# THE JAVA-DSP (J-DSP) PROJECT – FROM THE PROTOTYPE TO THE FULL IMPLEMENTATION AND DISSEMINATION[*]

Andreas Spanias and Venkatraman Atti

Department of Electrical Engineering, Ira A. Fulton School of Engineering,
Arizona State University, Tempe, AZ 85287-5706, USA.
e-mail: [spanias, atti]@asu.edu

## Abstract

This paper describes the roadmap followed to develop a prototype, obtain support for its full implementation, embed in the curriculum and assess, and then disseminate. We outline specifically how the J-DSP online laboratory concept was conceived, how the prototype "proof of concept" version developed, how a more elaborate version was embedded in the curriculum, and the way it was eventually disseminated. Even though the paper describes our experiences in the signal processing area the idea can be extended to other areas as well.

***Index Terms*** – Web-based instruction, on-line laboratories, interactive DSP simulations, assessment instruments, J-DSP dissemination, FIE workshop, and CCLI EMD proposal

## 1. Introduction

What motivated us initially to pursue the creation of software for online laboratories was simply the emergence of the Internet technologies as a force that would shape distance learning programs and undergraduate education as a whole. In the early to mid 1990s, just about everyone in the education community regardless of the area was talking about possibilities for web courses and web education.

[*] The project is being funded by the NSF CCLI EMD program # 0443137.

Some faculty already were placing select content on the web to support existing courses and some of the more daring proceeded with creating a web version of their course. Everyone in the faculty and administration circles was in search of new authoring tools for courseware. We applied and obtained some seed funding from the ASU research office to look at various web technologies and experiment with HTML and Java™. We were particularly keen in using Java.

## 2. Identifying a Problem in Engineering Education

Much like any research endeavor we thought of this initially as a research problem, i.e., given that Internet was evolving and would affect education, what are the key engineering education problems that would surface and requite innovative solutions. We wanted to do something different, something that was not available, and something that would ultimately be useful to the education community. At that time we identified that technologies for creating and disseminating web courses were essentially available but how to deliver online laboratory experiences was not evident. So we defined our problem theme as "the need to create technologies and/or software for delivering online laboratory experiences to distance learners."

### 2.1. Preliminary Solution

We knew that the solution, at least in our case, had to be software and we quickly became aware of Java and its capabilities. The first Java animations and visualizations were emerging on several web sites. We could not simply produce animations for labs; instead we needed an environment that supports online simulations of signals and systems and signal processing. The idea that emerged between the PI and his associates was to create a whole programming environment on the web; it had to be easy to use and intuitive, it had to be universally accessible with minimal resources, it had to be adequate for scientific programming. All these ideas converged in an environment supported by Java, that was object-oriented and allowed a block diagram programming approach.

### 2.2. Forming a Team and Seeking Seed Funding for a Prototype

We identified first local university programs that would provide seed funding. We formed a faculty team and wrote a 4 page proposal that gave as about $50k to work on a prototype. By the end of the summer of 1997, we were able to demonstrate the prototype that we called Java-Digital Signal Processing – abbreviated as J-DSP. We have demonstrated this prototype to several administrators and colleagues. Although they were not quite sure what it was, they were intrigued by the ability to run simulations on the Internet. The bandwidth needed to support it was small, anything "Java" attracted attention, and we convinced our chair to let us run a pilot study in the DSP class. We produced sample exercises and our first integrated simulation with animation; it was a success and got thumbs up from students.

### 2.3. Use of Prototype in the DSP Course and in Short Courses

Our next step was to create four laboratories for use in a DSP course. We created labs on: z transforms, frequency responses, FFT, and FIR filter design. After we got permissions, these four labs were featured in the DSP undergraduate course and shorter versions were given in short courses. Preliminary assessment instruments were disseminated. The most important part from this assessment was feedback on software bugs. A paper was submitted to IEEE ICASSP 98 in Seattle that featured signal processing education sessions.

