IEEE ICASSP-2001, Salt Lake city, Utah, May 2001.

DEVELOPMENT OF NEW FUNCTIONS AND SCRIPTING CAPABILITIES IN JAVA-DSP FOR EASY CREATION AND SEAMLESS INTEGRATION OF ANIMATED DSP SIMULATIONS IN WEB COURSES+

Andreas Spanias and Fikre Bizuneh

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

Abstract

Arizona State University (ASU) has developed an on-line DSP laboratory that is based on an object-oriented Java tool called Java Digital Signal Processing (J-DSP). J-DSP is currently being used in a senior-level DSP course at ASU. 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 speech analysis. In this paper, we present a series of significant functionality extensions of J-DSP enabling on-line laboratories in the systems-related areas of Communications, Image Processing, and Controls. The extensions in communications are presented in some detail. In addition to these important functionality extensions, we present enhancements in the infrastructure of J-DSP thatprovide embedded scripting capabilities. Scripts enable easy creation and seamless integration of interactive animations in DSP web content. The latter is very important for instructors creating their own DSP-related web courses as it provides for user-friendly development of interactive visualization modules through the J-DSP applets.

1. Introduction

Electrical engineering education in linear systems and in the related subjects of digital signal processing (DSP), wireless digital communications, and controls has changed dramatically during the last ten years. While in the early nineteen eighties many universities offered topics associated with these subjects in first year graduate courses, in the nineteen nineties many undergraduate programs introduced DSP, wireless communications, and several aspects of image processing and advanced controls in required or elective undergraduate courses. In fact, in some well known undergraduate programs, faculty have argued that discrete-time signal processing could be de-coupled from analog circuits and used as the "first" [1] introductory course in linear signals and systems. At Arizona State University (ASU) systems area courses are quite popular among undergraduates and in particular DSP, communications, and multimedia-related electives are very well attended by on-campus and distance learning students. As an attempt to further satisfy educational needs of undergraduate distance learners in emerging high-tech areas, ASU has established the Multidisciplinary Initiative on Distance Learning (MIDL). The MIDL has developed and tested successfully an exemplary laboratory tool [2] for use in the undergraduate DSP class. This web-based prototype [2] has capabilities for on-line signal processing simulations, and provides laboratory (lab) experiences to distance learning and on-campus undergraduate students. The tool is based on a collection of novel Java applets that support a user-friendly object oriented environment. This simulation environment enables students to establish and execute educational lab simulations from any computer platform equipped with a browser. The MIDL is currently developing and evaluating significant extensions of this important new prototype in other areas of undergraduate education. Extensions of this project include innovative developments leading to a fully developed undergraduate web-based lab tool for the areas of DSP, Communications, Image Processing, and Controls (see Fig. 1).

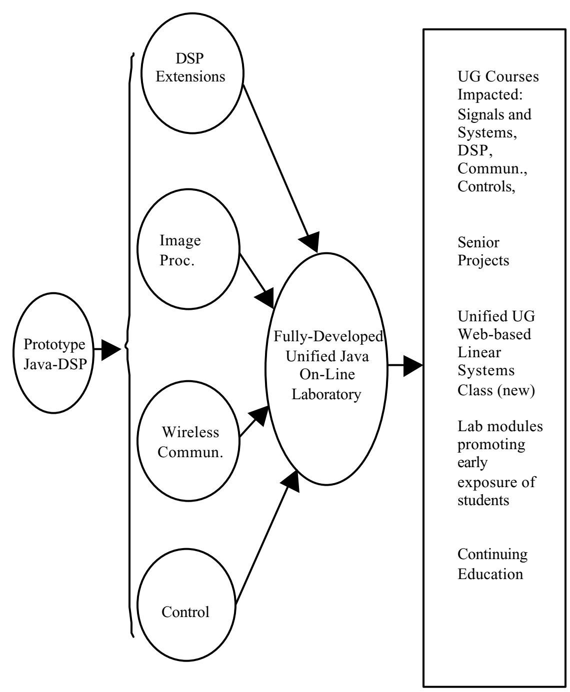

Fig. 1:J-DSP (http://www.eas.asu.edu/~midle/jdsp) extensions to other systems areas

+ *Funded in part by a Grant from NSF CCLI. For more information on J-DSP and its dissemination and contact spanias@asu.edu*

An overview of these extensions is given later in the paper. The communications extensions are described in some detail. In addition, infrastructure extensions are presented with emphasis on the development of new scripting capabilities that enable user friendly creation of visualization modules and seamless integration to web course content material.

