# DEVELOPMENT AND EVALUATION OF A WEB-BASED SIGNAL AND SPEECH PROCESSING LABORATORY FOR DISTANCE LEARNING [+]

*Andreas Spanias[*], Susan Urban, Argyris Constantinou, Maya Tampi, Axel Clausen,*
*Xiaopeng Zhang, Jeff Foutz and Georgios Stylianou*

Department of Electrical Engineering and Department of Computer Science & Engineering
Multidisciplinary Initiative on Distance Learning
Arizona State University,  Tempe, AZ 85287-7206, U.S.A.

## ABSTRACT

In this paper, we describe an internet-based signal processing laboratory that provides hands-on learning experiences in distributed learning environments.  The laboratory is based on an object-oriented Java™ tool called Java Digital Signal Processing (J-DSP).   J-DSP has been developed at Arizona State University (ASU) and is being used for a virtual laboratory in a senior-level DSP course. J-DSP is written as a platform-independent Java applet that resides on the web and is thereby accessible by all students through the use of a web browser.  J-DSP has a rich suite of signal processing functions that facilitate interactive on-line simulations of modern statistical signal and spectral analysis algorithms, filter design tools, QMF banks, and state-of-the-art vocoders. J-DSP is accompanied by administrative software tools for secure internet-based lab-report submission and evaluation including servlets for maintaining web-based grade books.  A series of J-DSP laboratory exercises has been developed and delivered using the ASU distance learning facilities. Student evaluations as well as assessments by experts have been compiled and preliminary results are quite encouraging.

## 1. INTRODUCTION

The development of the world wide web (WWW) has had a profound impact in distance learning education. Browsing technologies have enabled asynchronous distance learning education and many universities are eager to expand their clientele by reaching students outside their campus borders. To this date, there are some fine examples of Internet use in engineering education. Examples of using the Internet include web publishing of supplemental course material, concept visualization software, web courses accompanied by streaming audio/video, or even entire degree programs based on WWW courses. In the field of signal processing, several important contributions [1-5] associated with this theme appeared in special sessions organized by the DSP Education committee in previous ICASSPs.  Although there have been several notable attempts to post DSP educational material with interactive visualization applets, there has not been enough done for DSP laboratory experiences for distance learners.

In this paper, we present a comprehensive virtual laboratory environment that was tailored to provide hands-on laboratory experiences to distance learning students over the WWW. The virtual, on-line DSP laboratory described in this paper, consists of the graphical DSP editor, called J-DSP, and a software environment for electronic lab report submission and grading. Students use the J-DSP editor to establish and execute DSP simulations of several important algorithms. The editor includes a suite of built-in signal processing functions ranging from simple signal manipulators to complex filter design functions as well as speech, statistical, and adaptive signal processing algorithms. A suite of laboratory exercises that promotes hands-on DSP learning experiences of distance learners has also been developed for J-DSP. Although the DSP simulation capabilities provided by J-DSP are also typically provided by commercial packages such as MATLAB and SIMULINK[**], J-DSP is platform independent, user-friendly, and universally accessible through the use of common WWW browsers. In fact, J-DSP represents a new paradigm of an object-oriented DSP simulation environment built from the ground up for WWW laboratory education. J-DSP is accompanied by tools that facilitate secure lab-report submission including servlets for maintaining web-based grade books. Lab report submission and grading is achieved through a three-tier Client/Application Server/Database Server model described later in the paper.

## 2. THE J-DSP EDITOR

### 2.1 Overview

J-DSP provides a user-friendly environment through Java's graphical capabilities.  Its highly intuitive graphical user interface (GUI) is easy to understand and students quickly learn to use it with very little assistance.   All functions appear in J-DSP as graphical blocks that are divided into groups according to their functionality.  Selecting and establishing individual blocks can be done by a drag-and-drop process.  Each block is associated with a specific signal processing function.  Figure 1 shows the J-DSP editor environment.  By connecting blocks together, a variety of DSP systems can be simulated.  Signals at any point of a simulation can be examined through the appropriate graphics blocks.  Blocks can be edited through dialog windows, allowing

---

the user to change the corresponding function's parameters to desired values and/or to view results. Blocks can easily be manipulated (i.e. edit, move, delete and connect) within the specified drawing area, using the mouse. System execution is dynamic, which means that any change at any point of a system will automatically take effect in all related blocks. Figure 2 shows a sample simulation with J-DSP. Any number of block windows can be left open to enable viewing results at more than one point in the system.

