USING J-DSP TO INTRODUCE COMMUNICATIONS AND MULTIMEDIA TECHNOLOGIES TO HIGH SCHOOLS

Andreas Spanias, Thrassos Thrasyvoulou, Yu Song, Constantinos Panayiotou[1]

Abstract *- This paper describes interactive Java software and modular web content developed at Arizona State University aimed at introducing elements of digital signal processing (DSP), multimedia, and communications technologies to high school (HS) students. The effort is motivated by several needs and trends that include: initiatives for graduating technology-aware HS students, the need to attract HS students from diverse backgrounds to engineering programs, emphasis on mathematics through applications that appeal to high school students, etc. The software used to support this effort is based on an NSF-funded object-oriented simulation environment called Java-DSP (J-DSP). J-DSP was developed from the ground up at ASU and enables simulations and DSP demonstrations over the internet. In addition, exercises and demonstrations can be embedded seamlessly in HTML educational modules. The web modules developed at ASU consist of HS-friendly lessons that contain J-DSP based examples connecting elements of music, telephony, and Hi-Fi entertainment to math and DSP. Efforts are underway to disseminate these materials to Phoenix high schools and assess their impact.*

Index Terms - DSP, High-school K12 education, math education, teaching emerging technologies in high schools.

INTRODUCTION

In the last fifteen years we have witnessed the emergence of engineering technologies that have had a profound impact in telecommunications, entertainment, computing, aerospace, medicine, and in several other areas that impact everyday life. There have been several nationwide efforts to introduce advanced information technologies in high school (HS) curricula and to college freshmen, e.g., the infinity project and DSP First [1,2]. Many of these efforts focus on relating HS mathematics to engineering technologies embedded in communications and multimedia applications. DSP is an enabling technology for explaining several applications that HS students find exciting such as cellular phones [3]-[5], MP3 [6,7], surround sound, and electronic cameras. DSP algorithms in these applications can be associated with concepts taught in HS math. For example, sinusoidal functions are used in transformations embedded in MP3 players, surround sound systems, and JPEG. Additionally, difference equations implement filters in audio graphic equalizers and cellular telephony handsets. We propose using Java-DSP [8]-[10] (http://jdsp.asu.edu) to help HS students to visualize how concepts from HS mathematics associate with DSP. J-DSP is a user-friendly web-based simulation environment developed at Arizona State University (ASU) to enable on-line labs in introductory DSP courses. To adopt JDSP for use in HS environments, we recently developed a series of J-DSP functions that are HS friendly. These functions include tone-generators (sinusoids) that have a graphical user interface (GUI) similar to that of a piano keyboard. We also coded digital filter functions with a GUI similar to that of sound equalizers in stereo systems. In addition, we introduced a series of J-DSP functions implementing popular sound effects, e.g., echo, reverberation, voice pitch modification, etc. Together with these functions, we included specific tutorial modules that associate these exciting applications with key math concepts accessible to HS seniors. Reading the tutorial modules and using the drag-and-drop J-DSP environment enables HS students to put math concepts to work. Ultimately students may use J-DSP to learn some elements of the algorithms used in cell phones and MP3 compression, voice synthesis, and JPEG. An effort is underway to train HS instructors in the greater Phoenix area to use the material and software developed.

J-DSP ENVIRONMENT

Java-DSP is a novel internet simulation concept developed at Arizona State University (ASU) by Prof. Andreas Spanias. This simulation environment is freely accessible from the internet through the use of a Java enabled browser and as such it is platform-independent and can be readily accessed by distance learners. J-DSP is implemented as a Java applet that provides an object-oriented work environment where all simulations can be visually established by forming a block diagram using a drag-and-drop process. The accessibility and simplicity of this simulation environment make it popular for use in engineering courses to facilitate on-line laboratories. In fact, the program is being used in three undergraduate courses at ASU where detailed assessment results have validated the utility and contributed to the improvement of this software. The J-DSP concept has been funded by NSF CCLI and has received an award from the IEEE Phoenix section.