2. Description of the Prototype Java-DSP

The need to democratize access to educational tools in DSP through distance learning is as high as ever. At ASU we have been delivering DSP courses for the last ten years through our distance learning TV network. One of the biggest problems in our early distance learning DSP offerings was the weakness in providing lab experiences of distance learners. As a response to this problem, we have developed and successfully tested a novel Java-based signal processing simulation environment for the DSP course called Java-DSP (J-DSP) [2]. This was integrated with web material and electronic lab-report submission software resulting in a comprehensive virtual DSP lab delivered on the web. The lab has been used in our undergraduate DSP class EEE407 attended by more than eighty students every semester. In this paper we describe extensions to the prototype in other systems area courses such as communications, image processing, and controls. Before we describe in detail these extensions, we provide a description of the prototype J-DSP and its capabilities.

The virtual lab environment is tailored to provide hands-on lab experiences to distance learning students over the web. The on-line DSP lab consists of the graphical DSP editor and a software environment for electronic lab report submission and grading. Students use the J-DSP editor to establish and execute lab simulations of DSP algorithms.

The J-DSP editor represents a new paradigm of an object-oriented DSP simulation environment built from the ground up for web lab education. This tool can be accessed using any ordinary browser at **http://www.eas.asu.edu/~midle/JDSP**. J-DSP includes a suite of built-in signal processing functions ranging from simple signal manipulators to complex filter design functions and some preliminary speech, statistical, and adaptive signal processing algorithms. Capabilities for experiments with real-life signals are also provided by using a sound card. Although off-the-shelf object-oriented software for signal processing, controls, and communications have been developed by private software vendors, e.g., Simulink®, SPW®, LabView® etc., much of it is platform-specific and expensive. Limited server-based extensions for some of this software have also appeared, but their integration on the web is by no means seamless. Instead, the integration of the J-DSP editor with web material is virtually seamless. In fact, with the proposed script functionality, one will not only integrate simulations but also have the flexibility to develop scripts, and create and run customized visualization modules. 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 2 shows the J-DSP editor environment. By connecting blocks together, a variety of DSP systems can be simulated.

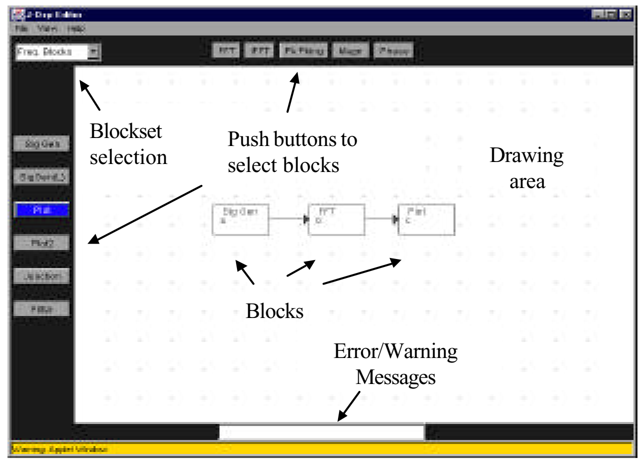

Fig. 2 . J-DSP environment [1]

The fundamental functions embedded in the prototype are: signal synthesis (sinusoids, random signals, periodic or non-periodic pulses), signal operations (addition, multiplication, convolution), filter functions (FIR, IIR, linear phase, impulse and frequency response), pole-zero plots and pole-zero filter synthesis, fast Fourier transform (FFT) and inverse FFT blocks, multi-rate DSP, down-sampling, up-sampling blocks, simple QMF filters, and filter design (FIR, IIR, Kaiser, Butterworth, Chebychev, Elliptic).

A preliminary set of more advanced DSP functions have also been developed and integrated in our prototype. These functions are: random signal processing functions (variance, mean, autocorrelation computations), periodograms, correlograms, parametric spectral estimation, linear prediction and Levinson-Durbin algorithm functions, and Speech processing by linear prediction.

3. Functionality Extensions of J-DSP

Several functions have been developed to support new experiments exposing undergraduates to additional DSP topics such speech analysis and synthesis; adaptive filtering (noise/echo cancellation); time-varying signal analysis. In addition, various functions were developed for communication applications including: digital source generation; amplitude modulation (AM); angle (frequency or phase) modulation; base-band digital modulation (pulse amplitude/position); band-pass digital modulation (amplitude /phase/frequency shift keying); source encoding (Huffman codes); channel encoding (convolutional coding); Viterbi algorithm. Extensions in image processing include: signal generation of 2-D discrete signals; 2-D filtering. Additional functions planned include 2-D filter design; 2-D FFTs; QMF filter banks; blocks supporting and image enhancement. Finally, several functions are being developed for Java controls labs including Bode plots, root-locus, and simple optimization algorithms.

