INVESTIGATING SIGNAL PROCESSING THEORY WITH *MATHEMATICA*

Brian L. Evans and *James H. McClellan*
School of Electrical Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0250

H. Joel Trussell
Dept. of Electrical and Computer Eng.
North Carolina State University
Raleigh, NC 27695-7911

ABSTRACT

We present extensions to the computer algebra and multimedia environment *Mathematica* to enable students to investigate signal processing theory. Students interact with the extensions at two levels. The lower level is a set of *Mathematica* programs called the Signal Processing Packages (SPP) that implement signals and systems not found in the core of *Mathematica*. The higher level is a set of interactive multimedia Notebook tutorials on analog filters, convolution, Fourier analysis, and the z-transform. Other Notebooks have been written to help students evaluate their knowledge of digital filter design and convolution.

1. INTRODUCTION

The signal processing society recently dedicated one session at ICASSP '92 (vol. 4, pp. 73-112) and the Fall 1992 issue of *The Signal Processing Magazine* to signal processing education. Those papers focus on four areas in which educators are using computers to supplement signal processing education in the laboratory and classroom:

DSP boards with software for downloading programs,

block-diagram prototyping environments such as Ptolemy,

computer-aided instruction based on HyperCard, and

interpretive computing environments such as Matlab.

This paper presents extensions to *Mathematica* [1], a combined computer-aided instruction and interpretive computing system, to create an environment in which students can investigate signal processing theory.

2. MATHEMATICA

We have chosen to start with *Mathematica* because it is a computer algebra (CA) environment [1] with a multimedia interface [2]. The CA abilities are necessary for students to explore the mathematics underlying signal processing theory (an approach complementary to a numeric processing environment like Matlab). We have programmed *Mathematica* to include knowledge about signal processing.

The Notebook interface to *Mathematica* can present material through formatted text, equations, and tables as well as sound, graphs, and animations. Tutorials written in the Notebook interface possess an added dimension because the student can interact with an equation first by modifying parameters and then by observing a graph, sound, or animation that represents the change. Students can use the Notebook interface to obtain and document solutions, write laboratory reports, and so forth.

3. SIGNAL PROCESSING PACKAGES

Mathematica already possesses some signal processing capabilities. It supports complex numbers, matrices, vectors, and polynomials which are representations of signals. It provides the 1-D and 2-D DFT as well as trigonometric, Bessel, and exponential functions. *Mathematica* 2.0 implements 1-D causal Laplace and Fourier transforms and provides continuous step and Dirac delta functions.

The research conducted by Mr. Evans and Dr. McClellan was supported in part by Joint Services Electronics Program contract #DAAL-03-90-C-0004.

To complete *Mathematica*'s signal processing capabilities, we have written a set of *Mathematica* programs called the Signal Processing Packages (SPP) [3, 4, 5]. These packages implement

1. new functions such as sinc, Dirichlet, pulse, step, and impulse functions,
2. new operators such as shifters and filters,
3. new graphical representations such as sequence and signal plots, pole-zero diagrams, and magnitude/phase plots for 1-D and 2-D signals,
4. 1-D symbolic discrete and piecewise continuous convolution, and
5. 1-D and m-D symbolic multi-sided linear transforms (Laplace, Fourier, z, DTFT, and DFT).

The SPP are currently being used in undergraduate and graduate signals and systems curricula at several colleges and universities [6].

At Georgia Tech, *Mathematica* is introduced in the first signals and systems course. In such a course, students would first learn how to represent signals and systems as mathematical objects and then as *Mathematica* expressions. Once that hurdle has been overcome, students would be asked to graph the effects of scaling and reversing the time axis in the continuous and discrete domains. Next, using the convolution routines in the SPP, the students would work discrete and continuous convolution problems *symbolically* and then plot the answers or convert them to numbers. The convolution routines can generate an animation sequence illustrating the "flip-and-slide" approach to any convolution problem.

In a second course in signals and systems at Georgia Tech, students would solve linear transform problems with the SPP. The SPP can show intermediate calculations, so the students can learn how to perform the transform by hand. The transform routines are flexible in that new transform pairs can be specified. For example, students might be asked to find the Fourier response of a communications system for an input signal $x(t)$ with a given spectral shape $X(\omega)$. Using the SPP, students would first define $X(\omega)$ in terms of *Mathematica* code and then call the Fourier transform with $x(t) \longleftrightarrow X(\omega)$.

In other classes, students could use the SPP to derive transfer functions of linear shift-invariant (LSI) systems by specifying an "abstract" transform pair for each input signal, such as $x[n] \longleftrightarrow X(z)$ in the case of the z-transform. For discrete linear time-varying systems, students could generate input-output relationships (an LSI transfer function plus aliasing terms). Because the z- and Laplace transforms return the region of convergence, the SPP can determine the conditions on free parameters to ensure stability.

