation except for one bit of growth or a saturation circuit on the final addition. The final multiply by 2 can be achieved with a bit shift.

C. Parallelization

The homogenized energy model parallelizes naturally. The kernel evaluations do not depend on one another, and thus the comparisons and mux output for each term within the sum may be computed in parallel if FPGA resources permit. Along with the necessary control logic, each Virtex II 3000 FPGA was able to contain the necessary logic to perform 64 evaluations in parallel. In fact, with this logic and the board model, only approximately 70% of the slices (62% on one chip, 74% on the other) were utilized, which leaves sufficient room to add a feedback controller.

COMPARISON OF FPGA AND FLOATING POINT C CODE RESULTS

As mentioned previously, a Pentium IV Xeon 3.8 GHz was capable of computing 100,000 timesteps per second with $N_c = N_I = 32$. Using two chips, each capable of performing 64 kernel evaluations in parallel, the FPGA implementation was capable of processing an input every 10 clock cycles (8 to cycle through the 1024 points through the FPGA, plus 2 overhead cycles due to bus limitations). That equates to 6.6 million timesteps per second, 66 times the rate obtained by the 3.8 GHz Pentium. Considering latency from input to output, the Pentium 3.8 GHz needed to 10 μs to produce the result for one timestep. The latency of the FPGA design over a factor of 20 smaller: 30 clock cycles or 450 ns.

The improved speed of the FPGA approach does come at the cost of lost accuracy; the integer arithmetic utilized in the comparisons and weights is less precise than the floating point arithmetic employed in the C code. Figure 2 shows results for an example input and parameters chosen to match the ferromagnetic material Terfenol-D, whereas Figure 3 shows similar results with parameters chosen to match the ferroelectric material PZT. The peak error is 0.68% of the remanence magnetization for the terfenol example, while the root-mean-square error is 0.066% of remanence. For the PZT case the peak error was 1.25% of the remanence polarization and the root-mean-square error was 0.05% of remanence. These error levels are typically much smaller than the modeling error present in the system, and as such no significant accuracy should be lost by the FPGA implementation for most applications.

