# EXPERIENCES IN TEACHING DSP FIRST IN THE ECE CURRICULUM

*James H. McClellan*[1] *Ronald W. Schafer*[1] *Mark A. Yoder*[2]

[1] School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332–0250 USA
[2] School of Electrical and Computer Engineering, Rose-Hulman Institute of Technology, Terre Haute, IN 47803 USA

## ABSTRACT

In this paper we describe experiences gained from teaching an introductory electrical engineering course based on *digital* signal processing rather than the traditional first course in analog circuit theory. We will discuss our motivation for teaching DSP first, before covering analog circuits and systems. We will describe the style of the course and point out difficulties, as well as advantages, in this organization of basic material. At Georgia Tech and Rose-Hulman, this beginning course has been required of all computer engineering majors. Finally, we will make some comments about extending this approach to encompass a wider range of students from other disciplines.

## 1. INTRODUCTION

An introductory electrical engineering course based on Digital Signal Processing (DSP) has been taught at Georgia Tech since 1993, and also at Rose-Hulman for the past two years. This course uses digital filters and simplified frequency domain descriptions to convey the important ideas of filtering, systems, frequency response and $z$-transforms. Since the course is based on *digital* signal processing, it has been quite easy to incorporate computer-based lab experiences into the flow of the material. Sound synthesis has been our primary vehicle for illustrating frequency content, and has even allowed us to introduce the idea of time-frequency analysis via the spectrogram. In addition, two of the labs treat image processing, where low-order FIR filters are used for blurring and sharpening, as well as image zooming via interpolation. Although we have not undertaken a formal evaluation of student performance and learning, it is clear that computer engineering students react favorably to this introduction. Many computer-oriented students discover an unexpected link between computers and the mathematics of signal theory, and have pursued further study in traditional DSP courses at the senior and graduate level.

## 2. DSP FIRST

The idea of using signal processing as the first course in electrical engineering is not new. Some have proposed teaching analog signal processing prior to circuit theory, so that a system viewpoint is presented prior to the details of circuit implementation. Our approach has been to start with *digital* signal processing, and then move into analog systems. This matches quite well with the everyday experiences of most of our students who have expertise with software packages that might contain DSP capabilities (e.g., image enhancement in a program such as Adobe Photoshop).

Even the DSP first approach is derived from other efforts, most notably Prof. Ken Steiglitz [1] at Princeton. His initial effort was ahead of its time, because the software to support actual processing was neither widely available nor easy to use in the 1970's. More recently, Steiglitz has written another book that presents DSP material to students with a computer science background, but with a keen interest in digital audio and computer music [2]. Both of these books have had a strong influence on our definition of this new introductory course.

### 2.1. Rationale

The rationale for using DSP as the first course is straightforward—computers are prevalent and easy to work with. Therefore, a first course can draw on many examples from non-trivial processing systems and motivate students to understand signal processing techniques that they have already used. Laboratory exercises require no special prerequisites beyond programming skills that are gained in an introductory CS course. Furthermore, the available computer software for implementing digital filters is extremely powerful. Sophisticated mathematical programs such as MATLAB [5] and *Mathematica* [6], permit students to program a moderately complex DSP system as a laboratory exercise in this first course.

### 2.2. Levels of Abstraction

Another factor in system design is what might be termed "levels of abstraction." When designing and building a particular system, it is possible to approach the design at many different levels as shown in Fig. 1. In present-day education there is the need to give students a view of these different levels of abstraction so they appreciate how a problem can be attacked in different ways.

A notable disadvantage of circuit theory is that it is fairly low on the ladder of abstractions because the circuit equations relate to a particular implementation with $R$, $L$ and $C$. Other higher level abstractions must be learned later in the educational process to appreciate how circuits are just one way to synthesize a frequency response, for example. In a traditional ECE curriculum, the higher levels of abstraction are generally postponed until the senior year. There is some basis in learning theory for this bottom-up approach where the student moves from specific cases to see the gen-

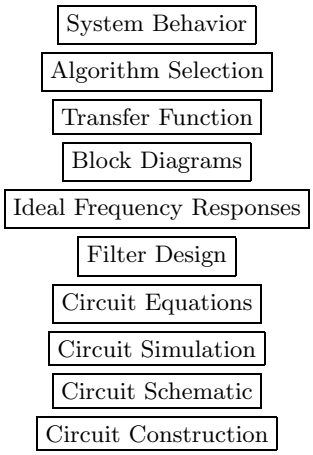


**Figure 1. Levels of Abstraction: Top-Down.**

eral picture. There is probably also a practical reason for limiting the scope of abstractions: difficulty in presenting multiple levels simultaneously.