## 3. Proposals to NSF CCLI EMD

Our first proposal to NSF CCLI was prepared and submitted in 1998. Unfortunately, it did not receive favorable reviews and was not funded. We obtained the reviews and although our initial feeling was frustration, we took the suggestions of the reviewers and developed a revised proposal that was resubmitted in June 2000. The result from the second submission was favorable and we were funded for three years starting 2001. This was very important for the program. We hired very good students – some of them were multitalented and helped us develop good publicity materials. The funded program was NSF CCLI 0089075: "On-line UG Labs in Signal/Image Processing, Communications, and Controls," PI: A. Spanias, Co-PIs: Karam, Duman, Papandreou-Suppappola, Tsakalis, $424,770, Jan. 01-Dec. 03 (no cost extension 04). In this project, we developed the J-DSP software [5] [13] [14]. J-DSP and a set of web-based laboratories were used in DSP, Communications [10], and Controls [11] courses. On-line assessment has been documented [6]. By-products include tools for embedding animations in web content [3]. J-DSP was rated as one of the top three educational resources for 2003 by the NEEDS panel and was recognized as such at the IEEE/ASEE FIE 2003; J-DSP was also recognized by the IEEE Phoenix section.

## 4. The J-DSP Simulation Tool

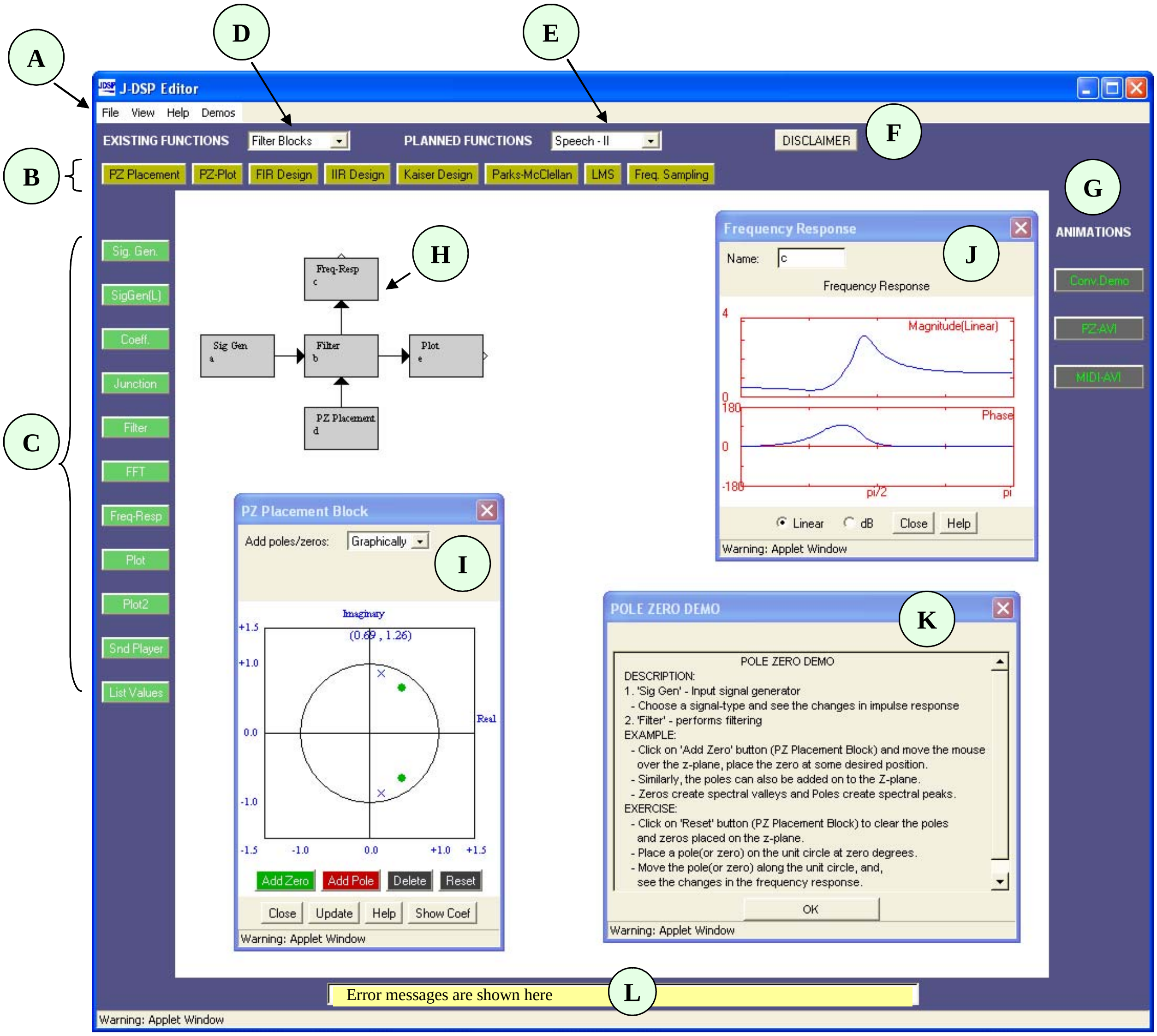
The J-DSP editor [2] [3] [13] is an object-oriented DSP simulation environment built from the ground up for education. J-DSP can be accessed using a Java-enabled browser at **http://dsp.asu.edu**. J-DSP has a rich suite of signal processing functions that facilitate interactive on-line simulations of 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), time and frequency response plots, etc. Advanced functions include digital filter design, statistical signal processing, spectral analysis, multi-rate DSP and quadrature mirror filter (QMF) banks, linear predictive coding, hidden Markov model (HMM) estimation, and blocks that support speech recognition [14].

