

Interactive Online Undergraduate Laboratories Using J-DSP

Andreas Spanias, *Fellow, IEEE*, and Venkatraman Atti, *Student Member, IEEE*

Abstract—An interactive Web-based simulation tool called Java–DSP (J-DSP) for use in digital signal processing (DSP)-related electrical engineering courses is described. J-DSP is an object-oriented simulation environment that enables students and distance learners to perform online signal processing simulations, visualize Web-based interactive demos, and perform computer laboratories from remote locations. J-DSP is accompanied by a series of hands-on laboratory exercises that complement classroom and textbook content. The laboratories cover several fundamental concepts, including z transforms, digital filter design, spectral analysis, multirate signal processing, and statistical signal processing. Online assessment instruments for the evaluation of the J-DSP software and the associated laboratory exercises have been developed. Pre/postassessment data have been collected and analyzed for each laboratory in an effort to assess the impact of the tool on student learning.

Index Terms—Assessment instruments, distance learning, interactive digital signal processing (DSP) demos, online signal processing laboratories, tools to embed digital signal processing (DSP) simulations in Web lectures, Web-based simulations.

I. INTRODUCTION

THE increasing popularity of Web-based education and the advancements in streaming media applications have had a profound impact on distance-learning programs. Web publishing of supplemental course materials, Internet courses accompanied by streaming audio/video lectures, and Java concept-visualization software have become common in *technology-driven* university education. In fact, during the past decade, several notable efforts to develop interactive Java applets have been undertaken in digital signal processing (DSP)-related courses [1]–[4]. In addition, award-winning textbooks and MATLAB software that introduces basic DSP concepts to first-year undergraduate (UG) students [5], [6] have also been developed. Graphical simulation tools to teach DSP and communications have also been presented [7]. Several stand-alone laboratories and supplementary MATLAB tools for use in signal processing and control systems classes have been proposed in [8] and [9]. Brown *et al.* [10] proposed an educational tool called Mallard for use in a DSP course. An application-oriented approach for UG DSP education has

been proposed in [11]. An online DSP course for “practicing engineers” has been offered [12], [13] for continuing DSP education.

Although many Web courses have been designed for distance education using a variety of innovative tools, the provision of hands-on laboratory experiences to distance learners [14]–[16] has been an open problem. To address this problem, an interactive software environment called Java–DSP (J-DSP) has been developed at Arizona State University (ASU), Tempe. J-DSP [20] is an object-oriented programming environment that enables students to perform interactive computer simulations on the Internet. All functions in J-DSP appear as graphical blocks (Fig. 1) that are grouped according to their functionality.

J-DSP has a rich suite of signal processing functions that facilitate interactive online simulations of basic and advanced signal processing algorithms. Basic functions include signal generators, real and complex arithmetic functions, convolution and digital filtering, the fast Fourier transform (FFT), and time and frequency response plots. Advanced functions include digital filter design, statistical signal processing, spectral analysis, multirate DSP, quadrature mirror filter (QMF) banks, linear predictive coding, hidden Markov model (HMM) estimation, and blocks that support speech recognition.

The J-DSP laboratory software and exercises address the following objectives:

- 1) provide online laboratory experiences to DSP students in electrical engineering;
- 2) enable programmable Internet simulations with embedded animations;
- 3) enhance learning by exposing students to hands-on manipulation of signals and systems;
- 4) provide intuition and complementary information usually not available in lectures and textbooks.

To accomplish these objectives, pedagogical structures have been formed that are described later in the paper. In addition, an assessment process has been designed to measure the following:

- 1) the progress toward accomplishing these objectives;
- 2) the effectiveness of the pedagogies adopted;
- 3) the overall impact of the software tool on an undergraduate course.

The rest of the paper is organized as follows. Background on J-DSP along with the utility of the J-DSP laboratory software (“labware”) is given in Section II. Section III describes the J-DSP simulation environment. Section IV gives an overview of some of the *basic* signal processing functions available in J-DSP. Section V reviews some of the *advanced* functions in

Manuscript received August 2, 2004; revised June 20, 2005. This work was supported in part by the National Science Foundation (NSF) Course Curriculum and Laboratory Improvement (CCLI) program (NSF-DUE-CCLI-0089075) and through the Combined Research and Curriculum Development (CRCDD) initiative.

The authors are with the Department of Electrical Engineering, Ira A. Fulton School of Engineering, Arizona State University, Tempe, AZ 85287-5706 USA (e-mail: spanias@asu.edu; atti@asu.edu).

Digital Object Identifier 10.1109/TE.2005.854569

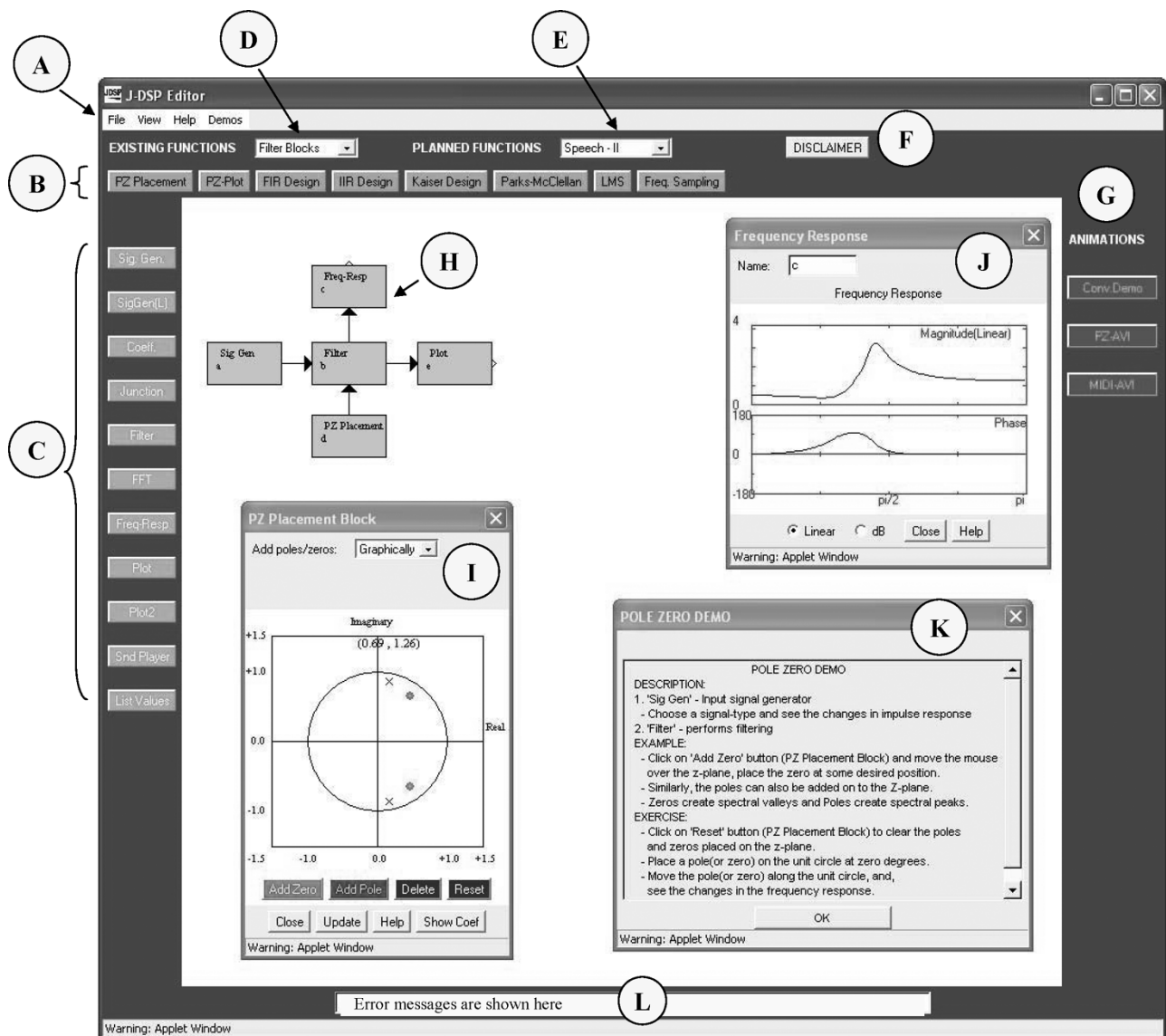


Fig. 1. J-DSP simulation environment. A: Menu items; B: filter blocks (this section changes according to the selection of D or E); C: permanent blocks; D and E: list menu to select the group of functions; F: disclaimer; G: interactive visual demonstrations; H: simulation flowgram; I: dialogue window (corresponding to the pole-zero (PZ) placement block in the block diagram H); J: plot window to view the results; K: help window provided for all the blocks; and L: a field that shows error messages and warnings.

J-DSP that cover the following interdisciplinary areas: speech processing, communications, controls, image processing, and time–frequency analysis. In an effort to introduce high school (HS) students to DSP and multimedia, several HS-friendly functions [e.g., bass/treble controls and musical instrument digital interface (MIDI)] have been carefully designed. These functions are described in Section VI. Section VII deals with the J-DSP scripting tools that enable instructors to embed interactive simulations in Web courses. Section VIII addresses the pedagogical and adaptability issues associated with the J-DSP labware. Brief descriptions of the online laboratories developed specifically for the EEE 407 DSP course at ASU are also included in Section VIII. Section IX analyzes in detail the J-DSP assessment results and describes the impact of the J-DSP tool on student learning and on the EEE 407 course curriculum at ASU.

Section X presents the conclusion, J-DSP extensions to other interdisciplinary areas, and future directions.