## 2.2 Supported Features

J-DSP provides a rich set of blocks that can be used to perform lab exercises that are typical in a senior-level undergraduate DSP class. These blocks facilitate signal generation, fundamental DSP arithmetic functions, multirate DSP and QMF, filter design, quantization, speech processing, LPC vocoder functions, spectral estimation, adaptive filtering, and some limited image processing.

### 2.2.1 Signal Generation

The signal generator blocks provide a suite of deterministic and random signals as well as samples of real-life signals that can be linked to algorithms supported by J-DSP. For example, speech, music and other time-domain signals can be generated and linked to other blocks.

### 2.2.2 Fundamental DSP Functions

Basic DSP functions like windowing, filtering, simple signal manipulators (i.e. adders, multipliers) and the forward and inverse fast Fourier transform can be implemented. These functions can be combined to build more complex algorithm simulations.
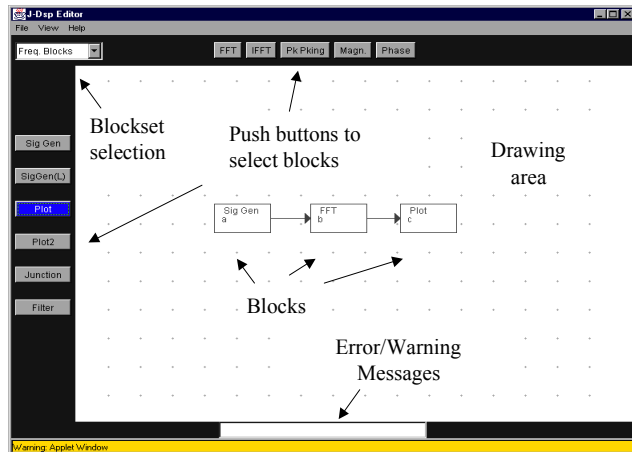


**Figure 1**. J-DSP environment

### 2.2.3 Multirate DSP

There are a number of different DSP applications that make use of multirate signal processing. J-DSP can be used to up-sample or down-sample a signal. This enables building simulations of multirate systems like decimators, interpolators, and even Quadrature Mirror Filter (QMF) banks. In addition, applications of QMF in sub-band coding can be simulated using the quantization block.
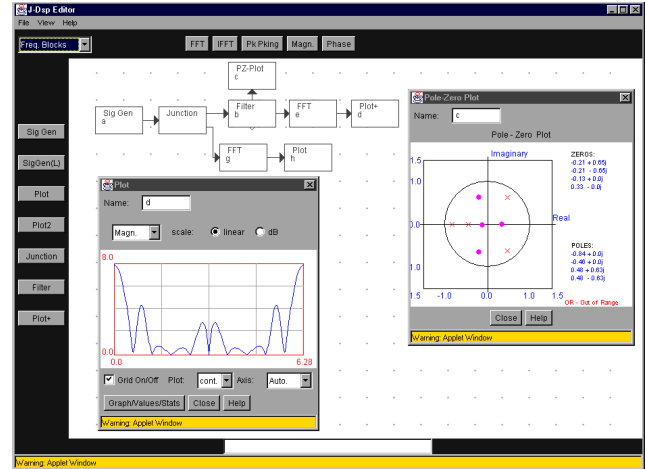


**Figure 2**. Sample simulation
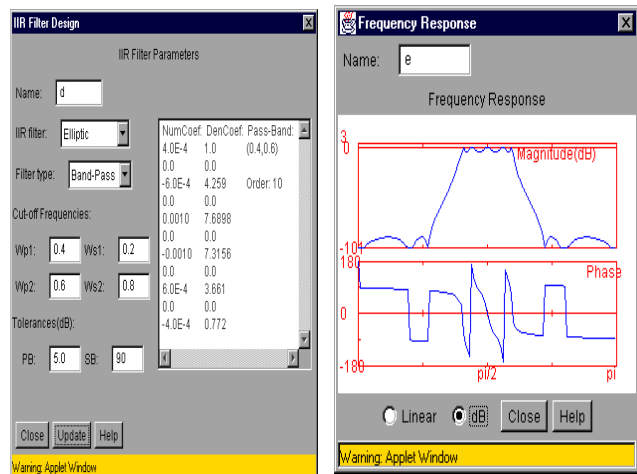


(a)                                    (b)

**Figure 3.** (a) IIR filter design user interface, allowing to enter filter parameters, (b) Frequency response of the filter specified by the parameters in (a)

### 2.2.4 Filter Design

J-DSP supports Infinite Impulse Response (IIR) and Finite Impulse Response (FIR) filter design. Dialog windows are provided for entering the desired filter specifications: cutoff frequencies, filter type (low-pass, pass-band etc.), pass-band and stop-band tolerances. IIR filter design of Butterworth, Chebyshev type I and II, and Elliptic filters is based on the bilinear transformation. FIR filter design is based on the windowing method and supports design with rectangular, Bartlett, Blackman, Hamming, Hanning, and the Kaiser windows.