Recent extensions on the software include AM/FM modulation [10], 2-D filters and 2-D transforms [9], spectrograms [8], speech coding [7], controls [11]. Also included are functions that allow the user to import and export signals, playback sound, display processed images, etc. All DSP signal manipulation functions appear in J-DSP as blocks that are brought in the simulation environment by a drag-and-drop process. By connecting blocks together signal flow is established and a program executes. J-DSP blocks can be designed to perform animations, create html embeddable demos, perform numerical computation, etc. As such J-DSP lends not only to on-line platform-independent laboratories but also for creating interactive web content and embedding it in web courses. Researchers at Arizona State have developed capabilities for scripting, developed a preliminary interface to MATLAB, and maintained the software so that it is compatible with the latest browser technology. Although commercial object-oriented DSP software has been developed by several software vendors, e.g., Simulink®, SPW®, LabView® etc., much of it is expensive and platform-specific. J-DSP allows free access and represents a different paradigm for education where simulation is done over the internet from any platform. The scope of J-DSP is not to provide tools for comprehensive numerical simulations but instead to consider ideas motivated by education and create functions that enhance the educational experience. J-DSP was in fact exclusively developed having education in mind. Integration of the J-DSP editor with web material is virtually seamless [3]. In fact the script functionality provides instructors with capabilities to establish and run customized visualization modules without engaging into tedious low-level Java programming.

The J-DSP laboratory software and exercises address the following objectives: i) provide on-line laboratory experiences to DSP students in Electrical Engineering; ii) enable programmable Internet simulations with embedded animations; iii) accelerate learning by exposing students to hands-on manipulation of signals and systems; and iv) provide intuition and complementary information usually not available in lectures and text books. To accomplish these objectives, pedagogical structures have been formed that are described in [14]. In addition, an assessment process has been designed to measure: a) the progress towards accomplishing these objectives, b) the effectiveness of the pedagogies adopted, and c) the overall impact of the software tool on an undergraduate course.

### 4.1. The J-DSP Editor

J-DSP provides a user-friendly environment that exploits the graphical capabilities of Java. Figure 1 shows the J-DSP simulation area. In this figure, section '*A*' represents menu items and section '*B*' corresponds to a *floating panel*. Note that 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").



POLE ZERO DEMO
DESCRIPTION:
1. 'Sig Gen' - Input signal generator
- Choose a signal-type and see the changes in impulse response
2. 'Filter' - performs filtering
EXAMPLE:
- Click on 'Add Zero' button (PZ Placement Block) and move the mouse over the z-plane, place the zero at some desired position.
- Similarly, the poles can also be added on to the Z-plane.
- Zeros create spectral valleys and Poles create spectral peaks.
EXERCISE:
- Click on 'Reset' button (PZ Placement Block) to clear the poles and zeros placed on the z-plane.
- Place a pole(or zero) on the unit circle at zero degrees.
- Move the pole(or zero) along the unit circle, and, see the changes in the frequency response.

Figure 1. J-DSP™ simulation environment

Each block is linked to software that performs a specific signal processing function. Block parameters can be edited through the corresponding dialog windows. For example, '*I*' represents the dialog window corresponding to the *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, which means 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 in order 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 trouble-shooting and exception handling features.