The J-DSP simulation shown in Figure 1 contains a MIDI generator, two plot blocks and a Fast Fourier Transform block. While establishing these blocks and their interconnections, the students invoke the underlying

[1] All authors are with the MIDL, Department of Electrical Engineering, Arizona State University, Tempe, AZ 85287-7206. spanias@asu.edu, thrassos@asu.edu, yu.song@asu.edu, costasp@asu.edu

software and activate the simulation. By double clicking on the MIDI block, a dialog window that resembles a piano keyboard appears. Every key is associated with a tone and when pressed, generates a sinusoid which can be viewed and analyzed as a waveform. The sinusoid can also be played back using the sound player block. The simulation environment in Figure 1 can be used to introduce HS students to the notion of a sinusoidal signal and its spectrum.

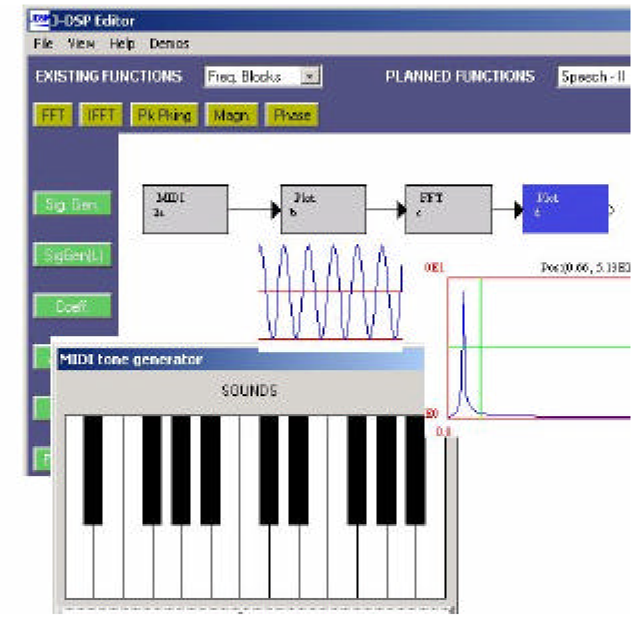

FIGURE 1
J-DSP SIMULATION ENVIRONMENT WITH MIDI GENERATOR

PEDAGOGICAL FEATURES

The J-DSP HS functions developed have been chosen carefully to facilitate hands-on-learning by referring to popular audio applications. A MIDI generator block was developed to help the students associate musical tones with sinusoidal functions. An FFT function along with a simple qualitative lecture on the spectrum provides the student with the basics of the frequency spectrum. Tonal components of a musical score are associated with spectral lines in the frequency domain. Visualization with the plot functions enables the students to see the waveforms both in time and frequency. Other functions developed include a dual tone multi-frequency (DTMF) generator of the type used in cellular phones (Figure 2) and more audio effects functions. These include echo, reverberation and shelving digital filters implementing bass and tone controls of the type used in high fidelity systems. These simple examples that associate with digital filters help the students acquire some basic knowledge on filtering, frequency responses and other DSP concepts. In addition, the students perform exercises where filters are designed with graphical tools such as a graphical equalizer. More complex functions have also been provided for the student that wishes to explore more advanced DSP topics. For example a simple quantization function may be used to introduce the students to binary representations of the signals. The perceptual effects of quantization can become audible through the use of the sound player.

