

AN INTEGRATED GRAPHICAL ENVIRONMENT FOR WEB-BASED LEARNING

Shalin Mehta, Andreas Spanias, Jayaraman J. Thiagarajan,
Mahesh K. Banavar, Karthikeyan Natesan Ramamurthy, Robert Santucci,
Constantinos Pattichis² Photini Spanias¹ and Harish Krishnamoorthi

SenSIP Center, School of ECEE, MLFTC¹
Arizona State University

²Department of Computer Science
University of Cyprus

Abstract

In this paper, we present an integrated learning framework that includes course delivery, interactive online laboratories, video streaming, Java visualization modules, testing, and assessment. The proposed framework uses HTML and Java based technologies to augment the existing J-DSP environment, in order to improve the overall learning experience of students in both undergraduate and graduate level signal processing courses. An interactive DSP quiz forms the central part of the proposed interface, through which the different learning components can be accessed. In addition, functionality that enables students to execute custom Java code in a J-DSP block diagram has also been developed. The initial assessment results are encouraging and indicate that students as well as instructors find the proposed infrastructure helpful.

Introduction

The use of web-based technologies in the modern teaching process is motivated by the need to stimulate student interest and to facilitate personalized learning among students. These technologies typically incorporate several components such as interactivity, dynamic displays, online computations, audio, and video. Including innovative tools such as interactive demos and animations to complement the traditional teaching material will be helpful in engaging students, including distance learners. The crucial role of interactivity in knowledge acquisition and the development of cognitive skills is discussed in detail in [1]. Furthermore, in [2], it is emphasized that in comparison to

other web applications, education tools typically require a higher degree of interactivity in order to enhance learning potential. The current advancements in streaming media applications also present a huge potential for web-based education. In addition to being universally accessible, web-based software is the natural choice for distance learning. Efforts to employ web-based technologies in engineering education have been reported in [3-7].

In the last decade, web-based learning has had an increasing impact in Digital Signal Processing (DSP) education and course delivery. Several software applications that incorporate visual components in learning to simplify the understanding of complex theoretical concepts have been developed. These applications are often characterized by rich user interaction and ease of accessibility. This motivated the design of web-based, platform-independent applications. Some of the earliest efforts include the Infinity Project [8] that introduced a signal processing based engineering curriculum to high school students, and Mallard [9] that enabled students to learn DSP concepts through an interactive quiz environment. Several Java-based modules to demonstrate essential DSP concepts were developed as part of the *WebEng* project [10]. In [11], the authors developed a DSP teaching tool using Shockwave and Macromedia Director and employed the Technology Acceptance Model (TAM) [12] to understand its strengths and weaknesses. In order to enable students to perform DSP labs over the Internet, the authors in [13] developed J-DSP, a visual programming environment. J-DSP was designed as a zero footprint, standalone Java applet that can run

directly on a browser [14]. Several interactive laboratories have been developed and assessed in undergraduate courses.

One limitation of these tools is the lack of a centralized interface, similar to that shown in Figure 1, to access all the required components such as lecture slides, course notes, audio/video content, links to websites, animations and software modules. In this paper, we address this problem by developing a web-based interface that augments the J-DSP simulation environment, to integrate the discrete components and to enable students to organize their learning. Figure 2 illustrates the proposed interface for integrated web-based learning. As it can be observed, a quiz interface forms the centerpiece of the proposed framework. In addition to testing the student understanding of DSP concepts, the quiz interface contains links to access the discrete learning components provided by the instructor. Since the proposed interface resides in the J-DSP environment, students can create J-DSP block diagrams while accessing the quiz interface. Furthermore, the proposed framework provides a functionality that allows users to include custom Java functions into an existing block diagram and dynamically execute them. In addition to expanding J-DSP beyond the existing palette of functions, this feature allows students to customize simulations corresponding to a quiz question.

The proposed interface has features that appeal to both students and instructors. In cases where a student makes a mistake in the quiz, links to the different modules are activated. The student can then review the materials and attempt a different question, drawn from the same concept, posed by the quiz engine. This feedback process guides the students in clarifying the concepts that were not clear to them the first time. Instructors are provided with a user-friendly interface to edit the quiz questions and specify answers for the quiz engine to evaluate the student responses. Furthermore, they can use the J-DSP scripts to