## 5. Assessment of J-DSP as Used in the ASU EEE 407 DSP Labs

Assessment of the J-DSP laboratory concept was done with on-line instruments generated with the help of an ASU assessment specialist. Various forms of assessment have been performed including general assessment of the software, ergonomics (accessibility, user friendliness), concept specific assessment, and qualitative assessment. Some of the results of this continuing assessment have been presented in [6] [13]. In summary, from two DSP class samples of 93 and 96 students, 95% of the students responded in favor of internet-based labs, and 70% stated that it took them less than 10 minutes to learn to use J-DSP. Sample results from concept-specific assessment showed that, on the average, 85% of the students agreed that J-DSP helped them comprehend specific DSP concepts that were not evident from the lectures and text book.This was particularly the case when the Java simulation included animations. The *pre/post-lab* quiz showed a 24% average improvement specifically attributed to J-DSP; in some specific cases associated with z-plane simulations *pre/post-lab* yielded improvements of about 65%. More statistical analysis on the assessment results is included in [13] [14]

## 6. Preliminary Dissemination

In a J-DSP software dissemination contacted in the 2002 IEEE/ASEE Frontiers in Education and the 2002 IEEE DSP workshop, we have obtained agreements for beta sites from several faculty including: RICE university, MIT, University of Puerto Rico, Georgia Tech, and about twenty other universities. This dissemination was done by organizing publicity materials and renting an exhibits booth at FIE 2002. It was very important to disseminate this version of J-DSP as it made the program known to several of our colleagues.

### 6.1. Web Site, Publicity, and Dissemination at IEEE FIE, ICASSP, and ASEE Conferences

The J-DSP website, i.e., http://jdsp.asu.edu, featured all the latest developments, a manual, AVI demos, etc. As part of our dissemination and testing activities, we have already contacted and obtained preliminary agreements from faculty in several undergraduate programs as well as engineers in relevant fields in the industry. Universities that have already agreed to evaluate the modules and all relevant materials include:

- Georgia Tech
- University of Southern California
- University of Maryland
- University of Minnesota
- University of New Mexico
- University of Texas-Austin
- University of Texas-Dallas
- University of Central Florida
- Northeastern University

We also have agreements with several other universities that have previously obtained the J-DSP software and agreed to evaluate future updates. Universities that agreed to establish J-DSP web sites and test lab content include:

- University of Kent
- Marquette University
- Stevens Institute of Technology
- Georgia Institute of Technology
- Blekinge Institute of Technology
- Drexel University
- University of Nebraska
- Cal Poly Pomona
- University of Detroit-Mercy
- University of Pennsylvania
- Rice University
- Massachusetts Institute of Technology
- University of Akron
- University of Connecticut
- University of Puerto Rico
- Clemson University
- North Carolina State University
- Ecole Nationale
- Polytechnique-Algeria
- Bogazici University-Turkey

## 7. A Collaborative Research Proposal on J-DSP with Other Universities

A new full-scale EMD proposal on a collaborative project involving five universities, namely, Arizona State University (ASU), the University of Washington Bothell (UWB), the University of Texas at Dallas

(UTD), the University of Rhode Island (URI), and the University of Central Florida (UCF) was submitted in June 2004. The proposed project has significant educational technology innovations and software extensions that enable J-DSP [4] [5] [7]-[12] to be used in four courses at the five different universities. The proposal tasks consisted of the following important components: educational innovation achieved by generating a large volume of new Java software that upgrades considerably the J-DSP graphical user interface (GUI), a software development task to extend the mathematical and signal processing functionality of J-DSP so that it can support on-line computer laboratories in four courses, a comprehensive on-line laboratory exercise development task engaging all the Co-PIs at the different universities (see Fig. 2).