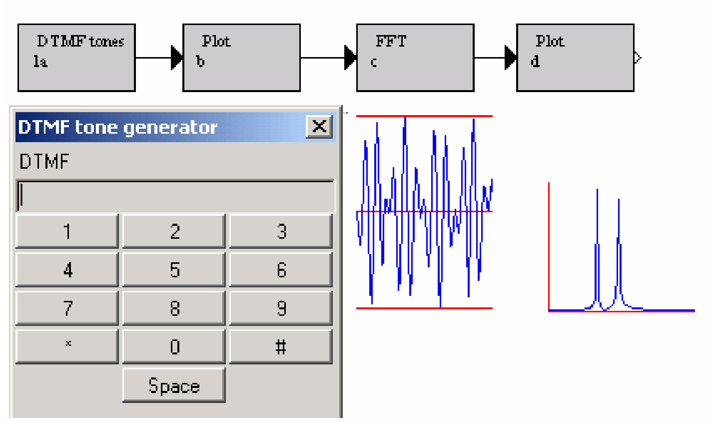

FIGURE 2
SIMULATION OF DTMF GENERATOR USED IN CELLULAR PHONES.

INSTRUCTION TOOLS; EMBEDDING INTERACTIVE SIMULATIONS IN WEB LECTURES

The simulation described above involves only three of the multitude of J-DSP functions that among others also include communications, controls and image processing functionality. J-DSP enables instructors to create a variety of simple simulations associated with elementary concepts that connect mathematics and DSP with applications in music, audio and video processing. These demonstrations can be formed and posted on the internet by simply using the script generator of J-DSP. J-DSP scripts [8], [10] can be embedded in HTML content and create lectures integrated with animated simulations. These simulations can be embedded in a seamless manner and with minimal effort simply by establishing graphically the simulation and activating the function *Export as Script* under the *File Menu*. The resultant script is shown in Figure 3. Cutting and pasting this in HTML content representing say a HS lecture module, enables the instructor to seamlessly create an animated MIDI simulation in the lecture without performing the tedious low-level Java programming usually involved in such tasks.

J-DSP FUNCTIONS DEVELOPED FOR HIGH SCHOOL TECHNOLOGY LECTURES

In the following we describe the functions developed to support HS technology lectures.

MIDI block

The Musical Instrument Digital Interface (MIDI) standard is ideal for teaching the utility of tonal signals in music. In its simplest form, MIDI music can be obtained by generating various tones at frequencies described by the standard. The J-DSP MIDI block has been designed to generate tones at the range of frequencies between 146.83 and 493.88 Hz, corresponding to one octave. Its dialog window resembles a small piano keyboard and was presented in Figure 1. When a piano key is pressed using the mouse, a tone is generated at the respective frequency. The tone length can vary between 256, (1 frame), 1280 (5 frames) and 16384 (64 frames) samples. The sampling frequency is 8 KHz. The MIDI block can also generate a sequence of pre-recorded tones. All tones generated can be used in a DSP simulation and are audible using the J-DSP provided sound player seen in Figure 4. More information on the MIDI standard can be found on the official MIDI webpage at www.midi.org.

DTMF block

Dual Tone Multi Frequency (DTMF) tones consist of a combination of two basic sinusoidal tones. The basic tone frequencies are given in Table I. In order to generate a single tone using the block, the user must press the appropriate button. If for example button 8 is pressed, frequencies $f_1 = 852$Hz and $f_2 = 1336$Hz are selected from Table I and used to generate two sinusoids that are combined with a sampling frequency of 8KHz.

TABLE I
DTMF TONE PAIR FREQUENCIES

Low band (Hz)	High band (Hz)		
	1209	1336	1477
697	1	2	3
770	4	5	6
852	7	8	9
941	*	0	#

Echo block

The echo block is a good example for introducing students to the notion of a system that adds components of the signal through a difference equation. The students can therefore be exposed to the notion of input and output. Subsequently, since the input-output relationship in echo blocks can also be viewed as a filter equation the students can get their initiation to filtering. As the name implies the *Echo* block generates an echo effect. The basic echo effect is obtained by mixing the input signal with its delayed version. The proportion of the delayed signal and amount of the delay relative to the original (straight through) signal determines how perceptible the echo is. The input-output relation is given below:

$$output(t) = input(t) + A\ input(t\text{-}delay)$$

In order to obtain a perceptible echo, R should be relatively large (a value of at least 800 is recommended). The block dialog window is shown in Figure 5.