II. J-DSP BACKGROUND

The initial version of the J-DSP software was tested at ASU in a senior-level Electrical Engineering DSP course (EEE 407) during the late 1990s [21]. In 2000, the National Science Foundation (NSF) awarded a Course-Curriculum and Laboratory Improvement (CCLI) grant to upgrade the DSP functionality and extend the use of J-DSP in other areas, such as communications [30], image processing [25], [29], and controls [31]. Functions on bio-design/genomics [37], psychoacoustics and auditory psychophysics [39], and speech analysis–synthesis [27], [28], [35] were also supported by the NSF through a course curriculum and research development program.

The J-DSP Version 1 (CD-ROM ISBN 0-9 724 984-0-0—2002) consists of approximately 42 000 lines of Java code and was disseminated to 25 universities worldwide [38].

A. The Utility of the J-DSP Labware

J-DSP is different from the platform-specific commercial simulation packages, such as MATLAB, SIMULINK, or the LabView. Although the “block diagram approach” is used in the aforementioned commercial packages as well, J-DSP is unique in that it is platform independent and does not require installation. The program can run from any Java-enabled Internet browser and, hence, is *freely* and *universally* accessible over the World Wide Web (WWW) [20]. Moreover, the inherent Java backbone enables J-DSP to be interfaced with streaming media, Internet lectures, and animated visual demonstrations [20], [24]. J-DSP is also valuable for instructors since it enables them to create and embed seamlessly interactive demos in their Web lecture content using the built-in J-DSP scripts [24]. The interface to MATLAB [36] allows synergies with the MATLAB environment.

B. J-DSP Online Laboratories and Electronic Laboratory Report Submission

Several laboratory exercises have been developed to expose students to the key DSP concepts [20]–[23], [25], [26]. The J-DSP computer laboratories perform the following functions:

- 1) provide hands-on exposure to digital filters and FFTs;
- 2) expose students to quantization effects, filter bank simulations, and random signal analysis;
- 3) provide visualization examples relating time-domain, z -domain, and Fourier transforms;
- 4) provide hands-on experiences with finite-impulse response (FIR) and infinite-impulse response (IIR) digital filter design;
- 5) expose students to periodograms, correlograms, and linear prediction.

A custom-made electronic laboratory report submission tool has been devised to allow students to complete a media-rich online laboratory report [20], [23]. Upon submitting the report, all of the students’ answers, comments, and graphs are placed together in a static hypertext markup language (HTML) page that is available to the instructor for grading.

C. J-DSP Assessment Process

Assessment of J-DSP has been carried out on a semester-by-semester basis. Web-based assessment instruments have been developed and posted on the J-DSP website [20]. Initially, J-DSP assessment was performed within the DSP course evaluation forms. Since 2002, the degree of learning attributed specifically to J-DSP and the associated online laboratories have been evaluated every semester separately. Comprehensive statistical analysis of the J-DSP assessment results based on the effect size (ES) measures [19] has been included in Section IX. Assessment results show that students found J-DSP simulations and computer laboratories highly intuitive [23], [25], [26]. In fact, 95% of the students liked the concept of Internet-based

simulations and responded that it took them just a few minutes to become familiar with the J-DSP environment.

III. THE J-DSP SIMULATION ENVIRONMENT

J-DSP provides a user-friendly environment that exploits the graphical capabilities of Java. Fig. 1 shows the J-DSP simulation area. In this figure, section A represents menu items, and section B corresponds to a floating panel. This section changes according to the selection of list items D or E. All functions in J-DSP appear as graphical blocks. Each block is associated with a specific signal-processing function. Panel C shows some of the frequently used blocks; list menus D and E include a group of existing and planned functions, respectively. Panel G provides interactive visual demonstrations. A variety of DSP algorithms can be simulated using a drag-and-drop process to establish the desired blocks and then by connecting them to form a signal flow. H shows an example simulation-flow diagram (in short, simflowgram).

Each block is linked to software that performs a specific signal-processing function. Block parameters can be edited through the corresponding dialogue windows. For example, I represents the dialogue window corresponding to the pole-zero (PZ) placement block shown in H. Signals at any point of a simulation can be visualized through the appropriate blocks, e.g., a plot window J. System execution is dynamic, meaning that any change at any point of a simulation will automatically take effect in all subsequent blocks. A help window (K) is provided for all the functions to facilitate students with quick understanding and use of J-DSP blocks. The J-DSP simulation environment has also been equipped with a series of troubleshooting and exception-handling features.

IV. SIGNAL-PROCESSING FUNCTIONS IN J-DSP

J-DSP has a suite of signal-processing functions that include signal generators, digital filter design functions, multirate DSP, and statistical signal-processing functions [20]–[23].

- 1) *Basic DSP functionality*: Some of the basic DSP functionality implemented in J-DSP includes simple arithmetic functions, windowing, up-sampling, down-sampling, convolution, filtering, signal-to-noise ratio (SNR), the FFT and the inverse FFT (IFFT), and PZ placement.
- 2) *Filter design*: J-DSP supports FIR filter design based on the windowing method, the Kaiser design, the Parks–McClellan (Min–Max) algorithm, and frequency-sampling method. IIR filter design experiments based on IIR analog filter approximations (Butterworth, Chebyshev, and elliptic filter realizations) can also be performed in J-DSP.
- 3) *Multirate DSP*: J-DSP offers the capability to up-sample or down-sample a signal. This capability enables building multirate systems, such as the decimators, interpolators, and QMF banks. Students gain knowledge of filter bank design considerations as employed in a typical audio coding application.
- 4) *Statistical DSP*: Spectral estimation techniques, such as the classical and parametric techniques, have been

implemented in J-DSP. Autocorrelation, correlograms, periodograms, symmetric autocorrelation, the Levinson–Durbin algorithm, and autoregressive (AR) spectral estimation are some of the statistical DSP functionalities supported by J-DSP. In order to compute the time-varying spectra of speech and other signals, a spectrogram function has also been developed.

Several of these functions have been inspired by interactions with students in an effort to explain concepts that were not immediately obvious from textbooks, homework sets, or blackboard-type explanations.

V. J-DSP FUNCTIONALITY FOR USE IN INTERDISCIPLINARY AREAS

Some of the advanced functions developed in J-DSP cover the following areas: communications [30]; control systems [31]; acoustics and music synthesis [32]; genomics and bio-informatics [37]; psychoacoustics and auditory psychophysics [39]; machine-learning and voice recognition modules [35]; speech analysis–synthesis techniques [27]; time–frequency transforms [28]; and image and two-dimensional (2-D) signal-processing methods [29]. A brief overview of some of these J-DSP functions follows.

A. Communications

Analog modulation blocks, such as amplitude modulation (AM) (double-sideband suppressed-carrier AM, single-sideband AM, and conventional AM) and angle modulation (frequency modulation and phase modulation) have been developed [20], [30]. Digital modulation schemes include binary pulse amplitude modulation (PAM), m -ary PAM, phase-shift keying (PSK), quadrature PSK, and m -ary PSK. Receiver blocks supported are the matched-filter demodulator and maximum-likelihood detector. In order to compute bit-error probabilities and to plot the performance curves, a Monte Carlo simulation block has been developed. These functions have been used in a senior-level communications-related course: EEE 455 at ASU [20]. The EEE 455 course curriculum includes seven online laboratories that cover the following topics [33]: AM schemes, angle modulation schemes, noise effects in analog communications, digital modulation schemes, intersymbol interference (ISI) and channel equalization, matched filter, and channel coding.

B. Control Systems

In addition to the communications functions, more functionality has been developed to facilitate control systems simulations [31]. The controls systems simulation capabilities of J-DSP involve blocks for state space and transfer function representation of systems, a simple adder, gain, a step-signal generator, and Bode and Nyquist plot. The J-DSP controls functions have been used in a UG feedback control systems course, i.e., EEE 480 at ASU. (For a detailed description of the online control systems laboratories and the pedagogies adopted, readers are referred to [20] and [31].)

C. Acoustics, Music Synthesis, and Audio Effects

J-DSP enables simulations relating filter design with some real-life applications, such as echo, reverberation, bass/treble control, and graphic equalizer. These filter design techniques form a special class of digital filters in audio processing and high-fidelity systems. The audio effect blocks supported in J-DSP include the dual-tone multiple-frequency (DTMF) generator, the MIDI generator, and shelving–peaking digital filters. More detailed descriptions on J-DSP simulations involving the DTMF and MIDI tone generator are given in Section VI.

D. Genomics and Bio-Informatics

Recent findings on DNA (deoxyribonucleic acid) sequences and microarrays provide great promise in the field of genomic signal processing. Specific functions in J-DSP to analyze the DNA sequences [37] include numerical mapping, FFT power spectrum, and correlations. Through the numerical mapping function, nucleotide sequences can be transformed into the numerical domain through binary, complex, or real number mappings.

E. Psychoacoustics and Auditory Psychophysics

In [39], several innovative software modules and computer experiments have been presented for exposing students to 1) the perceptual masking properties of the human ear, 2) the perceptual entropy notions, and 3) the psychoacoustic models employed in the MP3 players. In particular, the laboratories cover several concepts, including the absolute threshold of hearing, critical-band analysis, the Bark scale, the spread of masking, and the simultaneous and temporal masking effects [18]. In addition to the online laboratories, an FFT-based computer project has been designed to provide students with hands-on experiences to the psychoacoustic model employed in the International Organization for Standardization/International Electro-technical Commission (ISO/IEC) MPEG-1 audio standard [18]. The computer laboratories and the MPEG-1 psychoacoustic model-1 project were first used in a multimedia class in summer 2004 at ASU.

F. Machine Learning and Voice Recognition Modules

Inspired by the applications that involve machine-human interface, more importance has been given toward the development of speech recognition modules. In particular, simulations to teach the basic concepts of isolated digit recognition and hidden Markov model (HMM) training have been developed [35]. The software is complemented by streaming lectures, a set of online demonstrations with animation, and exercises that take the student through HMM training.