### 2.2.5 Spectral Estimation

Spectral estimation (SE) techniques such as the classical and parametric techniques have also been implemented in J-DSP. J-DSP supports the two classical estimators, i.e., the correlogram and the periodogram. In addition, J-DSP provides capabilities for parametric Autoregressive (AR) SE. Each SE is implemented as a

separate block. The user can define the number of autocorrelations to be used for computing the correlogram. There is also a lag window block that can be used to control the correlogram estimator resolution and bias. In the periodogram block, the user can choose either the sample spectrum or the Welch method and can perform, Daniell smoothing. An AR parametric SE can also be computed using Linear Prediction (LP) where the LP parameters are computed using the Levinson-Durbin algorithm. The LPC order can be de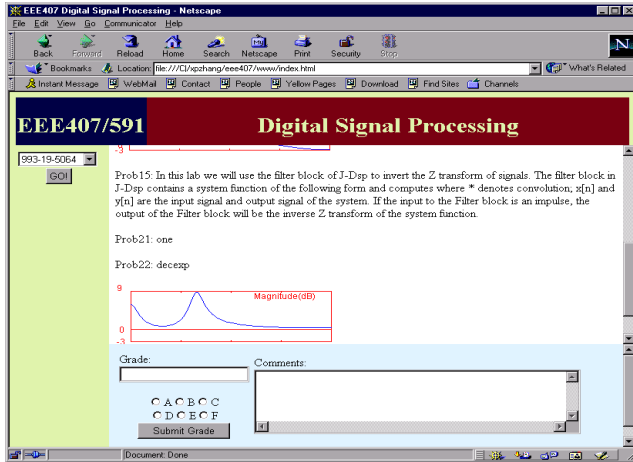fined by the user. All the blocks have a graph area, which displays the estimated power spectral density of the signal. Figure 4 shows a sample spectral estimation simulation.



**Figure 4.** Sample spectral estimation simulation.

## 2.2.6 Speech Processing

J-DSP also provides blocks that would facilitate the implementation of simple vocoders. Blocks such as the LPC analysis, pitch and voicing detection, residual and vector excitation, as well as LPC parameter transformation and quantization have been developed. These blocks facilitate the simulation of various LPC vocoders. Students can experiment with LPC parameter transformations, i.e., transformations involving direct form, reflection parameters, and line spectral frequencies. Dynamic parameter quantization is also available enabling the students to learn the quantization properties of the different types of LPC parameters. Using J-DSP one can simulate: ADPCM, Residual Excited Linear Prediction (RELP), two-state excitation LP, and Code Excited LP (CELP). Figure 5 shows a LPC vocoder as simulated in J-DSP.

## 2.2.7 Viewing results

Viewing results is made possible through blocks that provide for x-y plots, polar plots, frequency responses or just displaying signal values or statistics in text form. The option to listen to processed speech is also available. Processed speech is played as streaming data through the appropriate Java classes.

## 2.3 J-DSP+

A new version of J-DSP is currently under development, i.e., J-DSP+. This version has all the dynamic features of the current version (J-DSP) and it supports feedback. Adaptive LMS and

BLMS filter blocks were added and simple noise cancellation experiments are now possible with noisy speech signals. Another important addition is the ability to save and load systems giving more flexibility to the user.



**Figure 5.** LPC Parameter Transformations using J-DSP

## 3. ELECTRONIC REPORT SUBMISSION

### 3.1 Overview

In the senior level DSP course offered at Arizona State University, students perform laboratory exercises using the J-DSP editor. After completion of each exercise, they complete an on-line, media-rich report that includes a quiz over the laboratory material. This report is accessed by the students through an Internet browser and consists of a dialog box for entering answers to questions, observations and comments regarding the concepts learned in the lab exercises.