FIGURE 3
J-DSP SCRIPTS TO EMBED SIMULATIONS IN WEB LECTURES.

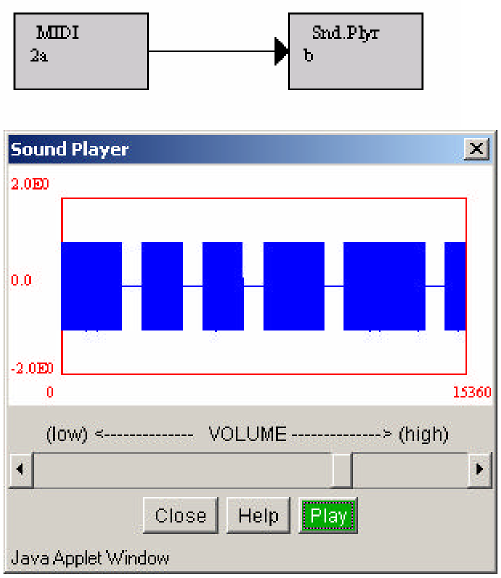

FIGURE 4
MIDI SIMULATIONS AND SOUND PLAYER WITH MIDI TONE RECORDING.

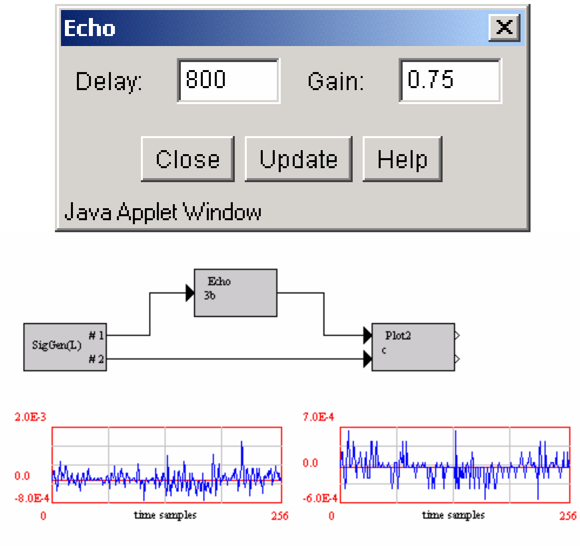

FIGURE 5
ECHO DIALOG WINDOW AND SIMULATIONS.

Reverberation block

The *Reverb* block generates a reverberation effect of the input signal. The basic reverberation effect is obtained by mixing the input signal with a delayed output. This form of feedback results in multiple echos. The input-outpout relation is.

$$output(t) = input(t) + A\ output(t\text{-}delay)$$

where A is the attenuation constant ($|A|<1$). The block dialog window is shown in Figure 6. The *Reverb Demo* block simulates five specific cases of reverberation effect. These are “Cavern”, “Dungeon”, “Garage”, “Acoustic Lab” and “Closet”.

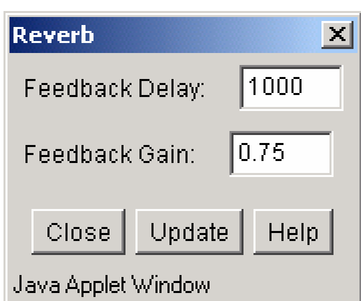

FIGURE 6
REVERB DIALOG WINDOW.