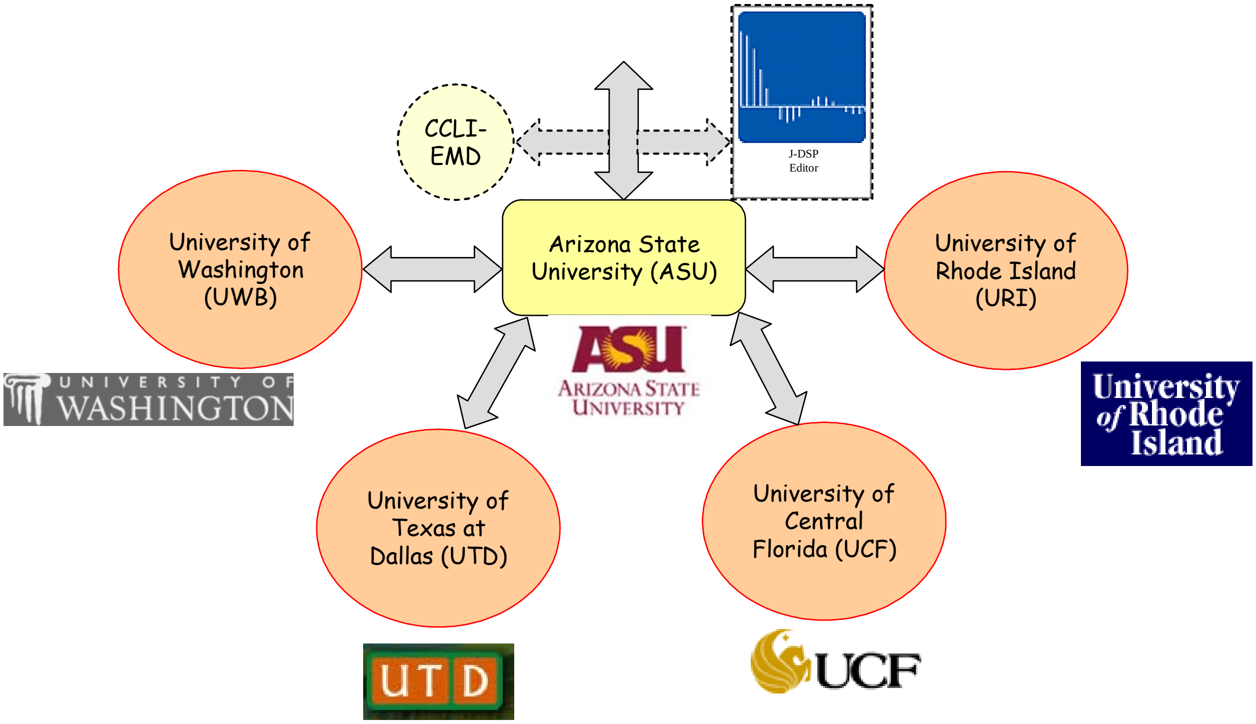

CCLI-EMD
J-DSP Editor
University of Washington (UWB)
Arizona State University (ASU)
University of Rhode Island (URI)
ASU
Arizona State University
University of Washington
University of Rhode Island
University of Texas at Dallas (UTD)
University of Central Florida (UCF)
UTD
UCF

Figure 2. CCLI EMD collaborative research

## 8. Book that Features Theory and J-DSP Labs

The J-DSP book is intended primarily for use in an undergraduate DSP class to complement the theory of signal processing. Although the chapters are populated mostly by J-DSP laboratory exercises, every chapter starts with a compact description of the necessary theory and equations. The first chapter presents a review of linear signals and systems followed by a simple J-DSP laboratory that introduces the students to the J-DSP environment and exposes them to the basics of processing discrete-time signals. The second chapter covers the basics of digital filters and exposes students to the notion of the frequency response. J-DSP is used to simulate several simple digital filters and visualize their time- and frequency responses. The third chapter is on z transforms and the associated J-DSP exercise exposes students to transfer functions, z transform properties, and various filter types. The fourth and fifth chapters deal with digital filter design. FIR and IIR filter design J-DSP exercises helps students understand some of the properties of FIR linear phase design as well as those of IIR analog filter approximations. The sixth chapter is on the basics of multi-rate signal processing. The associated J-DSP exercise exposes the students to the fundamental rules of up-sampling and down-sampling and to the basics of QMF banks. The seventh chapter deals with the utility of the FFT in signal processing. The FFT J-DSP exercise focuses on spectral leakage effects and the use of windows and zero padding. The eighth chapter is on the basics of discrete-time random signal analysis. The J-DSP exercise on random signals focuses on correlogram, periodogram, and parametric AR spectral estimators.

## 9. Faculty Workshop at FIE 2005 Conference: “Designing Laboratories, Exercises, and Demos using the Java-DSP Laboratory Software in Signals and Systems Courses”

This workshop will expose participants to the utility of the ASU Java-DSP software technology in Signals and Systems courses. The session will be interactive and participants will use and assess an exercise that involves step-by-step online simulations on prepackaged online laboratory exercises that can be disseminated in their junior level signals and systems classes. Assessment forms and pre- and post-evaluation quizzes will be discussed.