3.1 Communications Functions

The J-DSP's communication functions support simulations of AM, FM, and various digital modulation techniques. Signals can be modulated using the AM and FM blocks. Figure 3 shows an AM modulated signal. The Modulator block takes a binary sequence and provides for digital modulation. Some of the modulation techniques supported by J-DSP are BPSK, QPSK, OQPSK, 16-QAM, MSK, CPFSK, and NFSK. If the input to the modulator is a signal source, a source encoder is used to provide the binary equivalent of the input signal. The output of the modulator can be seen using an in-phase-quadrature signal viewer, called I/Q Plot block. A Demodulator block performs demodulation of the signal received using signal detection methods such as the matched filter implementation. An FFT block can be used to look at the spectral characteristics of the modulator output in both channels. Figure 4 shows a simulation of a QPSK modulated signal.

3.2 Wireless Channel Modeling

Communication channels can also be implemented in J-DSP for simulation purposes. These channels could have additive white Gaussian noise (AWGN) or Rayleigh Fading with filtering characteristics. The channel block simulates the channel impulse response. The channel block dialog allows one to enter coefficients and additive noise parameters (mean and variance or SNR). Figure 5 shows an eye diagram of the output of a channel block for a QPSK modulated signal.

3.3 Channel Coding and Equalization

J-DSP also provides for channel coding and adaptive equalization simulations for communication systems. The Conv_Coder block performs a convolution code on the input sequence. The dialog box of the block allows the user to set the code rate and code word length. Adaptive equalization can also be implemented using the Equalizer block in J-DSP. The equalizer uses the LMS algorithm to estimate the channel filter coefficients and apply them to undo the effects of the channel. Once a signal is equalized and demodulated, a Viterbi block decodes the coded bits. The above blocks provide J-DSP with some capabilities to run digital communication simulations.

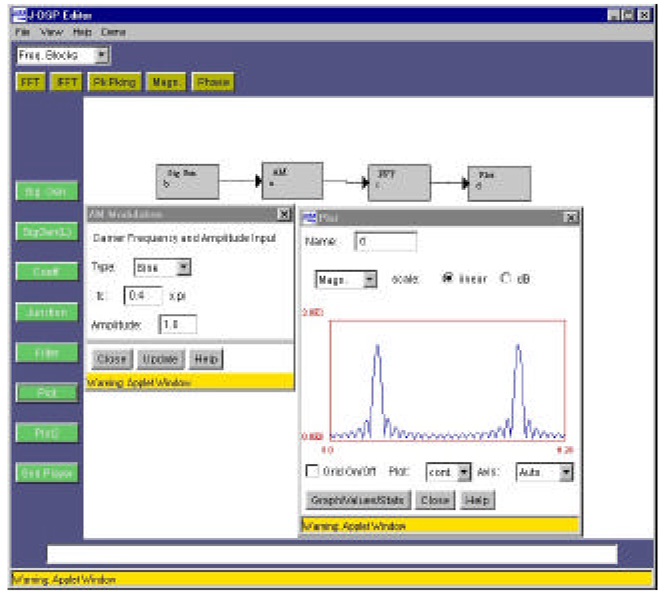

Figure 3: AM Simulation using J-DSP.

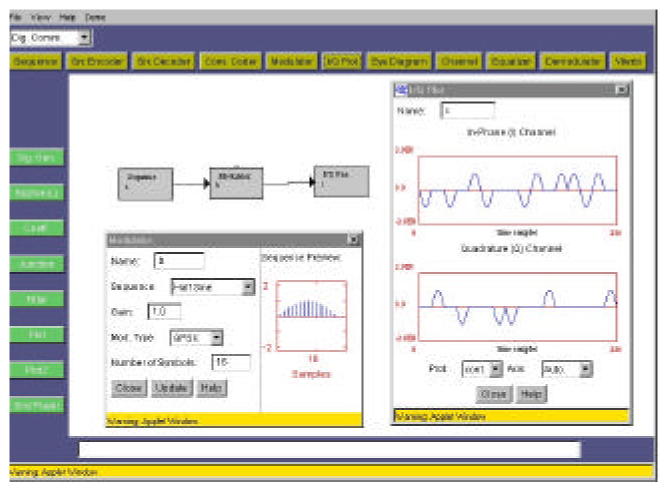

Figure 4: QPSK Modulation simulation

3.4 Communications Labs Based on J-DSP

Communications functionality enables students to analyze the various signal processing stages in a communication system.

- **AM/FM Modulation and Demodulation .** J-DSP is used to simulate AM/FM modulation visualization of the spectrum. Simulation of AM/FM demodulators using filter and envelope detector blocks.
- **Digital Modulation and Demodulation.** Students practice different digital modulation techniques using J-DSP. The effects of pulse shaping on the PSD of the modulated signal are examined in detail. The design and implementation of different receivers can be examined.
- **Channel Characteristics and Coding.** Students simulate different types of channels, such as AWGN and Rayleigh

fading channels. In addition students perform channel coding and convolutional coding exercises.