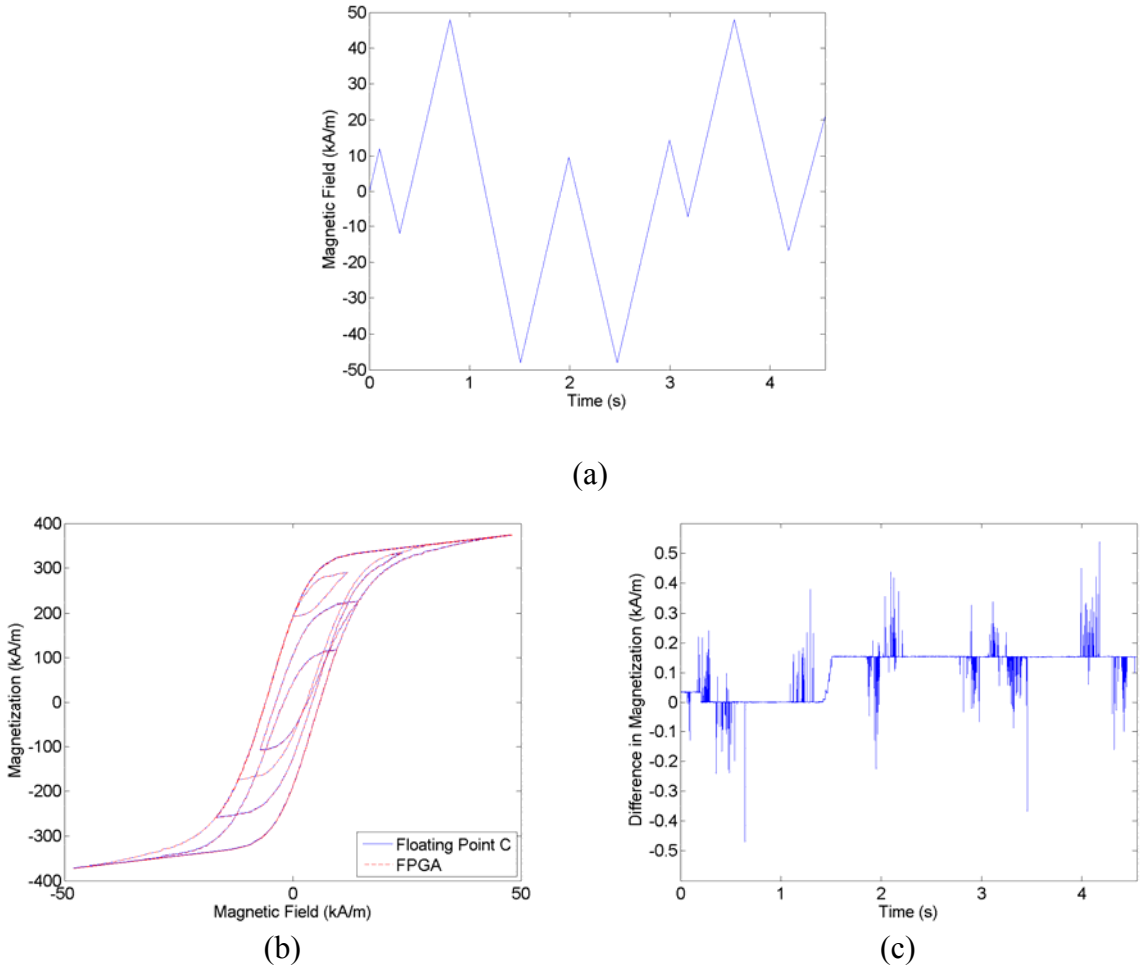

Figure 2: (a) Input Magnetic Field utilized to test the homogenized energy model. Comparisons of the FPGA and floating point C code results are given in (b), and the difference between the implementations is given in (c). Materials parameters were determined by a least squares fit to Terfenol-D data.

CONCLUDING REMARKS

We have demonstrated that FPGAs offer the potential to reduce the computation time of the homogenized energy model for ferroelectric and ferromagnetic materials by over an order of magnitude when compared to a traditional processor without losing significant accuracy. As noted previously, the same results can be expected from the Presiech model as well. It should be noted, however, that these results are somewhat specific to the FPGAs employed; a larger FPGA or an FPGA capable or being run at a higher clock speed would allow this rate to be increased, while in a smaller FPGA the rate would necessarily be slower. A different board may not have the same bus limitation as the board we are currently employing, in which would also increase the throughput by 20%, while decreasing the latency by 75 – 100

ns. In any case, the compact size and high data rates offered by an FPGA make them attractive platforms for model-based control designs.