Tone Controls block

In order to introduce HS students to filtering and frequency responses we used the paradigm of bass/treble controls as used in audio systems. The *Tone Controls* block modifies the input audio signal by applying a gain or attenuation to the respective frequency bands. The low frequencies are affected by bass adjustments with the audio signal processed through low-pass shelving filters. The high frequencies are affected by treble adjustments with the speech signal processed through high-pass shelving filters. While we will not teach HS students z-transforms we give pertinent transfer functions below for the purpose of describing the filter functions compactly.

$$H(z) = 1 + (m-1)\frac{b(1+z^{-1})}{1-az^{-1}}$$

The coefficients a and b are directly related to the low-pass filter cutoff frequency f_c and the shelving filter boost/cut gain in dB g by:

$$a = \frac{1-(4/(1+m))\tan(\Omega_c/2)}{1+(4/(1+m))\tan(\Omega_c/2)}$$

$$b = (1-a)/2$$

where $\Omega_c = 2\pi f_c/f_s$, which is the normalized cutoff frequency; and $\mu = 10^{g/20}$. The transfer function of a high-pass shelving filter is implemented as

$$H(z) = 1 + (m-1)\frac{b(1-z^{-1})}{1-az^{-1}}$$

A simulation using the *Tone Control* block is shown in Figure 7.

Peaking Filter block

The *Peaking Filter* block introduces the user to the concept of a bandpass filter. This block allows students to attenuate audio signal components outside a specified frequency range. The center frequency can be varied. The transfer function of the peaking filter is implemented as

$$H(z) = 1 + (m-1)\frac{b(1-z^{-2})}{0.5 - a_1 z^{-1} + a_2 z^{-2}}$$

A simulation using the *Peaking Filter* block is shown in Figure 8.

Graphic Equalizer block

The paradigm of graphic equalization was used to introduce HS students to the cascaded peaking filters. The *Graphic Equalizer* block modifies the frequency response of the input audio signal by dividing its audible frequency spectrum into five frequency bands. It alters the frequency response of each band independently using the available slide bars. The graphic equalizer applies a set of peaking filters to modify the frequency response of the incoming audio signal. These filters maintain constant center frequencies (170Hz, 310Hz, 600Hz, 1kHz, 3kHz respectively) but support variable gains. These gains can be altered using the slide-bars on the dialog window of this block, each bar corresponding to a specific band. When all bars are viewed side by side, they resemble the frequency response applied to the input audio signal. The transfer function of the graphic equalizer is implemented as a cascade of several peaking filters. A simulation with the Graphic Equalizer is shown in Figure 9.

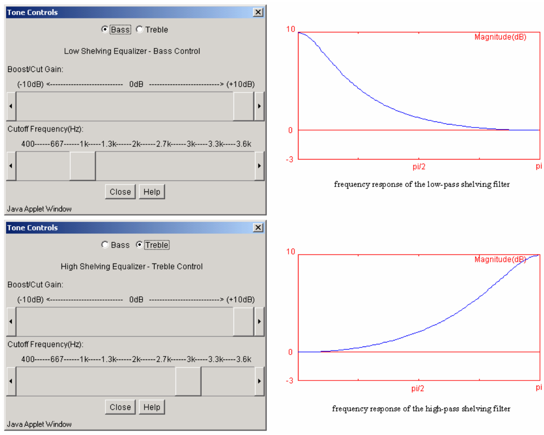

FIGURE 7
J-DSP SIMULATION USING THE TONE CONTROLS BLOCK.

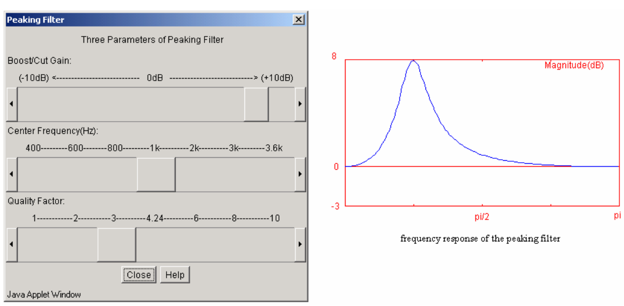

FIGURE 8
J-DSP SIMULATION USING THE PEAKING FILTER BLOCK.