However, the alternative approach is the top-down method in which the general ideas are presented first and then filled in with detailed examples. Since our aim has been to illustrate more of these abstraction levels in the first ECE course, we are trying to use more of the top-down model. This is feasible because we have powerful software tools for implementation, so we can delve into the lower levels quickly (as part of the lab). At the same time, we can maintain a higher-level viewpoint in the lecture presentations. This ability to float between levels of abstraction is a feature that needs further refinement as we serve a wider cross section of students.

The labs fit well with the multi-level abstraction hierarchy, because they encourage students to think about the theoretical material at two levels: equations that define the processing and computer programs that implement the processing. MATLAB also supports this easy movement between abstractions. The development of a complete processing system may take only a few hours once the system is described in terms of its frequency response and system block diagram. This top-down approach makes it possible to use DSP to show theory, applications and implementations together in an introductory course.

### 2.3. DSP Before Circuits

This introductory course represents a major departure from the traditional EE curriculum where circuit theory has long been the first course. The reasons for starting with circuit theory are: (1) it is a good subject for introducing the use of mathematics in representing and analyzing physical systems and (2) the knowledge gained is useful for most of the subsequent classical EE subjects. Since these points remain true, why change?

One reason is that basic circuit theory focuses on a very low level of engineering systems, and therefore it is difficult to motivate students who ask, "Where can I apply this subject?" On the other hand, introductory discrete-time signals and systems is easy to motivate *if it goes beyond the mathematics of DSP;* e.g., by processing audio and video signals. Furthermore, in this era of digital systems, a basic DSP course is more naturally tailored to the widespread use of computers for instrumentation and processing.

In the coming years, there is likely to be a general recognition that signal processing methods are more important than circuit theory for a wide audience of engineering students outside of ECE. For example, mechanical engineers find significant use for signal processing in measurement systems, and computer scientists involved in multi-media applications can benefit greatly from a deeper understanding of discrete-time signals and systems. Thus, an "introductory DSP" course would be attractive to other majors because the material would not be tied to a long prerequisite chain as is now the case. Instead, second or third year students could have a glimpse of computer processing methods that might include applications from their own area.

### 2.4. Essential Ideas in a First Course

If one examines circuit theory for its essential ideas, it is relatively easy to identify concepts that have a broad impact. But it is just as easy to find topics that only relate to circuit implementation in a narrow sense. It is also possible to identify activities in the course that can only be considered skill building for problem solving. The broad based topics include phasors, complex exponential signals, frequency response, input-output characterization of a system, Kirchoff's Laws, transforms. The narrower focus is found in such topics as special circuits and transformation of circuits. Skills include such things as reducing complicated circuit expressions, solving differential equations for transients, complex number manipulation, etc.

All of these circuit theory activities are important in their own context, but they need not be first in the EE curriculum. Some ideas must be planted in the first course, but it is just as easy to present ideas such as phasors or frequency response by using digital filters, as it is by using a circuit as the example system. The essential concepts must be covered in a basic course, but the underlying implementation can vary.

### 2.5. Fitting DSP into a Crowded Curriculum

The suggestion of adding "yet another course" to the already overcrowded curriculum is not made lightly. First, Georgia Tech and then Rose-Hulman have both found that DSP fits in the sophomore year of the computer engineering curriculum as a required course. If DSP is to be presented first in an EE curriculum, the introductory circuits and signals & systems courses that follow can be modified to make room. Doing DSP first means the students have already had a solid introduction to sinusoids, phasors, frequency response, and spectrum. These topics don't need to be introduced in the courses that follow.

A natural place to put DSP first is in Computer Engineering (CmpE). Often the CmpE curricula are little more than EE curricula with some computers squeezed in. Many schools are now revising their CmpE curricula to better meet the needs of their students. During such a revision at Rose-Hulman the question was raised, "What makes CmpE different from EE?" It was decided that the CmpE's should have a greater exposure to DSP, and thus the curriculum was built to present DSP first.

## 3. COURSE DESCRIPTION

In order for this paper to be self-contained, we include an outline of the topics presented in the lectures and in the laboratory experiments.