### 9.1. Workshop Activities and Objectives

This workshop will include tutorials and examples to demonstrate how instructors can use J-DSP in their classes both as a demonstration tool and as a tool that enables them to provide laboratory experiences to on-campus and distance learning students. The session will be conducted as a computer workshop and participants will attempt to use, program, and execute J-DSP-based exercises and scripts. The objectives are:

- to learn how to use Java-DSP. Documentation will be given to get participants started.
- to learn how to use the signal generator and filter design functions. Programming J-DSP demos to help students understand linear system and filtering concepts
- to learn how to use the FFT to compute spectra of signals. Programming demos to help students understand the Fourier transform properties
- to earn how to use other pre-canned signal functions
- to learn how to use J-DSP scripts to embed demonstrations from web course content
- to enable participants to design their own J-DSP laboratory exercises
- to demonstrate how to carry an assessment of the exercises and practices with J-DSP

### 9.2. Audience

The workshop is intended for Electrical Engineering, Computer Science, and Electromechanical Technology instructors, faculty members, engineers and scientists that are interested in integrating laboratory experiences in their signals and systems and DSP related activities and courses. Although the organizers will bring 4-5 extra laptops, participants are strongly encouraged to being their own laptops.

### 9.3. Expected Outcome

Participants will be able to use the J-DSP software and laboratories, design their own J-DSP based laboratories and exercises, and form and execute an effective evaluation plan of these exercises. Participants will be able to plan an adaptation of these practices and materials to their courses.

## Bibliography

[1] A. Clausen, A. Spanias, A. Xavier, and M. Tampi, “A Java signal analysis tool for signal processing experiments,” in *Proc. of IEEE ICASSP-98*, Vol. 3, pp. 1849-1852, May 1998

[2] A. Spanias, S. Urban, A. Constantinou, M. Tampi, X. Zhang, M. Tampi, C. Stilianou, “Development and evaluation of a web-based signal and speech processing laboratory for

distance learning," in *ASEE Computers in Education Journal*, Vol. X, pp. 21-26, Apr-Jun 2000.

[3] 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. of IEEE ICASSP-01*, Vol. 5, pp. 2717-2720, May 2001

[4] A. Spanias, V. Atti, Y. Ko, T. Thrasyvoulou, M.Yasin, M. Zaman, T. Duman, L. Karam, A. Papandreou-Suppappola, K. Tsakalis, "On-line laboratories for speech and image processing and for communication Systems Using J-DSP", in *Proc. of 10th DSP workshop and 2nd Sig. Proc. Ed. workshop*, pp. 174-179, Oct 13-16, 2002.

[5] The J-DSP™ web-page, MIDL LAB, Arizona State University: http://jdsp.asu.edu

[6] Spanias A., K. Ahmed, A. Papandreou-Suppappola, and M. Zaman, "Assessment of the Java-DSP (J-DSP) On-Line Laboratory Software," in *33rd ASEE/IEEE FIE-03*, Boulder, Nov. 2003

[7] V. Atti and A. Spanias, "On-line simulation modules for teaching speech and audio compression," in *33rd ASEE/IEEE FIE-03*, Boulder, Nov. 2003

[8] M. Zaman, A., and A. Spanias, "Advanced concepts in time-frequency signal processing made simple," in *33rd ASEE/IEEE FIE-03*, Boulder, Nov. 2003

[9] M. Yasin, L. Karam, and A. Spanias, "On-Line laboratories for image and two-dimensional signal processing," in *33rd ASEE/IEEE FIE-03*, Boulder, Nov. 2003

[10] Y. Ko, T. Duman, and A. Spanias, "J-DSP for communications," in *33rd ASEE/IEEE FIE-03*, Boulder, Nov. 2003

[11] T. Thrasyvoulou, K. Tsakalis, and A. Spanias, "J-DSP-C, A control systems simulation environment for distance learning: labs and assessment," in *33rd ASEE/IEEE FIE-03*, Boulder, Nov. 2003

[12] A. Spanias, T. Thrasyvoulou, C. Panayiotou, and Y. Song, "Using J-DSP to introduce communications and multimedia technologies to high schools," in *33rd ASEE/IEEE FIE-03*, Boulder, Nov. 2003.

[13] A. Spanias, V. Atti, A. Papandreou-Suppappola, *et al.*, "On-line signal processing using J-DSP," in *IEEE Trans. Sig. Proc. Letters*, vol. 11, no. 10, pp. 1-5, Oct. 2004.

[14] A. Spanias and V. Atti, "Interactive On-line Undergraduate Laboratories Using J-DSP," submitted to *Trans. on Education for Special Issue on Web-based Instruction*.