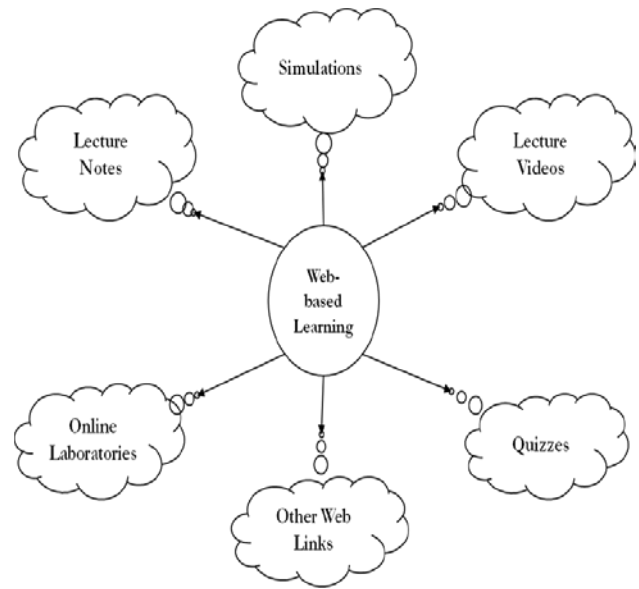


Figure 1: The standard distributed infrastructure that presents the need to integrate several learning modules in DSP education.

provide students with pre-developed animations that demonstrate DSP concepts. Instructors are also provided with a utility to perform detailed assessments of student performance.

In order to determine the impact of the integrated learning framework in student understanding of the course material, we administered assessments to students from an undergraduate DSP course at Arizona State University. The evaluation was carried out to understand if the proposed interface enhanced and reinforced student learning in DSP topics. The results of the assessments are discussed in this paper. Addressing the issues identified based on the student evaluation will enable us to provide prescriptive recommendations concerning strengths, replication, and sustainability of the learning framework.

The rest of this paper is organized as follows. In the next section we briefly describe the existing J-DSP simulation environment and its functionality. The architecture of the proposed learning framework and descriptions of its several features are presented in the following section. We then describe the functionality to include custom Java code in J-DSP simulations.