G. Speech and Audio Coding

Emphasis has been given to design interactive graphical modules that highlight the basic concepts employed in speech and audio standards. In particular, the speech modules in J-DSP include linear predictive coding (LPC), Federal Standard (FS)-1015, FS-1016, speech analysis–synthesis techniques, open-loop and closed-loop pitch estimation methods, vector quantization, algebraic codebook concepts, low-delay coding, and some variable-rate codecs [17]. Furthermore, work has

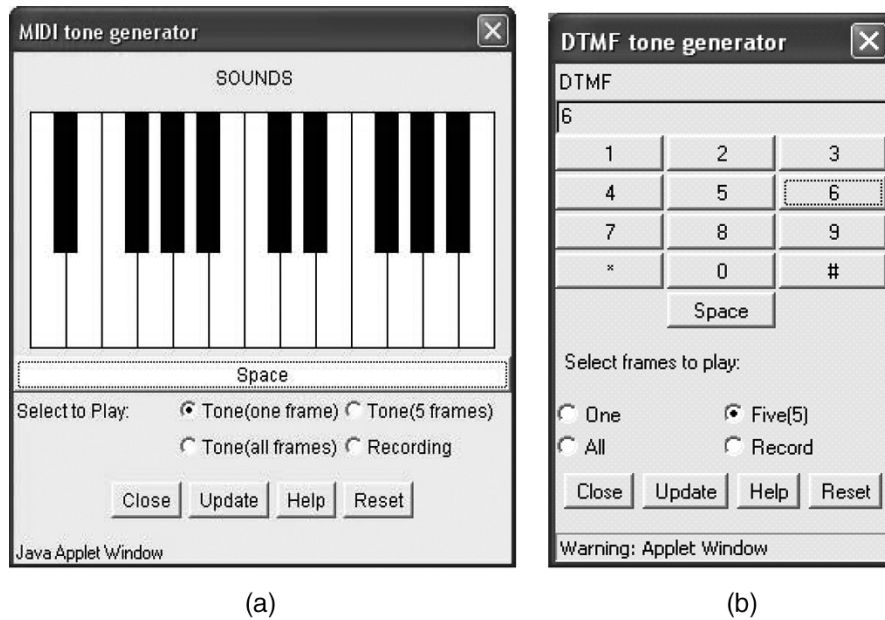


Fig. 2. (a) MIDI function. (b) DTMF function in J-DSP.

been done in the area of perceptual audio coding in parallel to the development of speech modules. These include graphical modules that expose students to the analysis–synthesis filter bank structures, psychoacoustic analysis, modified discrete cosine transform (MDCT) analysis, and quantization.

VI. J-DSP FUNCTIONS TO INTRODUCE HIGH SCHOOL STUDENTS TO DSP AND MULTIMEDIA

Several notable efforts have been made to introduce basic DSP concepts in HS curricula and to college freshmen, e.g., the Ptolemy project [1], the Infinity project [2], and DSP First [5]. 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 [17] and MP3 [18]. 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 and surround-sound systems; difference equations implement filters in audio graphic equalizers and cellular telephony handsets. To adopt J-DSP for use in HS environments, a series of J-DSP functions that are HS friendly have been developed.

A MIDI generator block [Fig. 2(a)] was developed to help students associate musical tones with sinusoidal functions. The MIDI block dialogue window resembles a small piano keyboard and was presented in Fig. 2(a). When a piano key is pressed using the mouse, a tone is generated at the respective frequency. An FFT function and 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 students to see the waveforms both in time and frequency.

Other functions developed include a DTMF generator [Fig. 2(b)] of the type used in cellular phones, echo [Fig. 3(a)], reverberation [Fig. 3(b)], 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 (Fig. 4).

VII. J-DSP SCRIPTING TOOLS FOR INSTRUCTORS

This section addresses two unique tools of J-DSP for Web-based instruction: 1) J-DSP scripts [24] enabling instructors to create and seamlessly integrate J-DSP simulations in their Web courses and 2) automatic MATLAB script generation [36].

A. J-DSP Scripts

J-DSP scripts enable instructors to create a variety of simple simulations and seamlessly integrate them into their Web content or online course lectures. In particular, J-DSP versions have been designed with the ability to interpret parameters contained in a simple HTML file and, in turn, load the J-DSP editor and run a simulation described by these parameters. This J-DSP editor capability has been designed to allow for interactive J-DSP content to be integrated in Web pages, thereby enabling instructors to establish DSP visualization examples without engaging in the tedious task of Java programming. Fig. 5 shows an example simulation that demonstrates the script-generating procedure. This procedure consists of the following three steps.

- Step 1) Set up the simulation by creating a flowgram.
- Step 2) Select File→Export as Script as shown in Fig. 5(a). A new window, called the Script Export window [Fig. 5(b)], containing the simulation description appears.
- Step 3) Export this J-DSP script code into an HTML file.

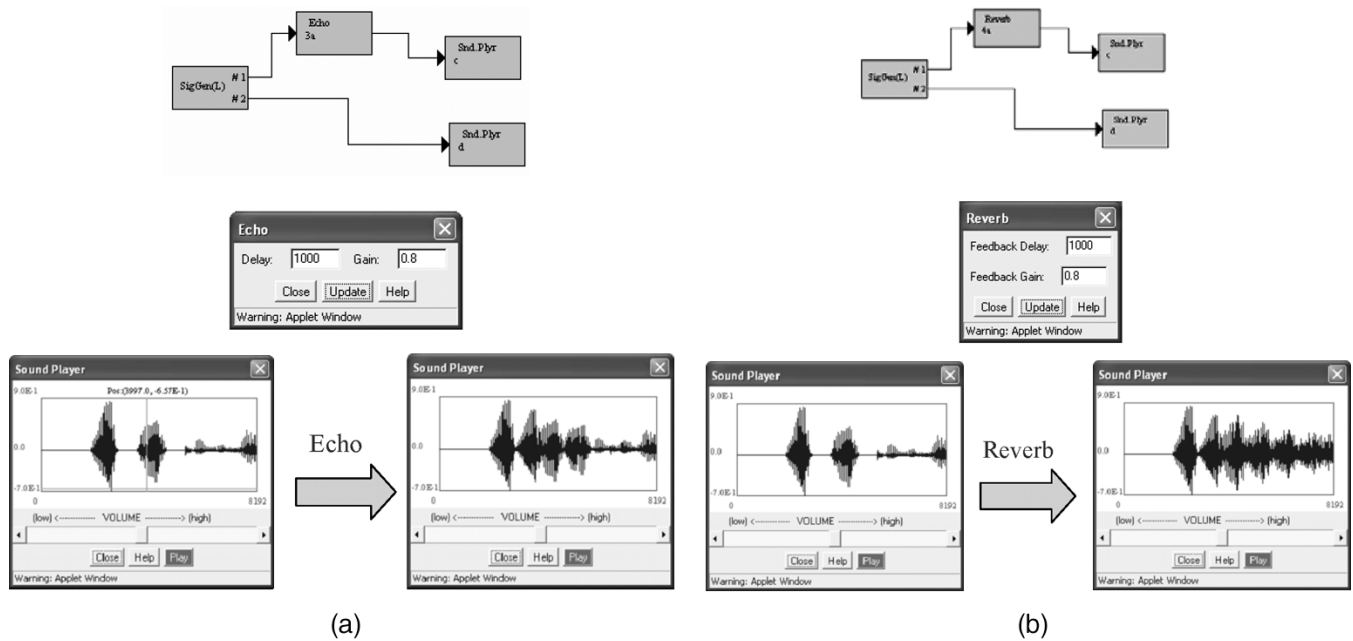


Fig. 3. Filter design for audio effects simulations: (a) echo and (b) reverberation.

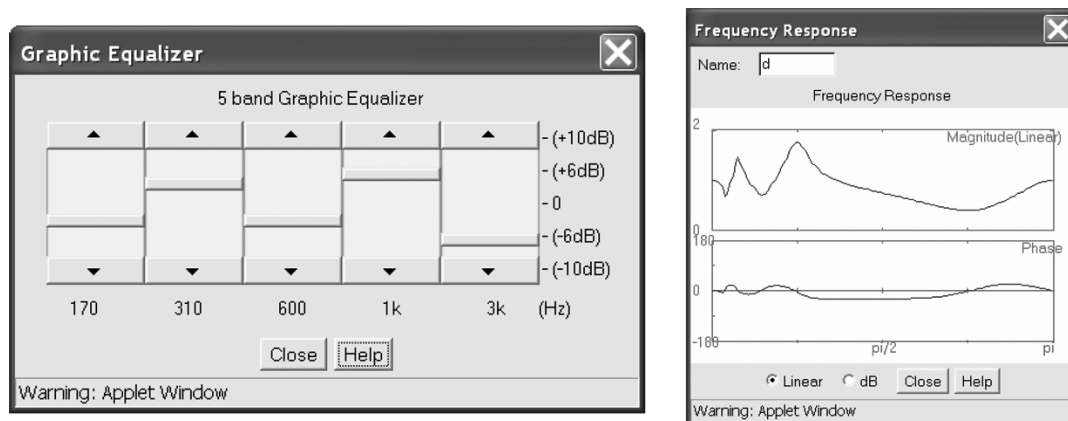


Fig. 4. Graphic equalizer simulation in J-DSP.

B. Interfacing With MATLAB

The synergy of the J-DSP object-oriented environment with MATLAB programming enables students and instructors to exchange data and perform DSP simulations on both these platforms. This new feature acts as an interface between the MATLAB and the J-DSP object-oriented environment. In particular, this interfacing attribute allows users to “translate” the J-DSP simflowgram into MATLAB code. Moreover, this feature enables users to repeat, verify, and (most important) expand J-DSP simulations in the MATLAB environment. The latter provides a vast selection of functionalities for more complex simulations than the object-oriented environment of J-DSP. The advantage here is that complex algorithmic programming can be done visually on the Internet using J-DSP, executed in MATLAB. Conversely, MATLAB programs can be mapped to flowchart-like diagrams and run in J-DSP [36].