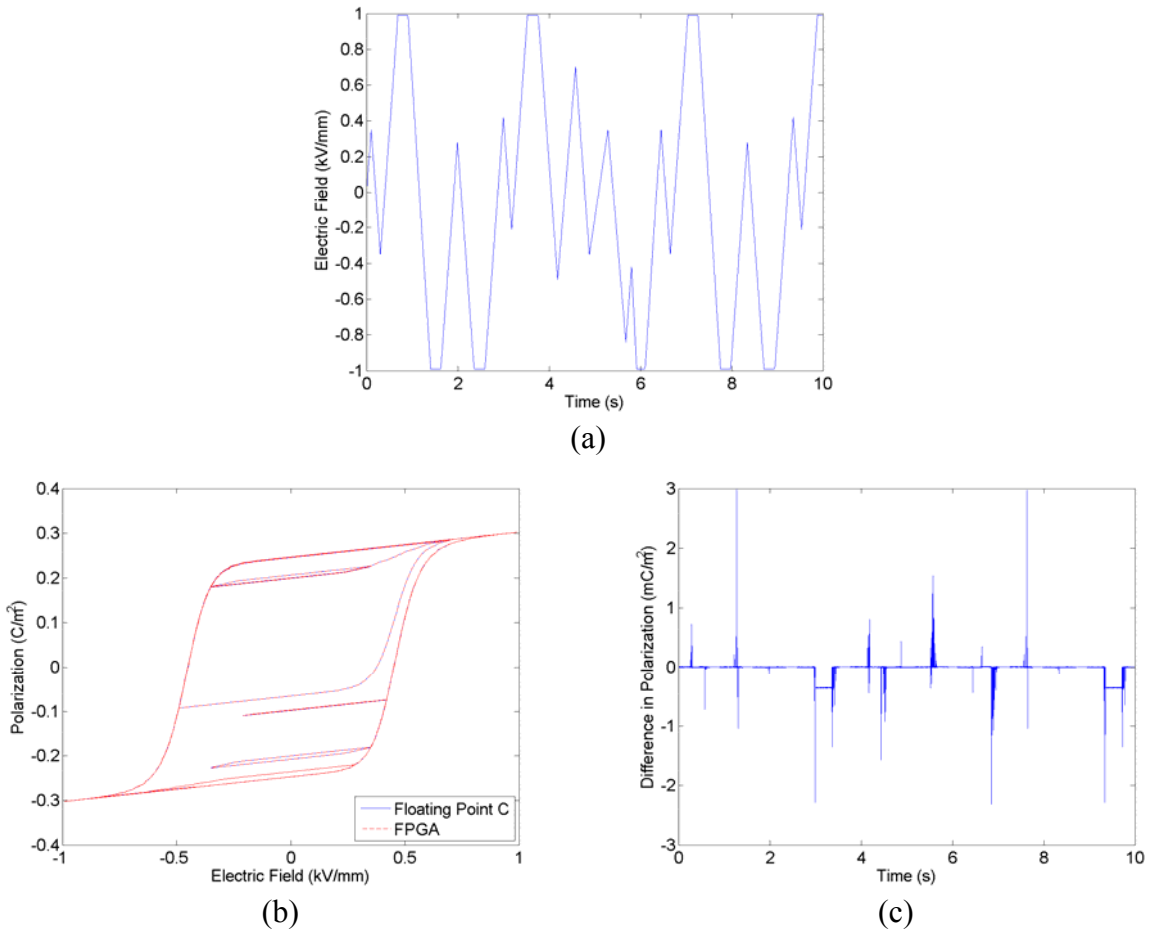

Figure 3: (a) Input Electric Field utilized to test the homogenized energy model. Comparisons of the FPGA and floating point C code results are given in (b), and the difference between the implementations is given in (c). Materials parameters were determined by a least squares fit to PZT data.

ACKNOLEDGEMENTS

The authors wish to thank the United States Air Force Office of Scientific Research, who supported this research under grant AFOSR-FA9550-04-1-0203. Thanks are also given in Etienne Fiset and Daniel Carbonneau of Lyrtech Signal Processing for their assistance in getting the hardware board up and running.

REFERENCES

1. T.R. Braun and R.C. Smith, "High-speed model implementation and inversion techniques for ferroelectric and ferromagnetic transducers," CRSC Technical Report CRSC-TR07-10. Journal of Intelligent Material Systems and Structures, to appear.
2. M.J. Dapino, R.C. Smith, L.E. Faidley, and A.B. Flatau, "A coupled structure-magnetic model for magnetostrictive transducers," Journal of Intelligent Material Systems and Structures, 11(2), pp. 134-152, 2000.
3. P. Ge and M. Jouaneh, "Modeling hysteresis in piezoceramic actuators," Precision Engineering, 17, pp. 211-221, 1995.
4. D.C. Jiles and D.L. Atherton, "Theory of ferromagnetic hysteresis", Magnetism and Magnetic Materials, 61, pp. 1265-1281, 1984.
5. J.E. Massad and R.C. Smith, "A domain wall model for hysteresis in ferroelastic materials," Journal of Intelligent Material Systems and Structures, 14(7), pp. 455-471, 2003.
6. I.D. Mayergoyz, Mathematical Models of Hysteresis, Springer-Verlag, New York, 1991.
7. S. Mittal and C-H. Meng, "Hysteresis compensation in electromagnetic actuators through Preisach model inversion," IEEE/ASME Transactions on Mechatronics, 5(4), pp. 394-409, 2000.
8. G. Robert, D. Damjanovic and N. Setter, "Preisach modeling of piezoelectric nonlinearity in ferroelectric ceramics," Journal of Applied Physics, 89(9), pp. 5067-5074, 2001.
9. R.C. Smith, *Smart Material Systems: Model Development*, SIAM, Philadelphia, 2005.
10. R.C. Smith, M.J. Dapino, T.R. Braun and A.P. Mortensen, "A Homogenized energy framework for ferromagnetic hysteresis," IEEE Transactions on Magnetics, 42(7), pp. 1474-1769, 2006.
11. R.C. Smith, M.J. Dapino and S. Seelecke, "A free energy model for hysteresis in magnetostrictive transducers," Journal of Applied Physics, 93(1), pp.458-466, 2003.
12. R.C. Smith and J.E. Massad, "A unified methodology for modeling hysteresis in ferroic materials," Proceedings of the 2001 ASME Design Engineering Technical Conferences and Computer and Information in Engineering Conference, Vol 6, Pt B, pp. 1389-1398, 2001.
13. R.C. Smith, S. Seelecke, M.J. Dapino and Z. Ounaies, "A unified framework for modeling hysteresis in ferroic materials," Journal of the Mechanics and Physics of Solids, 54(1), pp. 46-85, 2006.
14. R.C. Smith, S. Seelecke, Z. Ounaies and J. Smith, "A free energy model for hysteresis in ferroelectric materials," Journal of Intelligent Material Systems and Structures, 14(11), pp. 719-739, 2003.
15. G. Song, J. Zhao, X. Zhou and J.A. Abreu-Garci, "Tracking control of a piezoceramic actuator with hysteresis compensation using inverse Preisach model,", IEEE/ASME Transactions on Mechatronics, 10(2), pp. 198-209, 2005.
16. G. Webb, A. Kurdila and D.C. Lagoudas, "Adaptive hysteresis model for model reference control with actuator hysteresis," Journal of Guidance, Control and Dynamics, 23(3), pp. 459-465, 2000.