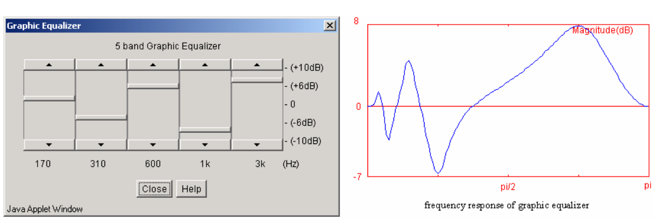

FIGURE 9
J-DSP SIMULATION USING THE GRAPHIC EQUALIZER.

Simple J-DSP Exercises with digital filters for use by HS Students

After establishing the concept of filtering through the tone and equalizer examples, we introduce exercises where HS students are asked to practice with the parameters of simple 1st order digital filters. The first exercise introduces to the students some basic concepts of digital filters. We use a digital filtering simulation that consists of five blocks as shown in Figure 10. The students go through a brief description of the five blocks, i.e., the signal generator, the filter, the coefficient, the frequency response, and the plot block.

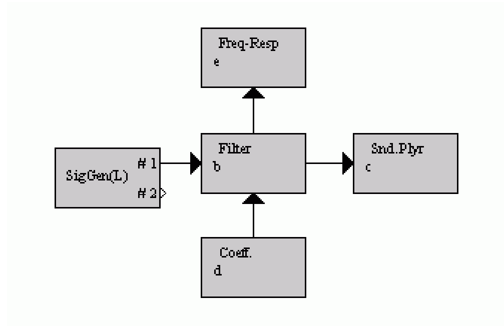

FIGURE 10
SOURCE - FILTER CONFIGURATION IN J-DSP USED TO SIMULATE DIGITAL FILTERS WITH SPEECH SAMPLES.

The first task involves the computation of a filter output for certain audio inputs. The participants are asked to examine the frequency response of the filter and observe the output. Then the students are asked to implement other filter functions and observe the new frequency response, and state the type of the realized filter (low pass or high pass). The objective of the second exercise is for the students to understand the perceptual differences between all-pass, low-pass and high pass filtering in audio signals. This is done by having the students compare a speech record from the signal generator and listen to the processed output using the sound player block. Following that, the students are asked to modify the filter functions by adjusting the appropriate filter coefficients. The students are requested to answer a few general questions on both exercises that assess their understanding of the differences between low and high-pass filtering in terms of frequency responses, parametric dependencies and perceptual quality.

ADVANCED EXPOSITORY SIMULATIONS

In order to expose HS students to some of the more advanced J-DSP simulations that are associated with current technologies we formed special web pages that explain certain DSP applications in qualitative terms. Then we exposed the students to pre-programmed J-DSP simulations. The students were asked to run these simulations and observe the signals at designated points where plot blocks were established. These simulations use advanced functions of J-DSP and are described below.

Simulation of the LPC Compressor used in Cell phones

The web module in Figure 11 qualitatively describes how the human speech production system is modeled with linear predictive coding (LPC) filters and signal generators in what is known as source-filter configuration.

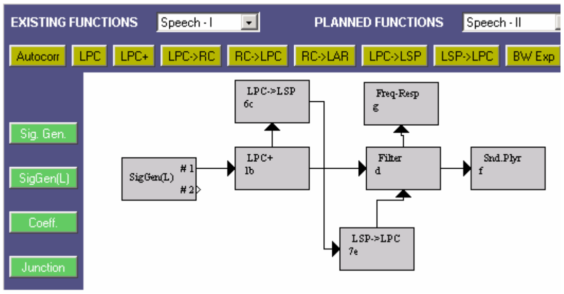

FIGURE 11
SIMPLIFIED LPC VOCODER AS USED IN CELL PHONES.

The source represents the activity of the lungs and vocal chords while the filter represents the mouth and nasal cavities. Several figures are given to describe qualitatively this model and then the students are asked to load a simulation and follow step-by-step instructions that help to visualize several aspects of this simulation. Each plot was carefully selected to represent a waveform which can be explained qualitatively. The simulation helps the students develop some basic intuition on this exciting application.