VIII. J-DSP ONLINE LABORATORIES

This section places emphasis on the pedagogies adopted for the use of J-DSP in the EEE 407 course. J-DSP online laboratories are also summarized in this section.

A. J-DSP and the EEE 407 Course

The J-DSP laboratory software and the associated computer exercises have been used in the EEE 407 DSP course at ASU. A brief overview of the EEE 407 course structure follows.

1) *EEE 407 DSP Course at ASU:* The EEE 407 course is offered twice a year, during the spring and fall semesters. Some of the topics integrated in the EEE 407 course syllabus include difference equations; digital filters and bounded input–bounded output (BIBO) stability; z transforms and frequency response; FIR and IIR digital filter design; impulse-invariant methods and the bilinear transformation; frequency-domain analysis,

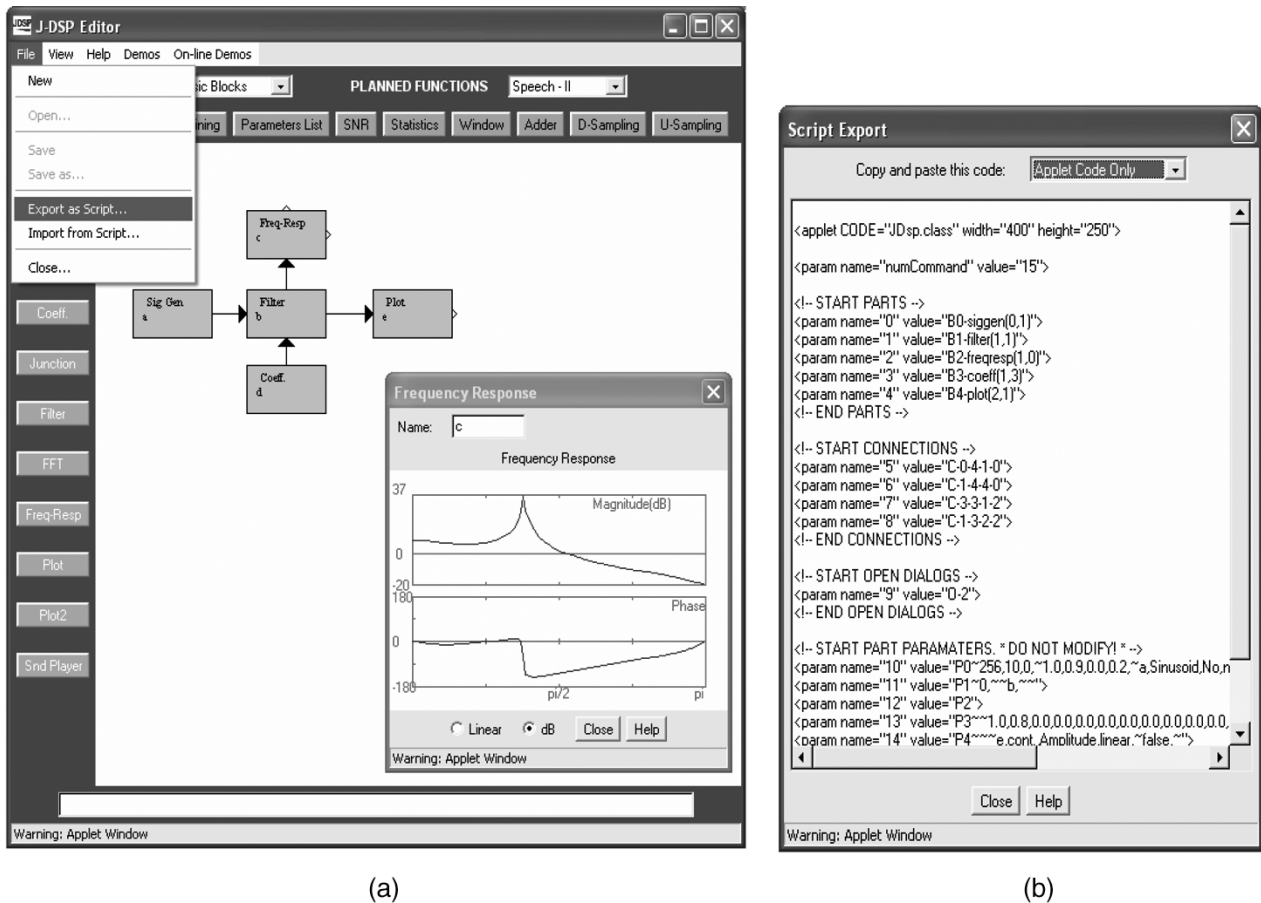


Fig. 5. J-DSP scripts enabling the saving capability: (a) J-DSP editor window and (b) generated J-DSP script.

discrete-time Fourier transform (DTFT), DFT, FFT, short-time Fourier transform (STFT) basics; multirate signal processing, simple sub-band QMF design; deterministic versus random sequences; stationary and ergodic sequences; and signal-processing application algorithm design and implementation. The course includes eight homework sets, six J-DSP computer laboratories, an application-oriented project (e.g., DTMF, noise canceling, LPC vocoders, and FFT-based speech enhancement), two midterm exams, and a final exam. The students' learning objectives associated with the EEE 407 course include the following:

- 1) understanding the issues and methods associated with the sampling of continuous time signals;
 - 2) becoming proficient in z -transform analysis;
 - 3) designing FIR and IIR digital filters;
 - 4) using the FFT;
 - 5) understanding elements of discrete-time random signal processing;
 - 6) developing software to simulate a modest DSP task associated with a modern application.
- 2) *Pedagogy:* The content in the J-DSP laboratory exercises is structured in a manner that follows the progression of topics in any undergraduate DSP course. In particular, the next laboratory exercise builds on knowledge gained from the previous one. The J-DSP laboratories and the visual demonstrations are placed right at the core of the EEE 407 course syllabus. A typical procedure, for example, can be summarized as follows.

First, a lecture including J-DSP visual demonstrations provides the necessary theoretical background. Second, a J-DSP laboratory assignment will help students to probe further into the details of the lecture. Finally, students are required to solve open-ended problems both as part of the homework and as part of the J-DSP laboratory exercises. This pedagogy used facilitates the “general” and “concept-specific” objectives associated with the J-DSP laboratories that were described in Sections I and II-B.

B. J-DSP Online Laboratories

Several hands-on J-DSP computer exercises have been carefully developed not only to reinforce the DSP concepts covered in class but also to expose students to complementary material that is not usually covered in detail neither in class nor in textbooks. The computational nature of the J-DSP laboratories and the inclusion of real-life signals made J-DSP particularly useful in providing engineering intuition and valuable hands-on experiences to the EEE 407 class participants. Currently, EEE 407 includes six computer laboratory assignments that are given on a weekly basis.

A typical scenario is for a student to read a high-level tutorial on each of the laboratories and then form and execute simulations. The J-DSP laboratories that have been outlined here, as well as additional ones not mentioned here, are given to students in detailed Web-based write-ups. These include a theory section and step-by-step instructions for operating the J-DSP

software and analyzing the intended results. The students perform the experiments and respond to questions in a quiz section. They then have to submit an electronic laboratory report containing relevant graphs that are ported from the J-DSP simulation environment.

1) *Lab-1: Difference Equations and the z Transforms:* This J-DSP laboratory introduces the students to the concepts of linear-time-invariant (LTI) systems, z transforms, and the impulse response of LTI systems. In particular, students observe the filtering effects and become familiarized with the source-filter configuration. Six problems have been developed and incorporated in this laboratory. In problem 1, students are asked to simulate a digital filter using a given transfer function. Fig. 6(a) shows an example digital filter simulation performed using J-DSP. In problem 2, students design and simulate a digital oscillator for different sinusoidal frequencies. In problem 3, an FIR filter is provided, and the students are asked to observe the behavior of the systems for different input signal types. In problem 4, the students gain insight to the concepts of symmetric and asymmetric impulse responses. In problem 5, the students compute the transfer function for various PZ representations. In problem 6, the students simulate cascade and parallel configuration with two systems.

2) *Lab-2: PZ Plots and Frequency Responses:* This J-DSP laboratory deals with the effects of PZ locations on the magnitude frequency response. The students observe the variations in the frequency response by graphically moving the poles and zeros in the z domain. They also design low-pass and high-pass filters based on the PZ placement method. Four problems are included in this laboratory. In problem 1, the students are asked to find the poles and zeros and observe the frequency response of a given filter. In problem 2, the students observe the variations in the frequency response of a system by graphically moving the poles and zeros in the z domain or by manually entering the values in circular or polar coordinates. In problem 3, low-pass and high-pass filters are designed based on the PZ placement method [Fig. 6(b)]. In problem 4, the PZ locations and the frequency response for an all-pass filter is examined.