### 3.1. Course Overview

The objective of this course is to show students the role that mathematical system theory can play in the development of computer applications/products such as audio and video signal processing systems. The course outline given below has been used for the past two years:

1. Definition of a signal and a system
2. Intro to MATLAB: assumes programming experience
3. Review vector/matrix notation
4. Complex numbers: represent sinusoids with phasors
5. Sinusoidal signals: amplitude, phase and frequency
6. Synthesizing sounds with general classes of sinusoids
7. Frequency content: harmonics, AM and FM signals
8. Sampling, aliasing and reconstruction
9. Linear filtering: the concept of smoothing data
10. Block level description of systems
11. FIR filtering: a generalization of running average filters
12. Recursive filtering: difference equations with feedback
13. Frequency response of FIR and IIR filters
14. Simulation of dynamic time response; impulse response
15. Z-transform analysis: rational transfer functions; omits the inverse transform

Since Georgia Tech and Rose-Hulman are both on a quarter schedule, the topic list would need to be expanded for a semester-length course. For a fifteen-week semester course, the following topics might be appropriate:

1. Synthesizing sounds with narrowband recursive filters
2. Fourier Spectrum analysis: spectrograms & windowing
3. Inverse Z-transform

### 3.2. The Computer Laboratory

While the format of the course still includes traditional lectures, a key part of the course is the laboratory experience. The laboratory demands that students be actively involved in learning how to take the theory and create working implementations in the form of computer programs. Many useful ideas for labs are appearing in new books dedicated entirely to DSP computer-based labs [3, 4]. For example, sinusoidal synthesis of music brings up the issue of choosing the sampling rate correctly for D/A conversion—a trivial matter to some, but quite confusing at first. In the area of image processing, different types of enhancement filters can be used to modify pictures for bizarre special effects. The mathematical software packages [5] that are now available for signal analysis allow for quick programming of all these different processing methods, so the lab exercises encourage the students to explore a variety of signal enhancement systems.

### 3.3. Laboratory Topics

The following list gives describes the type of labs we are using to promote the connection to real signals.

- Familiarity with MATLAB: introduced as needed by providing "warm up" exercises in the appropriate labs.
- Basic sinusoids: generating and plotting $A\cos(\pi ft+\phi)$ and playing a tone out through the D/A system and speakers. Extracting $A$, $f$ and $\phi$ from a waveform plot.
- Complex numbers and phasors: plotting vectors to represent complex phasors; the phasor addition theorem.
- Synthesis of musical tones: use sinusoids to synthesize a musical piece, (e.g., Beethoven's Fifth, Fur Elise, etc.)
- Chirps and FM synthesis: generate and listen to FM chirps; demonstrate aliasing. Use specific FM profiles to synthesize instrument sounds such as a clarinet, or a bell.
- Adding harmonics (i.e., Fourier series): Create square and triangular waves from a few harmonic sinusoids; create a vowel from formants and listen to its sound.
- Touch-tone phone: identifying multiple sinusoids.
- FIR filters (low-pass and high-pass filters): implementation of running average and first difference filters.
- Listening experiment: compare low-pass and high-pass filtering of a speech utterance.
- Frequency response: computing and plotting the frequency response of an FIR filter.
- Image filtering (blurring and sharpening): apply FIR filters to rows and columns of image. Use high-pass filter (first difference) plus threshold to find edges. Assess visual quality versus low and high frequency content.
- Image sampling and aliasing: sample a Fresnel zone plate and explain the aliasing patterns.
- Generating narrowband signals: use 2nd-order resonator to generate a sinusoid. Study relationship between pole location and the synthesized waveform.

## 4. COURSE MATERIALS

The WWW demonstrations, the lab resources, MATLAB drill programs and solved problems have been packaged together on a CD-ROM that will accompany a textbook for the course published by Prentice-Hall [8]. The outline of the book is given in the web-page screen shown in Fig. 2. It follows the course outline presented in Section 2, but also includes a chapter Fourier/spectrum analysis. The book is self-contained, but will be relatively short since a bulk of the material is on the companion CD-ROM.

### 4.1. CD-ROM Demonstrations

The in-class demonstrations are an effective means of presenting an overview of a particular topic. When these demonstrations are presented in class, the student can only influence the demonstration through the professor. Putting the demonstrations on the WWW is one way of facilitating increased student awareness [7]. All of the classroom demonstrations have been encapsulated for presentation on the WWW, so that students can have access to them outside