Simulation of Elements of MP3 Players

MP3 players are quite popular among HS students. In this module we attempt to teach them the notion of a high-fidelity frequency response and how different compression systems attempt to preserve it. We simulate for that purpose a filter bank along with a transformation. The concept of perceptual masking and the process of "exploiting" the properties of the human hearing mechanism are explained qualitatively.

Simulation of a JPEG Compressor

Finally, using the 2-D capabilities of J-DSP, we can display and process picture related data streams. We again provide simplified and qualitative descriptions while avoiding the details of the implementation. We then expose HS students to a J-DSP simulation of the Discrete Cosine Transform (DCT) and show them how it is used to achieve compression of the data stream. Students are asked to visualize the 2-D signal at different stages.

ASSESSMENT PLAN

Assessment results are not available at the current time but will be compiled during a science summer camp sponsored by ASU. We intend to run "pre" and "post" tests to assess whether J-DSP has accelerated the learning curve on several of the technology topics covered. During the fall of 2003 we will train instructors in select Phoenix high schools to run these modules along with the software as part of afternoon or Saturday activities.

CONCLUDING REMARKS

In this paper, we described the development of several functions in J-DSP that are intended to introduce HS students to DSP concepts as used in several exciting applications. The modules start from simple concepts using MIDI and DTMF encoders and then move to more involved simulations using filters and echo/reverberation systems. Finally some advanced simulations and demonstrations have been developed to familiarize HS students with the DSP concepts used in cellular telephony, JPEG compression, and MP3 players.

ACKNOWLEDGMENT

The J-DSP project is funded in part by the NSF CCLI program grant DUE 0089075.

REFERENCES

[1] Yoder, M., A. et al. "The Infinity Project: building a high school curriculum which emphasizes the engineering, math, and science principles of modern technology," *30th Annual Frontiers in Education Conference*, Vol. 1, 2000

[2] McClellan, H., J., Schafer W., R. and Yoder, A., M., " DSP First: A multimedia approach" Prentice Hall, 1998.

[3] Spanias S., A., "Speech Coding: A Tutorial Review," *Proceedings of the IEEE*, Vol. 82, No. 10, October 1994, pp. 1441-1582.

[4] Spanias S., A. and Painter, T., "An Educational Software Tool for the Study of Speech Coding Algorithms in a DSP Class," *Special Issue on DSP Education, IEEE Trans. on Education*, Vol. 39, May 1996, pp. 143-152.

[5] Atti V. and Spanias S., A., "A Simulation Tool For Introducing Algebraic CELP (ACELP) Coding Concepts In A DSP Course," *IEEE 2002 DSP Workshop*, Callaway, Georgia, October 2002

[6] Painter, T. and Spanias, S., A. "Perceptual Coding of Digital Audio," *Proceedings of the IEEE*, Vol. 88, No.4, April 2000, pp. 451-513 (2002 IEEE Donald G. Fink Prize Paper Award),

[7] Ramapriya, R. and Spanias, S., A. "A Simulation Tool for introducing MPEG - Audio (MP3) concepts in a DSP course," *Proc IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP-2002)*, Orlando, May 2002.

[8] Spanias, S., A. et al., "On-Line Laboratories For Speech And Image Processing and for Communication Systems Using J-DSP," *IEEE 2002 DSP Workshop*, Callaway, Georgia, October 2002.

[9] Spanias, S., A. et al., "Development of a Web-based Signal and Speech Processing Laboratory for Distance Learning," *ASEE Computers in Education Journal*, Vol. X, No.2, April-June 2000, pp. 21-26.

[10] Spanias, S., A. and Bizuneh, F., "Development of new functions and scripting capabilities in Java-DSP for easy creation and seamless integration of animated DSP simulations in web courses," *Proc. IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP-2001)*, Salt Lake City, May 2001.