3) *Lab-3: FIR and IIR Filter Design:* This J-DSP laboratory deals with the design and analysis of various FIR and IIR filter design methods. These include filter design based on windowing, frequency sampling, and IIR analog filter approximations. FIR filter design experiments based on the Parks–McClellan algorithm form an important laboratory exercise that exposes students to the concepts of optimal filter design, linear-phase filters, minimum-phase design considerations, and stability issues of the designed filter. Seven problems are included in this laboratory. Problem 1 involves the design of the four types of FIR filters, i.e., students are asked to observe the frequency response and calculate time-domain/ z -domain symmetry, and group delay for the previously mentioned filter types. Problem 2 deals with the design of low-pass filters by truncating the ideal impulse response. The truncation is accomplished by windowing. The following window types are supported in J-DSP: rectangular, Bartlett, Hamming, Hanning, and Kaiser. In problem 3, the students are asked to design high-pass filters using the Kaiser design method. Problem 4 deals with the FIR filter design using the frequency sampling

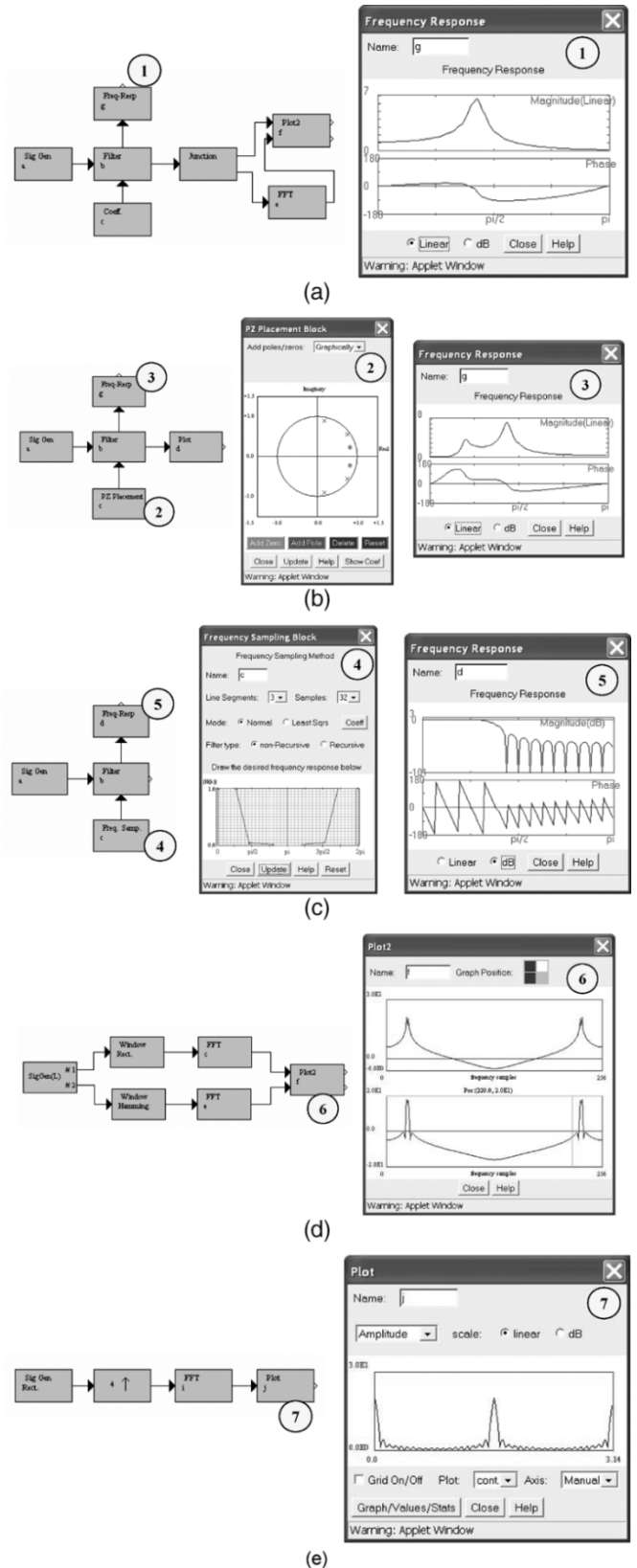


Fig. 6. (a) Example J-DSP simflowgram to simulate a digital filter. (b) Low-pass filter design using PZ placement method. (c) Frequency sampling filter design. (d) Windowing effects on the FFT spectra. (e) Duplication of spectral components as a result of up-sampling.

method [Fig. 6(c)]. In problem 5, the students are asked to design optimal FIR filters using the Parks–McClellan algorithm.

TABLE I
STATISTICS BASED ON THE GENERAL ASSESSMENT. TOTAL NUMBER OF STUDENTS THAT
CONTRIBUTED THE DATA = 87 (FROM SPRING AND FALL 2003)

Evaluation Questions	Strongly Agree (%)	Agree (%)	Neutral (%)	Disagree (%)	Strongly Disagree (%)
1. Establishing and connecting blocks is easy	53	39	7	1	0
2. The graphical interface of J-DSP is intuitive and user-friendly	31	63	5	1	0
3. I would consider J-DSP for small simulations apart from the lab exercises	86 (Yes)		-		14 (No)
4. In my opinion, this (J-DSP) type of on-line lab concept is beneficial for distance learning	97 (Yes)		-		3 (No)
5. With the help of a simple manual, it is possible to learn the basics of DSP by using J-DSP	86 (Yes)		-		14 (No)
6. The J-DSP should be established as a full-fledged tool	93 (Yes)		-		7 (No)
7. How long did it take to get familiar with the basics of the J-DSP environment	< 15 min : 36% 15-30 min : 32% 30-60 min : 24% > 60 min : 8%				

In problem 6, students are asked to compare the sidelobe levels obtained for the filters designed using the Parks–McClellan method, the Kaiser design, and frequency-sampling method. Problem 7 deals with the design of IIR filters based on the following methods: Butterworth, Chebyshev I, Chebyshev II, and elliptic.

4) *Lab-4: The Fast Fourier Transform (FFT)*: In this J-DSP laboratory, students gain familiarity with the estimation of DFT spectra, DFT spectral leakage, DFT resolution, Parseval's theorem for the DFT, and FFT properties and symmetries. Four problems are assigned in this laboratory. In problem 1, students examine the symmetries of the FFT. In problem 2, students see the effect of zero padding on the FFT spectra. In problem 3, the students are asked to examine and compare the FFT spectra of various signals. In problem 4, students are provided with two sinusoids that are closely spaced in the frequency domain and are asked to examine the FFT spectra after windowing. Fig. 6(d) shows the windowing effects on the FFT spectra.

5) *Lab-5: Multirate Signal-Processing and QMF Banks*: This J-DSP exercise deals with the effects and applications of sampling rate conversion and QMF banks. In particular, students experiment with aliasing and imperfect reconstruction in

decimation and interpolation of digital signals. Four problems are included in this laboratory. In problem 1, students examine the effect of down-sampling and up-sampling on FFT spectra. Fig. 6(e) shows the presence of imaging spectral components. In problem 2, students design a fractional sampler. In problems 3 and 4, students implement “two-band QMF” and “tree-structured QMF” in J-DSP.

6) *Lab-6: Introduction to Statistical Signal Processing*: This J-DSP laboratory teaches the basics of classical and parametric spectral estimation. Specifically, students become familiar with correlograms, periodograms, spectrograms, AR estimators, and LPC.

IX. J-DSP ASSESSMENT AND STUDENT FEEDBACK

The assessment results collected from the EEE 407 class (87 students) during the spring and fall 2003 are analyzed here. *General* and *concept-specific* evaluation forms have been developed. The general forms are used to obtain an overall subjective opinion on the J-DSP software. The concept-specific forms provide the laboratory evaluation and its impact on

TABLE II
STATISTICS BASED ON THE CONCEPT-SPECIFIC ASSESSMENT

Evaluation Questions	Strongly Agree (%)	Agree (%)	Neutral (%)	Disagree (%)	Strongly Disagree (%)
1. The J-DSP on-line labs helped me to understand the concepts of the z-transform	50	46	3	1	0
2. My understanding of the concepts of FIR and IIR filter design is enhanced by the J-DSP labs	42	47	8	3	0
3. The general concepts of using FFT in signal analysis is clear by performing a J-DSP simulation	24	61	13	2	0
4. I have learned how to generate a sinusoid with a digital filter	29	55	11	3	2
5. Enough information is available on the help screens and the dialog windows.	49		33.5		13.5
6. The relationship between the impulse response and the transfer function is clear	95 (Yes)		-		5 (No)
7. After performing the J-DSP lab it is clear that the FFT spectral resolution is limited by the FFT size, the window type, and the window size	99 (Yes)		-		1 (No)
8. J-DSP labs enhanced my learning of the basic DSP concepts (Labs 1-4 average statistics)	92 (Yes)		-		8 (No)

learning specific DSP concepts. The newest assessment instrument, called *prelab/postlab* assessment, focuses on evaluating whether learning of certain topics is attributed specifically to using J-DSP.

In particular, while designing the assessment instruments, the following criteria have been considered:

- 1) the simplicity and interactivity of the J-DSP simulation environment;
- 2) the adaptability of J-DSP and the associated online laboratories in DSP courses at other universities;
- 3) the effectiveness of the J-DSP laboratories and interactive simulations in teaching key DSP concepts;
- 4) the improvement in average student scores after performing J-DSP laboratories;
- 5) the improvement in number of students responding to a particular laboratory question correctly before and after viewing a J-DSP demo/interactive simulation;

- 6) the success of J-DSP in providing hands-on laboratory experiences to distance learners;
- 7) the impact of the J-DSP tool on the EEE 407 course at ASU;
- 8) the scalability of the J-DSP tool in terms of extending its functionalities to other interdisciplinary areas.