**Figure 6.** Sample electronic report and quiz.

**Figure 7.** Sample grade page.

In addition, the reports have dialog boxes where students can enter the filenames of graphs they generate with J-DSP during the lab exercises. Upon submitting the report, all of the student's answers, comments and graphs are placed together in a static HTML file to be graded by the instructor. Figure 6 shows a part of a sample lab report. Part of the grading process is done automatically while part of it requires instructor intervention. The automatic part is processed by a UNIX shell script on the server computer that grades the answers of the multiple choice and true/false questions. The instructor or teaching assistant completes the grading process by opening the grade page with an Internet browser. This grade page is illustrated in Figure 7. The instructor first selects a student whose work is to be graded. This student's work appears in the instructor's browser along with a box where comments can be entered regarding the student's report. After examining the student's graphs and observations, the instructor assigns a grade and types in a critique of the report, which is then saved on the server. The students can then view their grade along with the instructor's feedback online.

## 4. J-DSP LABORATORY EXERCISES

The J-DSP system has been used as part of a software lab for a senior level digital signal processing course at Arizona State University since the Fall semester of 1997. The following topics are covered in the laboratory exercises.

- **Z-Transform- Digital Filtering - Pole-Zero Plots.** J-DSP is used to examine the various properties of the z transform. Also, the lab exposes students to processing of signals with digital filters including special forms (all-pass, comb filters, digital oscillators). Students can view the pole-zero plot and the frequency response simultaneously. They are also exposed to filter design by pole-zero placement.
- **Linear Phase and FIR Filter Design.** Students perform exercises to understand the property of linear phase and view different types and properties of linear phase filters. They also design FIR filters using parametric and non-parametric windows. The Kaiser design method is examined in detail.

- **IIR Filter Design.** Students perform on-line IIR filter design by analog filter approximation. Butterworth, Chebychev, and Elliptic designs are performed and compared. The properties of the bilinear transformation are examined.
- **FFT and Spectral Analysis.** J-DSP is used to take the FFT of deterministic signals and examine spectral leakage issues using windows. In addition, fast convolutions and circular effects of FFTs are examined.
- **Multirate DSP.** Students perform exercises and implement upsampling and downsampling of signals. They are exposed to aliasing issues. They also design and implement a QMF bank which they use to implement a simple sub-band coder.
- **Statistical Signal Processing:** Students use J-DSP to understand how second-order statistics propagate through digital filters. Exposition to PSD estimation using the correlogram and periodogram.
- **Speech Processing.** Students are introduced to linear prediction. They simulate simple ADPCM and LPC analysis-synthesis systems and evaluate subjectively coded signals using the sound playing capabilities of J-DSP.
- **Adaptive Filtering and Noise Cancellation** Students are introduced to LMS adaptive filtering. System identification and simple noise cancellation experiments are performed.
- **2-D Filtering and Image Processing:** Students design and implement a 2-D filter and visualize its effect on images.

## 5. STUDENT REACTIONS AND CONCLUDING REMARKS

Students who have used J-DSP filled out detailed internet evaluation forms asking them basically to share their experiences and their opinion regarding the use J-DSP for a software lab. The overall responses have been positive and very encouraging. Learning to use the program took most students less than half an hour. The majority of students thought that the program helped them to get a better understanding of signal processing algorithms and they were able to examine applications not covered in text books. J-DSP is also promising for industry applications. In fact, some DSP chip vendors expressed interest in updates of J-DSP that will interface with DSP chips for real time signal processing.

## REFERENCES

[1] Cheneval, Y., Balmelli, L., Pradoni, P., Kovacevic, J., and Vetterli, M., "Interactive DSP education using Java," ICASSP-98, vol. 3, pp. 1905-8, Seattle 1998.

[2] Saad Lamouri, Yusuf Ozturk, Huseyin Abut, "A new collaborative active learning tool for signal processing education," ICASSP-99, vol. 4, pp. 2029-2032, Phoenix 1999

[3] Virginia Stonick, Wojtek Kolodziej, Otto Gygax, " Design of a Guided-Asynchronous Course in Multi-Media Signal Processing, ICASSP-99, vol. 4, pp. 2033-36, Phoenix 1999

[4] Delores Etter, Geoffrey Orsak, "Reflections on a Distance Education Experiment in DSP," ICASSP-99 Vol. 4, pp. 2017-20, Phoenix 1999

[5] Shaffer, J., Hamaker, J., and Picone, J., "Visualization of signal processing concepts," ICASSP-98, vol. 3, pp. 1853-6, Seattle 1998