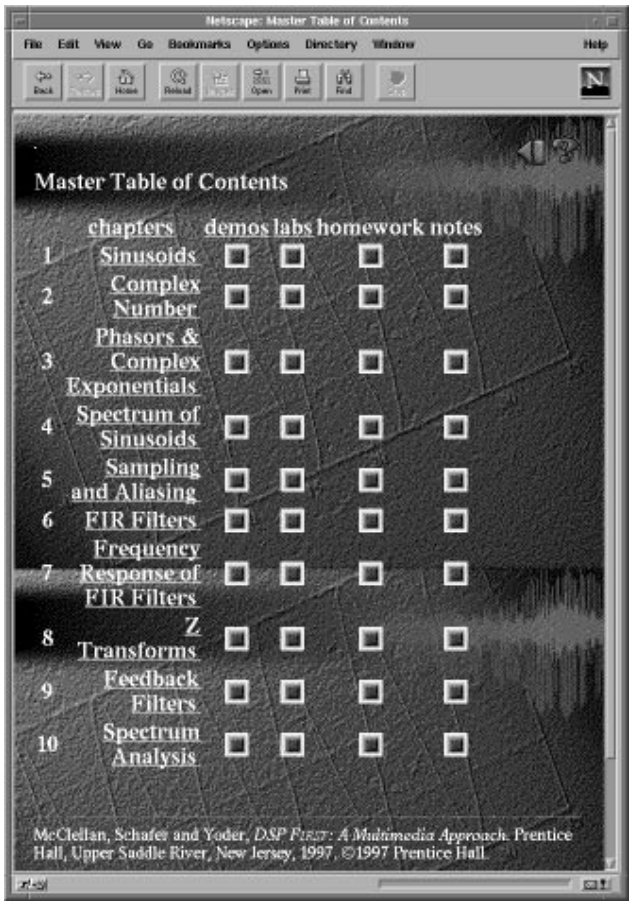


**Figure 2. Web page for table of contents**

the classroom. In addition, this material has been packaged in HTML format onto a CD-ROM equipped with a local Web browser. Below is a brief description of a few of the demonstrations and drill problems used.

- Tuning Forks: Quicktime™ movies showing various tuning forks and their sinusoidal sounds.
- MATLAB drill programs for reading sinusoidal plots and doing complex arithmetic.
- Rotating Phasors: several Quicktime™ movies of rotating phasors in the complex plane.
- Sampling: Quicktime™ movies of rotating disks "sampled" by a strobe light to illustrate aliasing.
- **PEZ**: a MATLAB GUI that allows pole-zero placement, movement and editing, along with the corresponding frequency response and impulse response.
- Spectrogram: Quicktime™ movies to show how a sliding window FFT computes the spectrogram image.

## 5. EXTENSIONS

In this final section, we will make some comments about extending this approach "up and down." By this we mean, "up" to impact the style of graduate courses, and "down" to encompass a wider range of students from other disciplines (e.g., computer science, other engineering majors, and science majors).

After 2–3 years of developing this course, it has become apparent that there is a well defined need outside of ECE where this type of course should be taught. The essential algorithms of signal processing are now commonly embedded in software, so many engineers/scientists are using DSP whether or not they are conscious of it. However, with a limited understanding of DSP, many confusing situations arise; e.g., the aliasing property of sampling or the windowing effects in an FFT. Therefore, a widely taught course in topics related to computerized instrumentation or data analysis seems timely. Such a course would obviously have a strong computer component to illustrate the DSP methods.

Likewise, the sophomore-level course that we have developed is already having a positive impact on the style of our graduate courses. The demos and MATLAB programs provide even greater insight when used by students with a mature understanding of DSP. The fact that the same demos can be used illustrates how the different levels of abstraction can be appreciated continually by students at different points in their learning.

## 6. CONCLUSIONS

The motivation behind our new sophomore-level course on discrete-time signals and systems was the desire to create an introductory electrical/computer engineering course that would couple modern computer technology into traditional EE topics such as phasors and Fourier analysis. Our experience with this course shows that students can understand relatively complicated material even during an introductory course, as long as sufficient computer tools exist to support that application.

## REFERENCES

[1] K. Steiglitz, *An Introduction to Discrete Systems*, John Wiley and Sons, New York, 1974.

[2] K. Steiglitz, *A DSP Primer: with Applications to Digital Audio and Computer Music,* Addison-Wesley, 1996.

[3] C. S. Burrus, J. H. McClellan, A. V. Oppenheim, T. W. Parks, R. W. Schafer, and H. W. Schussler, *Computer-Based Exercises for Signal Processing using* MATLAB, Prentice-Hall, Englewood Cliffs, NJ, 1994.

[4] V. Stonick and K. Bradley, *Labs for Signals and Systems using* MATLAB*,* PWS Publishing Co., 1996.

[5] *The Mathworks Inc.,* MATLAB *User's Guide*, Natick, MA, 1985–1995.

[6] S. Wolfram, *Mathematica: A System for Doing Mathematics by Computer*, Addison-Wesley, 1988.

[7] J. B. Schodorf, M. A. Yoder, J. H. McClellan, and R. W. Schafer, "Using Multi-Media to Teach the Theory of Digital Multi-Media Signals", *IEEE Trans. on Education,* Aug. 1995.

[8] J. H. McClellan, R. W. Schafer, and M. A. Yoder, *DSP First: A Multimedia Approach,* Prentice-Hall, 1997. (URL is `www.prenhall.com/~dspfirst`.)