A. General Assessment

The general assessment questionnaire consists of a specific set of items that assess the ergonomics and the usefulness of the J-DSP software. In this assessment, students provide information on logistics, software capabilities, academic standing, browser compatibilities, expediency of the user manual, etc. Student responses revealed that the graphical user interface (GUI) and the free access to J-DSP were received positively. Most students found J-DSP simulations highly intuitive. In fact,

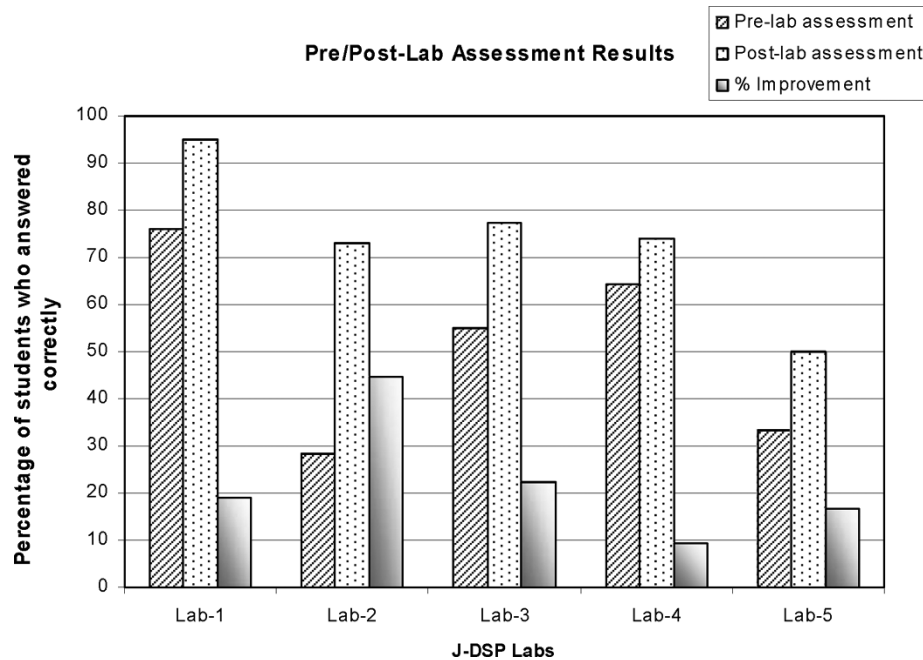


Fig. 7. Percentage increase in the number of students who have answered correctly.

95% of the students liked the concept of Internet-based simulations, and 70% of the students responded that it took them much less than 30 minutes to learn how to use J-DSP. Moreover, 85.5% of the students agreed that they would consider using J-DSP to construct their own educational simulations. Some of the statistics based on the general assessment is given in Table I.

B. Concept-Specific Assessment

The concept-specific assessment questions are directly related to the technical aspects of the J-DSP online laboratories. The concept-specific forms focus on each exercise by posing questions that determine whether the student has learned a specific DSP concept. For instance, 87% of the students agreed that the filter design exercise helped them understand which window is suitable for sharp transitions; 88% of the students understood better the signal symmetries in the FFT spectra because of J-DSP visualization; and 91% of the students reported that the z -transform exercise helped them understand the relation between the PZ locations and the frequency response plots. More results are given in Table II.

C. Prelab and Postlab Assessment

In the evaluation Question 8 shown in Table II, the students were asked if the J-DSP and the online laboratories enhanced the learning process. A total of 92% of the students responded positively. However, in order to obtain even more reliable statistical results for this scenario, the prelab/postlab assessment questionnaire was developed during the spring 2003 semester. In the prelab/postlab evaluation, the questions are technical and are posed to evaluate the student's understanding of the key DSP concepts before and after performing a particular J-DSP laboratory assignment. The prelab/postlab quiz and the laboratory are assigned after the relevant theory has been introduced in class. This assignment ensures that all the students have had some, or

TABLE III
IMPROVEMENT IN THE AVERAGE RAW SCORE
FROM THE PRELAB/POSTLAB ASSESSMENT

J-DSP Lab Number	Number of subtests or questions in each lab	Average raw score [MAX = 100]		
		Pre-lab	Post-lab	Improvement in the raw score
1	6	78.75	83.5	4.75
2	5	62.0	75.5	13.5
3	10	50.0	59.5	9.5
4	4	61.25	65.0	3.75
5	4	47.5	59.5	12.0

ideally the same, exposure to the topics covered in the laboratory so that the effect of the J-DSP laboratories on the student learning can be isolated. The students are asked to complete the prelab assessment before working on a J-DSP laboratory. After performing the J-DSP laboratory assignments, they submit the postlab assessment. The questions on the postlab assessment are the same as the prelab assessment but are given in a different order.

Table III lists the number of test items in each J-DSP laboratory and the corresponding improvements in the average raw score. J-DSP Lab-2, Lab-3, and Lab-5 resulted in significant improvements of 13.5, 9.5, and 12 marks (out of 100), respectively. This improvement can be attributed primarily to inclusion of simulations involving interactive graphics in these J-DSP laboratories. Moreover, this assessment result was influential in re-designing many of the J-DSP blocks to incorporate animations. This discussion is also well supported by Fig. 7, which depicts the percentage increase in the number of students who have answered the prelab/postlab questions correctly. The percentage

TABLE IV
ES ANALYSIS OF THE PRELAB/POSTLAB ASSESSMENT DATA

J-DSP Lab Number	Mean score		Standard deviation		Effect size $d = \frac{\mu_2 - \mu_1}{\sqrt{(\sigma_1^2 + \sigma_2^2)/2}}$	Correlation coefficient $r = \frac{d}{\sqrt{d^2 + 4}}$
	Pre-lab μ_1	Post-lab μ_2	Pre-lab σ_1	Post-lab σ_2		
1	78.75	83.5	16.41	15.05	0.302 (Medium)	0.149
2	62.0	75.5	16.98	11.84	0.922 (Large)	0.419
3	50.0	59.5	21.1	20.39	0.458 (Medium)	0.223
4	61.25	65.0	30.9	34.20	0.115 (Small)	0.057
5	47.5	59.5	9.72	12.50	1.072 (Large)	0.472

increase corresponding to Lab-2 is significant, i.e., 45%. Improvements of 22% and 15% can be noted in the Lab-3 and Lab-5, respectively.

D. Effect Size—Statistical Analysis

Analysis of the prelab/postlab assessment data based on the ES measure [19] is described next. The ES d and the ES correlation r are computed as follows:

$$\text{Effect size } d = \frac{\mu_2 - \mu_1}{\sigma_{\text{pooled}}}$$

where

$$\sigma_{\text{pooled}} = \sqrt{\frac{(\sigma_1^2 + \sigma_2^2)}{2}} \quad (1)$$

$$\text{ES correlation } r = \frac{d}{\sqrt{d^2 + 4}} \quad (2)$$

where $\{\mu_1, \sigma_1\}$ and $\{\mu_2, \sigma_2\}$ correspond to the mean and standard deviation of prelab and postlab assessment data, respectively. The ES d is the ratio of the difference between the means μ_2 and μ_1 to the pooled standard deviation σ_{pooled} . The ES measure does not describe the test of significance; however, it gives valuable information on how strong the effect of involvement was. Note that, in this case, the effect of involvement corresponds to the degree of student learning attributed specifically to J-DSP. Another ES metric typically used in the statistical analysis is the ES correlation. The ES correlation r measures the degree to which the prelab and the postlab assessment data are related [19]. The square of ES correlation, i.e., r^2 , can be interpreted as the percentage of variance.

All the ES analysis results are summarized in Table IV. From Table IV, the J-DSP Lab-1 through Lab-5 resulted in ES correlations of 0.149, 0.419, 0.223, 0.057, and 0.472, respectively. The corresponding percentages of variances are given by 2.22%, 17.56%, 4.97%, 0.3%, and 22.27%. The ES analysis results presented in Table IV are coherent with the prelab/postlab assessment results shown in Table III and in Fig. 7. Note that J-DSP Lab-2, Lab-3, and Lab-5 resulted in an ES of 0.92, 0.46, and

1.07, respectively. In general, ES values of 0.2, 0.5, and 0.8 relate to a “small,” “medium,” and “large” degree of involvement, respectively [19].

E. J-DSP Assessment and Pedagogy

The pedagogy adopted for the integration of J-DSP laboratories in the course included the following sequence of activities: lecture on the pertinent DSP topic, a prelab quiz associated with the topic, an online laboratory using J-DSP visual programming, and a postlab quiz. The prelab and postlab quizzes are part of both the pedagogy and assessment. The prelab quiz emphasizes the key concepts and provides information to the instructor and students on deficiencies and on areas that require further review. The J-DSP laboratory includes exercises that address these key concepts and enable students to improve their performance and eliminate deficiencies; the postlab quiz measures whether additional knowledge was gained and whether deficiencies have been eliminated by attempting the J-DSP laboratory. The assessment results described in Sections IX-A–C revealed that the J-DSP laboratories have indeed enhanced and reinforced student learning, particularly in topics such as filter design, FFT-based spectral estimation, and multirate signal processing. In particular, the general and concept-specific assessment results (Tables I and II, respectively) support the objectives associated with the J-DSP labware and the pedagogy. The J-DSP simulations involving seamless animations and interactive GUI have shown prominent differences in the prelab/postlab assessment. Moreover, the ES measures (Section IX-D) showed that the effect of J-DSP involvement in student learning is mostly “medium” or “large” (see Table IV). The J-DSP assessment also examines the following:

- 1) the effectiveness and utility of curricular materials;
- 2) student performance outcomes and attitudes;
- 3) faculty reaction and perceptions;
- 4) the program impact.

A final, summative J-DSP software and laboratory evaluation provides prescriptive recommendations concerning strengths, replication, and sustainability.

F. The Impact of J-DSP on the EEE 407 Course

One of the very basic problems in DSP classes is the overall perception and attitudes that the students have toward the class material. Past evaluations have shown that the class is perceived by students as too mathematical. For example, engineering applications of DSP were not immediately evident to the average student. The use of J-DSP has impacted the EEE 407 course in the following ways.