4. INTERACTIVE TUTORIALS

Using the SPP, *Mathematica* Notebooks can be written to provide examples of algebraic forms of signal processing expressions for the students to evaluate. We have written one set of Notebooks to serve as on-line help and another set to form a user's manual. By adding text, equations, and graphics, Notebooks can take the form of interactive tutorials. Our interactive tutorials arrange information in a hierarchy of sections, subsections, and so forth. When opening one of them, a student would see the table of contents. Students can select a topic of interest for further study, or they can skim the Notebook sequentially to get an overview. Students also have the option of searching for keywords (highlighted by the author) and patterns (chosen by the student). Each tutorial Notebook demonstrates at least one difficult concept through animation. Since the code to generate each animation is provided, students can create their own animations.

The tutorial Notebooks [5, 6, 7, 8] are:

AnalogFilters shows how to design Bessel and classical analog filters: Butterworth, Chebyshev, and Elliptic. Three animations showing how the magnitude response depends on filter parameters, e.g. Butterworth filter response vs. order.

DTFT teaches the discrete-time Fourier transform (DTFT) by relating it to the discrete Fourier transform (DFT), Fourier series, and the Fourier transform. The only animation shows the increasing accuracy in the Fourier series approximation as more terms are added.

PiecewiseConvolution discusses both discrete and piecewise continuous convolution and illustrates the mechanics of the "flip-and-slide" approach to convolution by means of animation.

zTransformI defines the bilateral z-transform and the region of convergence (ROC) which is necessary for uniqueness; discussing the ROC leads naturally into the subject of stability and causality of a signal based on the location of its poles relative to the ROC.

zTransformII demonstrates the relationship between the z-transform and the DTFT. Several animations showing the DTFT of a system as a phasor rotating around the unit circle of the z-transform of that system, a basic idea behind digital filter design.

zTransformIII shows how to choose poles and zeroes to construct digital filters. Two animations illustrate the effect on the magnitude response of a fourth-order IIR elliptic filter when two of its poles are slightly perturbed (see Figure 1).

Each tutorial contains enough formatted text and equations to introduce the subject. Their purpose, however, is not to duplicate material already found in textbooks but instead to enable students to interact with the subject material. Our ultimate goal is to write tutorial Notebooks on a comprehensive range of signal processing topics.

5. TESTING FUNDAMENTALS

In a standard text, students have access to a limited number of example and homework problems. When students run out of example problems, they can sometimes generate their own in *Mathematica*, as in the case of convolution problems. Professors can write their own example problems in Notebook form for the students to study or complete.

In the latter case, the instructor provides a framework which the student completes. The instructor can pose questions that can be algebraically or numerically intensive. The student answers the questions with graphs, text, and *Mathematica* code. Once completed, the student electronically mails the ASCII Notebook to the instructor for checking, as is done in some calculus courses [9].

At NC State University, EE students are asked to complete two exercise Notebooks. The Notebooks do not rely on any external packages, guide the student through the exercise, and help students evaluate their knowledge of the topic. They are:

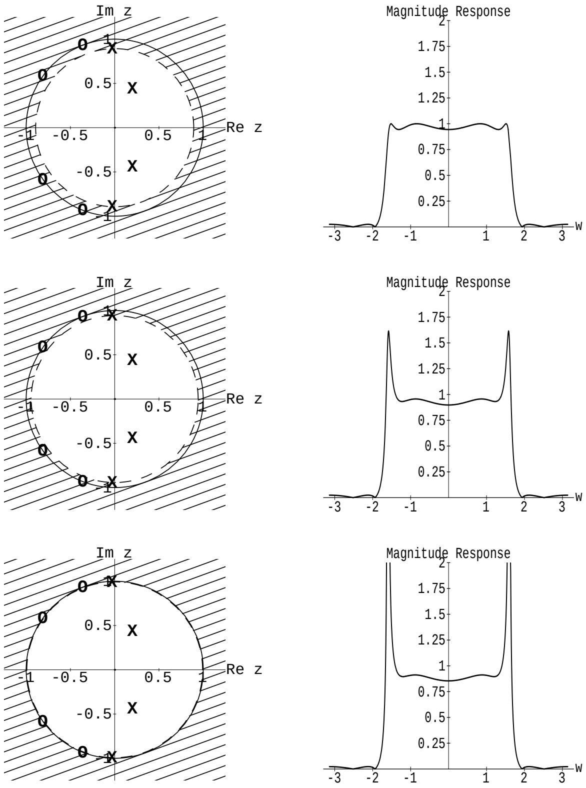

Figure 1: **Selected frames in an animation showing how a perturbation in the radius of the outer pole pair corrupts the magnitude response near the band edge**

makeconv asks the student to sketch a convolution kernel with lowpass characteristics which the Notebook checks. Question is repeated for a highpass kernel. Students can convolve various signals with their own kernels.

polezero asks the student to design four different digital filters by pole-zero placement (a sixth-order all-pole bandpass, an all-pole low pass, a better low pass, and a notch filter). Students evaluate their choices by inspecting the magnitude response. At any time, students can have *Mathematica* check the pole-zero placement.