- **Adaptive Channel Equalizers.** LMS algorithm simulation and its use in channel equalization.
- **Viterbi Decoding.** Students use J-DSP to decode convolution-coded signals. The Viterbi algorithm and hard and soft decision techniques are simulated.
- **Probability of Bit Error.** Simulation of communication systems and bit error vs SNR curves.

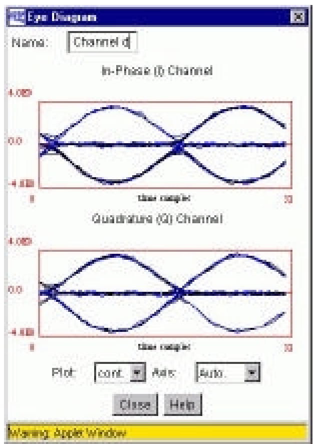

Figure 5: Eye diagram of output of the channel

4. J-DSP Scripting for Easy Creation and Seamless Integration of Simulations in Web Courses

A new feature of J-DSP is provision of HTML embedded scripting capabilities. This feature was developed and integrated into J-DSP to enable users to perform interactive simulations through HTML code. This convenient feature enables instructors to construct and integrate animated visualization modules for DSP and Communications using high-level J-DSP scripts that are embedded in their HTML code. Therefore if animated visualization is desired to explain a DSP concept in a web course, one does not have to develop it using low-level Java software but instead program it through a few high-level J-DSP scripts. This is an important feature that can come very handy in DSP web course design. Sample scripts embedded in HTML code are shown in Fig. 6. The user modifies the <applet> tag and adds the parameters required for running the specific simulation. Each input parameter starts with "param" and provides name and value. The name section is numbered to provide an easy way to remember the number of commands. The first letter in the value section stands for "Block", whereas the number next to it represents the block number. Figure 7 shows the output of the scripts of Figure 6.

5. Concluding Remarks

This paper presented a series of important functionality extensions and infrastructure enhancements on the Java-DSP tool. We emphasize in particular the use of HTML-embedded J-DSP scripts that provide for seamless integration of simulations in web courses. Dissemination and test sites for J-DSP will start in the near future. Future extensions include interface to MATLAB®, provisions to allow development and submission of new functions by other university faculty, provisions that enable multiple users to do new algorithm development and simulation from distributed sites, and interface to DSP chips.

```
<html>
<head>

<title>Document Title</title>
</head>
<body>
<p align="center">
<applet code="JDsp.class" width=400 height=250>

<param name=numCommand value=9>
<param name=0 value=B0-siggen(0,1)>
<param name=1 value=B1-filter(1,1)>
<param name=2 value=B2-plot(2,1)>
<param name=3 value=B3-pzplace(1,2)>
<param name=4 value=B4-frqrsp(1,0)>

<param name=5 value=C-0-4-1-0>
<param name=6 value=C-1-4-2-0>
<param name=7 value=C-3-3-1-2>
<param name=8 value=C-1-3-4-2>

</applet>
</p>
</body>
</html>
```

Figure 6: Sample Script HTML

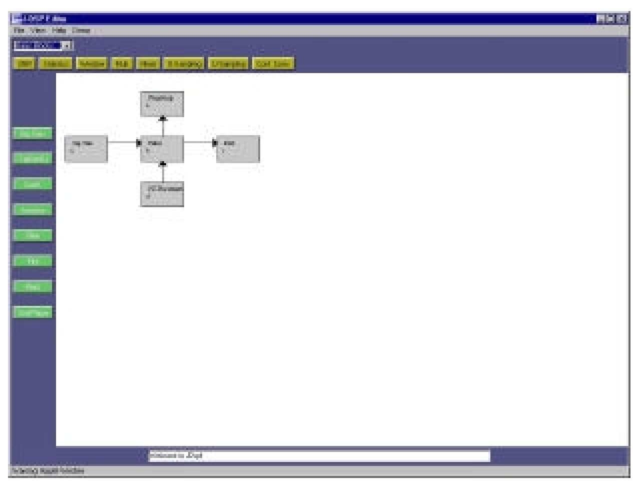

Figure 7: Result of the sample Script HTML of Figure 6

References

[1] McClellan, Schafer, and Yoder, DSP First, Prentice Hall 1998.

[2] A. Spanias et al, "Development and Evaluation of a Web-Based Signal and Speech Processing Laboratory for Distance Learning," Proc. IEEE ICASSP-2000, Istanbul, June 2000