- The J-DSP exercises and software have changed the attitudes of the students toward the class, resulting in increased enrollments.
- The use of the software gave the students hands-on experiences with signals and systems. Many concepts have become easier for the students to understand through J-DSP visualizations. In fact, the J-DSP visualizations involving PZ diagrams have shown prominent differences in prelab and postlab assessments that motivated even further integration of several seamless animations and interactive demonstrations [20].
- J-DSP has provided the students with opportunities to study open-ended problems with multiple solutions. Interviews with students revealed that the J-DSP laboratory experiences have instilled in them the process of scientific inquiry, i.e., the ability to pose questions on emerging technologies and the eagerness to seek additional information beyond the course material.
- Exposure to appealing applications, such as cellular phone and MP3 technologies through J-DSP speech and audio processing functions, enabled students to connect class theory with modern technology. In fact, the two last lectures are now spent on LPC and its use in digital cellular phones; these lectures are supported by J-DSP functions and demos.

X. CONCLUDING REMARKS

The online Java–DSP (J-DSP) educational software tool and a detailed assessment of its use in an undergraduate digital signal processing (DSP) class was described in this paper. J-DSP is a scalable object-oriented environment that enables students to establish and run educational DSP simulations on the Web, such as digital filter design, signal analysis, speech signal processing, and fast Fourier transform (FFT)-based spectral analysis.

A. J-DSP Assessment Versus Course Objectives—Summary

A discussion that correlates the J-DSP assessment results (in Section IX) with the EEE 407 course objectives (in Section VIII-A) and the evaluation criteria is presented next. Some of the criteria employed to study the success/failure associated with the J-DSP simulation environment and, in particular, the graphical user interface (GUI) include usability, simplicity, and interactivity. These criteria are evaluated using the general assessment (Section IX-A) instrument. Students asserted that they benefited from J-DSP and appreciated that the tool is freely available on the Web [20]. Industry-affiliated students, taking the course from distance-learning sites, cited the importance of J-DSP in providing online computer laboratory experiences to remote participants. The assessment data given in Table I

further evaluates the aforementioned criteria and supports the J-DSP objectives described in Section I.

Another important criterion employed is to test whether J-DSP functions have been useful in communicating key DSP concepts. In particular, this criterion analyzes directly the success/failure associated with the EEE 407 course objectives listed in Section VIII-A. For example, 89% of the students better understood the concepts of FIR and IIR filter design after performing the J-DSP laboratories (good support for one of the course objectives). More concept-specific assessment (Section IX-B) results given in Table II further evaluates this criterion and supports the EEE 407 course objectives. The percentage improvement in average student scores (Table III) after performing J-DSP laboratories supports the claim that the J-DSP has indeed attributed constructively in the student learning process. These results are further confirmed statistically using the effect size (ES) measures and metadata analysis (Table IV).

B. Usability and ABET

All the developed J-DSP software, laboratory exercises, J-DSP manuals, and support content associated with the J-DSP project are freely accessible from <http://jdsp.asu.edu>. Therefore, adaptation and implementation in DSP courses at other universities is possible. The J-DSP labware has been rigorously tested and disseminated to several universities at various conferences [38]. The University of Central Florida, Orlando, and the University of Texas at Dallas have already developed a pilot exercise with this tool in their signals and systems classes. Furthermore, ASU established resources to maintain the J-DSP software and the supporting website [20] perpetually.

A self-study has been carried out on the EEE 407 DSP course that evaluated how the course and particularly its J-DSP laboratory addresses Accreditation Board for Engineering and Technology (ABET)¹ criteria. With regard to ABET design requirements, students are asked to solve open-ended problems as part of the J-DSP laboratory exercises. The J-DSP laboratory involves extensive filter design using several methods, and students are asked to analyze these filter design methods and perform comparisons. Students are asked to design algorithmic steps and develop their J-DSP realizations. More specifically, the J-DSP laboratory component addresses well ABET Criteria 3 and 8. The J-DSP laboratory tasks in almost all the J-DSP laboratories engage the students in applying their knowledge of mathematics and engineering (ABET Criterion 3a). For example, in J-DSP Lab-1 and Lab-4, students apply the mathematical properties of z transforms and Fourier transforms to real-world engineering applications, such as filtering and spectral analysis (which relates to ABET Criterion 3a). There are specific laboratories that involve filter design (J-DSP Lab-3), where the student is required to use J-DSP to perform several designs and compare and choose the one that satisfies specific criteria (ABET Criterion 3b). ABET Criterion 3c relates to J-DSP laboratory design tasks, such as the design of linear phase filters (J-DSP Lab-3), digital oscillators (J-DSP Lab-1), and QMF filter banks (J-DSP Lab-5). Several engineering

¹<http://www.abet.org/>

problems are solved within the context of J-DSP laboratories relating to ABET Criterion 3e. J-DSP tasks also satisfy ABET Criterion 8 since discrete mathematics and linear algebra are inherent to DSP.

C. Interdisciplinary Extensions and Tools

The work on this National Science Foundation (NSF) Course-Curriculum and Laboratory Improvement (CCLI) J-DSP project is ongoing, and several new functions are being developed. Such functions involve advanced DSP functionality for graduate classes in speech processing [27], [35] and functions that support topics in analog/digital communications [30], image processing [29], audio effects [32], bio-informatics [37], and controls [31]. Infrastructure enhancements include a new GUI that allows integration of J-DSP functionality with streaming video/audio over MS PowerPoint that is being developed. A summary of some of the key interdisciplinary areas and tools supported by J-DSP is included below.

- 1) *Interdisciplinary areas and courses*: Some of the interdisciplinary areas targeted by J-DSP are given by psychoacoustics, DNA gene sequencing/bio-design, imaging and computer vision, and human auditory psychophysics. The J-DSP tool has also been used in several areas and courses outside the DSP arena (at ASU), for example, in communications [30] and control systems [31].
- 2) *Use of J-DSP in high schools*: In an effort to introduce high school (HS) students to DSP and multimedia, several HS friendly and appealing functions [e.g., bass/treble controls, music tone generator, dual-tone multiple-frequency (DTMF), and graphic equalizer] have been carefully designed [32].
- 3) *Multimedia, MP3, and cellular telephony*: J-DSP also provides a unique tool called Appealing-Application Assembler (AAA). In particular, the AAA tool helps students to design and build from a preliminary set of J-DSP modules a practical application (e.g., linear prediction-based speech analysis as used in cellular telephony, speech morphing applications, and MP3 players).

D. Future Directions

- 1) *Distributed computing*: The inherent Java backbone and the object-oriented approach of J-DSP allow students from various university sites to collaborate and exchange ideas. This tool is in its testing phase and targets primarily the senior-level design projects.
- 2) *NSF-funded combined educational materials development (EMD)/dissemination project*: This project also extends J-DSP for use in computer science and biomedical engineering, and it is collaborative among the University of Washington, the University of Texas at Dallas, the University of Rhode Island, and Arizona State University.
- 3) *Combined Research and Curriculum Development (CRCD)*: J-DSP is also used in an NSF-funded interdisciplinary CRCD program that aims to provide both scientific and investigative experiences to undergraduate (UG) students in various areas.

ACKNOWLEDGMENT

The authors would like to thank all the students of the EEE 407 DSP class for their valuable feedback on J-DSP and the associated online laboratories. The authors also would like to thank their graduate research associates and other faculty that contributed to this project, as well as the anonymous reviewers for their helpful comments that improved the clarity of the paper.

REFERENCES

- [1] E. A. Lee, "Overview of the Ptolemy project," Univ. of California, Berkeley, CA, Tech. Memo. UCB/ERL M03/25, Jul. 2003.
- [2] S. C. Douglas, G. C. Orsak, and M. A. Yoder, "DSP in high schools: New technologies from the Infinity Project," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing (ICASSP)*, vol. 4, May 2002, pp. 4152–4154.
- [3] Speech Recognition Java Demos. [Online]. Available: <http://www.isip.msstate.edu/projects/speech/index.html>
- [4] The Connexions Project. Rice Univ., Houston, TX. [Online]. Available: <http://cnx.rice.edu/>
- [5] J. H. McClellan, R. W. Schafer, and M. A. Yoder, *DSP First: A Multimedia Approach*. Upper Saddle River, NJ: Prentice-Hall, Dec. 1998.
- [6] J. H. McClellan et al., *Computer-Based Exercises for Signal Processing Using MATLAB ver.5*. Upper Saddle River, NJ: Prentice-Hall, Oct. 1997.
- [7] S. H. Ardalan and T. Adali, "DSP and communication engineering education with graphical block diagram simulation tools," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing (ICASSP)*, vol. 6, Apr. 1994, pp. 57–60.
- [8] D. A. Pierre, "Supplementary MATLAB tools for systems and control education," in *Proc. 26th IEEE Frontiers in Education Conf. (FIE)*, vol. 3, Nov. 1996, pp. 1215–1218.
- [9] B. Ando, S. Graziani, and N. Pitrone, "Stand-alone laboratory sessions in sensors and signal processing," *IEEE Trans. Educ.*, vol. 47, no. 1, pp. 4–9, Feb. 2004.
- [10] D. J. Brown, M. Covington, and M. L. Swafford, "Mallard: An educational tool for digital signal processing," in *Proc. 30th ASILOMAR Conf. Signals, Systems, Computers*, vol. 1, Nov. 1996, pp. 231–235.
- [11] M. D. Zoltowski, J. P. Allebach, and C. A. Bouman, "Digital signal processing with applications: A new and successful approach to undergraduate DSP education," *IEEE Trans. Educ.*, vol. 39, no. 2, pp. 120–126, May 1996.
- [12] J. Jackson, T. Barnwell, D. Williams, M. Hayes, D. Anderson, and R. Schafer, "DSP for practicing engineers: An online course for continuing DSP education," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing (ICASSP)*, vol. 5, May 2001, pp. 2721–2724.
- [13] P. Hong, D. Anderson, D. Williams, J. Jackson, and T. Barnwell et al., "DSP for practicing engineers: A case study in Internet course delivery," *IEEE Trans. Educ.*, vol. 47, no. 3, pp. 301–310, Aug. 2004.
- [14] M. H. Shor, "Remote-access engineering educational laboratories: Who, what, when, where, why, and how?," in *Proc. Amer. Control Conf.*, vol. 4, Jun. 2000, pp. 2949–2950.
- [15] H. Shen et al., "Conducting laboratory experiments over the Internet," *IEEE Trans. Educ.*, vol. 42, no. 3, pp. 180–185, Aug. 1999.
- [16] D. Gillet, O. D. Crisalle, and H. A. Latchman, "Web-based experimentation integration in engineering curricula: Experience in deploying resources distributed among universities," in *Proc. 34th Southeastern Symp. System Theory*, Mar. 2002, pp. 66–70.
- [17] A. Spanias, "Speech coding: A tutorial review," *Proc. IEEE*, vol. 82, no. 10, pp. 1541–1582, Oct. 1994.
- [18] T. Painter and A. Spanias, "Perceptual coding of digital audio," *Proc. IEEE*, vol. 88, no. 4, pp. 451–515, Apr. 2000.
- [19] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*, 2nd ed. Hillsdale, NJ: Lawrence Erlbaum Assoc., 1988.
- [20] The J-DSP Web Page. Arizona State Univ., Tempe. [Online]. Available: <http://jdsp.asu.edu>
- [21] A. Clausen, A. Spanias, A. Xavier, and M. Tampi, "A Java signal analysis tool for signal processing experiments," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing (ICASSP)*, vol. 3, May 1998, pp. 1849–1852.
- [22] A. Clausen, T. Painter, and A. Xavier et al., "J-DSP: An Internet-based educational tool for digital filter experiments," in *Proc. IEEE Symp. Advances Digital Filtering Signal Processing*, Jun. 1998, pp. 57–61.