6. A SURVEY OF OTHER EFFORTS

As a computer algebra and multimedia environment, *Mathematica* has helped students learn calculus [9], physics [10], robotics [11], and mechanical drawing [12]. EE students have explored Winograd FFT algorithms [13], visualized electromagnetics [14, 15], and analyzed laboratory data [16].

7. CONCLUSION

We have presented extensions to *Mathematica* to enable students to investigate signal processing theory at two different levels. The lower level is evaluating algebraic signal processing expressions using the new signals and systems defined by the SPP. At this level, students can create and solve their own example signal processing problems. For convolution and transform problems, students can also see how to find the solutions. The higher level is a set of interactive multimedia Notebook tutorials on several signal processing topics. The tutorial Notebooks give many examples in terms of *Mathematica* code for the students to evaluate and change.

The new environment is flexible. When students explore beyond the limitation of the environment, they add knowledge either by writing new routines or by overriding default options to existing ones (e.g., specifying new transform pairs).

Independent of the SPP, we have written two Notebooks to test a student's knowledge of digital filter design and convolution. The Notebook interface provides the ability to interact with rendered graphics, and *Mathematica* code tests the graphical and numerical answers obtained by the student.

Standard textbooks are limited as to the number of example problems, homework problems, and graphs that it can present. Furthermore, students cannot interact with the formulas discussed in the text. With the new system, students can generate their own example problems complete with solutions, interact with information presented in tutorial form, and evaluate their knowledge.

8. FTP INFORMATION

The SPP and Notebooks are available via anonymous FTP to `gauss.eedsp.gatech.edu` (IP #130.207.226.24). The `Mathematica` directory contains archives for three different operating systems: `SigProc2.0.mac.sea.hqx` for the Macintosh, `SigProc2_IBM_PC.zip` for DOS machines, and `SigProc2.0.tar.Z` for Unix machines. Direct questions and comments to `evans@eedsp.gatech.edu`. The file `OtherNotebooks.tar.Z` contains the Notebooks of Section 5, a Notebook on sampling theory by R. Bamberger of WSU, Notebooks on windows by J. Smith of Stanford, and a Notebook on cochlear signal processing by M. Slaney [2].

9. REFERENCES

[1] S. Wolfram, *Mathematica: A System for Doing Mathematics by Computer.* Redwood City, CA: Addison-Wesley, second ed., 1991.

[2] M. Slaney, "Interactive signal processing documents," *IEEE ASSP Magazine*, vol. 7, pp. 8–20, Apr. 1990.

[3] B. L. Evans, J. H. McClellan, and W. B. McClure, "Symbolic z-transforms using DSP knowledge bases," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, (Albuquerque, NM), pp. 1775–1778, Apr. 1990.

[4] B. L. Evans, J. H. McClellan, and W. B. McClure, "Symbolic transforms with applications to signal processing," *The Mathematica Journal*, vol. 1, no. 2, pp. 70–80, Fall, 1990.

[5] B. L. Evans and J. H. McClellan, "Symbolic analysis of signals and systems," in *Symbolic and Knowledge-Based Signal Processing* (A. Oppenheim and H. Nawab, eds.), Prentice-Hall, 1992.

[6] B. L. Evans, L. J. Karam, K. A. West, and J. H. McClellan, "Learning signals and systems with Mathematica," *IEEE Trans. on Educ.*, Feb. 1993.

[7] B. L. Evans, J. H. McClellan, and K. A. West, "Mathematica as an educational tool for signal processing," in *Proc. IEEE Southeastern Conf.*, (Williamsburg, VA), pp. 1175–1179, Apr. 1991.

[8] K. West and J. McClellan, "Symbolic convolution," *IEEE Trans. on Educ.*, 1993. To Be Published.

[9] D. Brown, H. Porta, and J. Uhl, "Calculus and Mathematica: Courseware for the 90's," *The Mathematica Journal*, vol. 1, pp. 43–50, Summer 1990.

[10] C. Taylor, "Mathematica in the classroom," *Computers in Physics*, vol. 5, no. 1, pp. 16–21, 1991.

[11] M. C. Leu, Z. Ji, and Y. S. Wang, "Studying robot kinematics and dynamics with the aid of Mathematica," *Int. Journal on Mechanical Eng. Education*, vol. 19, pp. 213–228, July 1991.

[12] J. Fine and L. Waite, "Spatial problem solving in an integrated first-year curriculum," *The Mathematica Journal*, vol. 1, no. 4, pp. 62–67, 1991.

[13] G. E. Sobelman, "Computer algebra and fast algorithms," *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, vol. 4, pp. 89–92, Mar. 1992.

[14] R. Cole, D. Krull, and M. Sweitzer, "Visual electromagnetics using Mathematica and HyperCard," *IEEE Antennas and Propagation Soc. Int. Sym.*, vol. 4, pp. 1671–1674, 1990.

[15] J. Schneider, "Electromagnetics with Mathematica," *IEEE Trans. on Educ.*, Feb. 1993.

[16] H. Treat, "Using LabVIEW and Mathematica in combination: data collection to graphical presentation in a single setting," *Computers in Education Division of ASEE*, vol. 1, no. 2, pp. 19–23, 1991.