- [23] A. Spanias *et al.*, "Development and evaluation of a Web-based signal and speech processing laboratory for distance learning," *ASEE Comput. Educ. J.*, vol. X, pp. 21–26, Apr.–Jun. 2000.
- [24] A. Spanias and F. Bizuneh, "Development of new functions and scripting capabilities in Java-DSP for easy creation and seamless integration of animated DSP simulations in Web courses," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing (ICASSP)*, vol. 5, May 2001, pp. 2717–2720.
- [25] A. Spanias *et al.*, "On-line laboratories for speech and image processing and for communication systems using J-DSP," in *Proc. 10th DSP Workshop 2nd Signal Processing Education Workshop*, Oct. 2002, pp. 174–179.
- [26] A. Spanias, K. Ahmed, A. Papandreou-Suppappola, and M. Zaman, "Assessment of the Java-DSP (J-DSP) on-line laboratory software," in *Proc. 33rd IEEE Frontiers in Education Conf. (FIE)*, vol. 1, Nov. 2003, pp. T2E 16–T2E 21.
- [27] V. Atti and A. Spanias, "On-line simulation modules for teaching speech and audio compression," in *Proc. 33rd IEEE Frontiers in Education Conf. (FIE)*, vol. 1, Nov. 2003, pp. T4E 17–T4E 22.
- [28] M. Zaman, A. Papandreou-Suppappola, and A. Spanias, "Advanced concepts in time–frequency signal processing made simple," in *Proc. 33rd IEEE Frontiers in Education Conf. (FIE)*, vol. 1, Nov. 2003, pp. T2E 10–T2E 15.
- [29] M. Yasin, L. Karam, and A. Spanias, "On-line laboratories for image and two-dimensional signal processing," in *Proc. 33rd IEEE Frontiers in Education Conf. (FIE)*, vol. 1, Nov. 2003, pp. T3E 19–T3E 22.
- [30] Y. Ko, T. Duman, and A. Spanias, "On-line laboratory for communication systems using J-DSP," in *Proc. 33rd IEEE Frontiers in Education Conf. (FIE)*, vol. 1, Nov. 2003, pp. T3E 13–T3E 18.
- [31] T. Thrasyvoulou, K. Tsakalis, and A. Spanias, "J-DSP-C, A control systems simulation environment: Labs and assessment," in *IEEE Frontiers in Education Conf. (FIE)*, vol. 1, Nov. 2003, pp. T4E 11–T4E 16.
- [32] A. Spanias *et al.*, "Using J-DSP to introduce communications and multimedia technologies to high schools," in *Proc. 33rd IEEE Frontiers in Education Conf. (FIE)*, vol. 2, Nov. 2003, pp. F3A 22–F3A 27.
- [33] V. Atti, A. Spanias, C. Panayiotou, Y. Song, and Y. Ko, "On the use of J-DSP for on-line laboratories in linear systems courses," in *Proc. 37th ASILOMAR Conf. Signals, Systems, Computers*, vol. 2, Nov. 2003, pp. 1375–1379.
- [34] A. Spanias, V. Atti, A. Papandreou-Suppappola, K. Ahmed, M. Zaman, and T. Thrasyvoulou, "On-line signal processing using J-DSP," *IEEE Trans. Signal Process. Lett.*, vol. 11, no. 10, pp. 1–5, Oct. 2004.
- [35] V. Atti and A. Spanias, "Web-based experiments for introducing speech recognition basics in a DSP course," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing (ICASSP)*, vol. 5, May 2004, pp. 1045–1048.
- [36] A. Spanias, C. Panayiotou, T. Thrasyvoulou, and V. Atti, "Java–DSP interface with MATLAB and its use in engineering education," presented at the Amer. Soc. Engineering Education (ASEE) Conf., Salt Lake City, UT, Jun. 20–23, 2004.
- [37] A. Spanias *et al.*, "Teaching genomics and bioinformatics to undergraduates using J-DSP," presented at the Amer. Soc. Engineering Education (ASEE) Conf., Salt Lake City, UT, Jun. 20–23, 2004.
- [38] The J-DSP Dissemination Web Page. Arizona State Univ., Tempe. [Online]. Available: http://jdsp.asu.edu/jdsp_dissemination.html
- [39] Y. Song, A. Spanias, V. Atti, and V. Berisha, "Interactive Java modules for the MPEG-1 psychoacoustic model," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing (ICASSP)*, vol. 5, Mar. 2005, pp. 581–584.

Andreas Spanias (S'84–M'85–SM'94–F'00) is a Professor in the Department of Electrical Engineering, Ira A. Fulton School of Engineering, of Arizona State University (ASU), Tempe. While at ASU, he has developed and taught courses in digital signal processing (DSP), adaptive signal processing, and speech coding. He has also developed and taught continuing-education short courses and Web courses in DSP and speech coding. He is Associate Director of the ASU Arts, Media, and Engineering (AME) program, where he heads a program on sound localization for smart stages using microphone arrays. He has been the Principal Investigator on research contracts from Intel Corporation; Sandia National Laboratories; Motorola, Inc.; the National Science Foundation; and Active Noise and Vibration Technologies. He has also served as a Consultant with Inter-Tel Communications, Intel Corporation, Motorola, Texas Instruments, DTC, and the Cyprus Institute of Neurology and Genetics. In his work with Intel Corporation, he contributed to the development of architectures with signal processing capabilities. He and his student team developed the computer simulation software Java-DSP (J-DSP-ISBN 0-9724 984-0-0), which is being used in the ASU DSP courses. His research interests are in the areas of adaptive signal processing and speech processing.

Dr. Spanias is a Member of Eta Kappa Nu and Sigma Xi. He is involved extensively in IEEE scientific activities. He is a Member of the DSP Committee of the IEEE Circuits and Systems Society and has served as a Member of the Technical Committee on Statistical Signal and Array Processing of the IEEE Signal Processing Society (SPS). He was a General Co-Chair of the 1999 International Conference on Acoustics Speech and Signal Processing (ICASSP'99) in Phoenix, AZ, and served as the IEEE Signal Processing Vice-President for Conferences and the Chair of the Conference Board. He has been Chair of the IEEE Communications and Signal Processing Chapter in Phoenix, AZ. He served as a Member of the IEEE Signal Processing Executive Committee, the IEEE SPS Publications Board and as Associate Editor of the IEEE SIGNAL PROCESSING LETTERS and the IEEE TRANSACTIONS ON SIGNAL PROCESSING. He is Member-at-Large of the IEEE SPS Board of Governors. In his work with Intel Corporation, he received an award from Intel-Chandler citing "his leadership and contributions to the development of the Intel 60172 processor architecture." He also received the 2003 Teaching Award from the IEEE Phoenix Section for the development of J-DSP. He is corecipient of the 2002 IEEE Donald G. Fink Paper Prize Award and was appointed Distinguished Lecturer of the IEEE SPS for 2004. He received the Meritorious Service Award from the IEEE SPS at ICASSP 2005.

Venkatraman Atti (S'00) received the B.Tech. degree in electronics and communications engineering from S.V. University College of Engineering, Tirupati, India, in 2000 and the M.S. degree in electrical engineering from Arizona State University (ASU), Tempe, in 2002. He is currently working toward the Ph.D. degree in electrical engineering at ASU.

From 2000 to 2004, he worked as a Research Associate on development of Java-based educational software, i.e., J-DSP, for speech coding techniques with complete simulations and online laboratory curriculum. He was also a Software Team Leader in the award-winning J-DSP project (<http://jdsp.asu.edu>). His current research interests are in the areas of low-bit-rate/low-delay speech and audio coding.

Mr. Atti is a Member of the IEEE Signal Processing Society and a Member of the Audio Engineering Society. He was recently nominated for the Best Student Paper Award at the 2005 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP'05), Philadelphia, PA, for his Ph.D. research on "speech analysis by estimating perceptually relevant pole locations." He also received the prestigious Viswanath Memorial Award for his academic excellence in his undergraduate work.