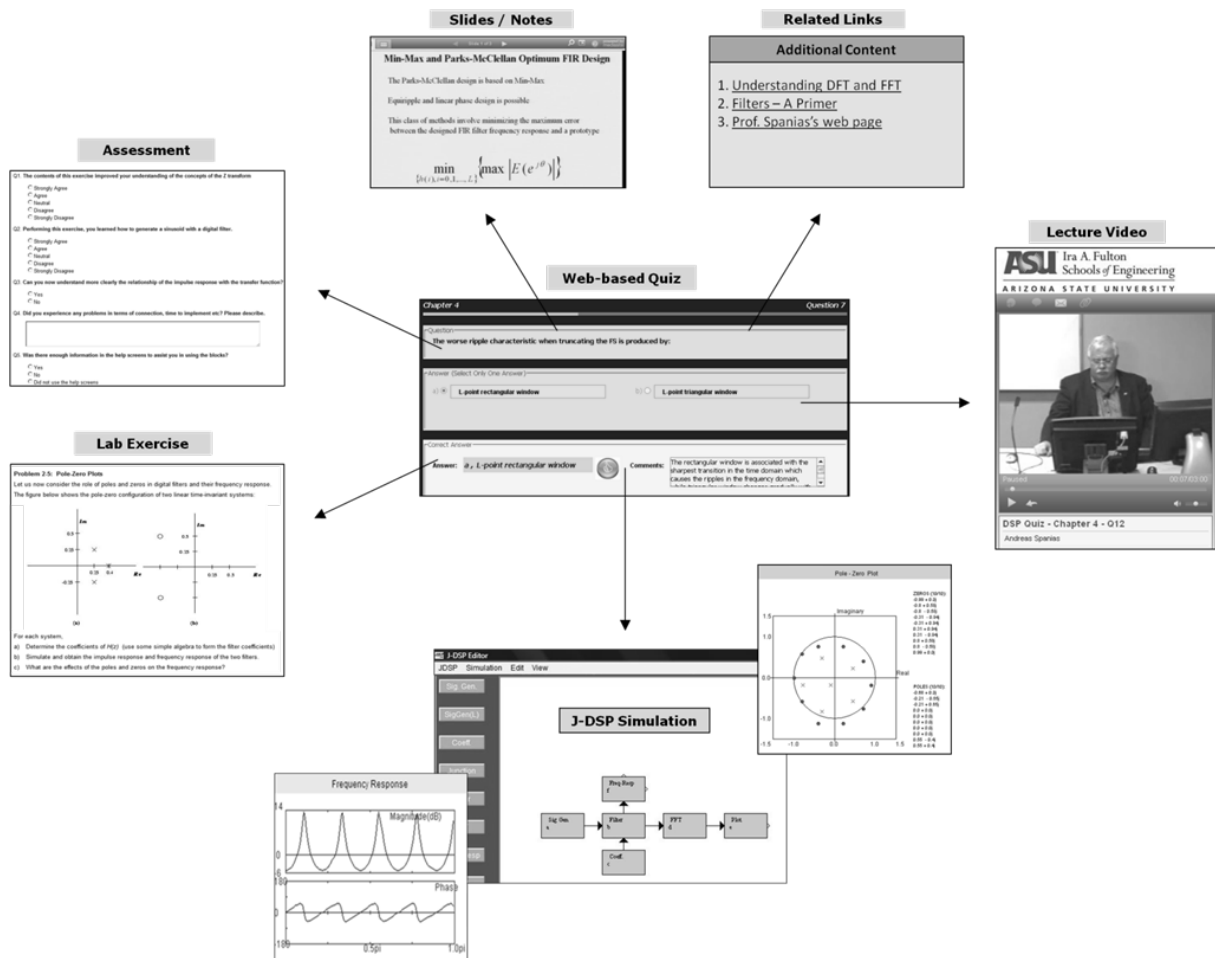


Figure 2: Proposed interface for integrated web-based learning using J-DSP. The different components/modules are accessible from the centralized quiz interface. Using this integrated environment, students can view video lectures, run simulations, take interactive quizzes, file lab reports and participate in assessments.

The assessment results of the proposed framework are then discussed. Finally, the concluding remarks are presented.

J-DSP Simulation Environment

The primary objective of the J-DSP software was to provide undergraduate students and distance learners hands on experience with DSP concepts and to simplify the understanding of complex mathematical procedures. All signal manipulation functions appear in J-DSP as graphical blocks that are brought into the simulation environment by a drag-n-drop process. Signal and data flow is established by simply linking the blocks. Each block contains a dialog window with which the parameters of the

function can be updated dynamically. The students can plot outputs at various stages of a simulation and thereby understand the signal flow in the implementation. Since J-DSP is developed in Java, it has the advantage of inherent platform independence. Users can perform simulations on any operating system of their choice. The system requirements to run the software is minimal and it includes a stable internet connection and a Java run time enabled web browser. Since J-DSP is a zero footprint software, it does not require any installation. The software provides free and universal accessibility over the World Wide Web (WWW). Moreover, the inherent Java backbone enables J-DSP to be interfaced with streaming media. Internet lectures, and visual demonstrations. Furthermore, 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. The interfaces to MATLAB [15] and LabVIEW [16] allow synergies with other third party environments.

The basic J-DSP functions include sampling, convolution, Fast Fourier Transform (FFT), digital filter design and arithmetic operations. Advanced functions include statistical DSP algorithms, speech processing functions, multirate signal processing and spectral analysis functions. In addition, it is supported by toolboxes for image processing, control systems, time-frequency analysis, and analog/digital communications. Furthermore, hardware interfaces to TI DSK [17] and Crossbow sensor nodes [18] have also been developed. Additional J-DSP software functions and blocks associated with speech and audio processing algorithms developed at ASU have been formed using the theory and methods described in [26-30]. Furthermore, recent efforts to extend the J-DSP framework to multidisciplinary applications have resulted in independent versions for genomic signal processing [19], earth systems signal processing [20] and RF amplifier linearization [21].

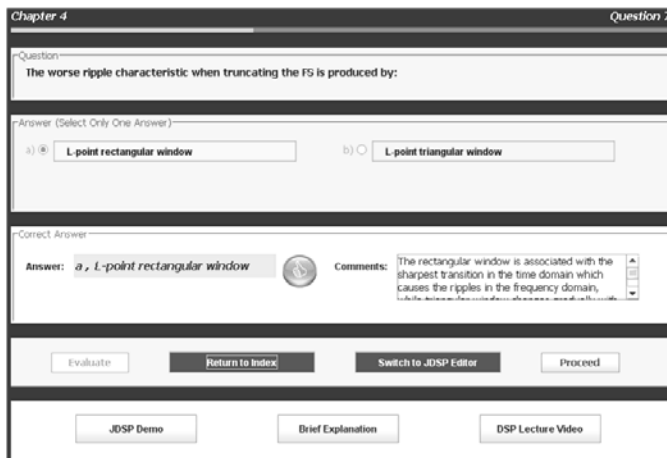
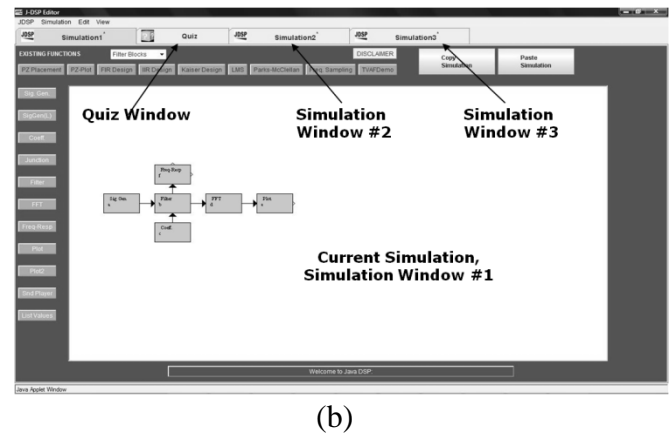
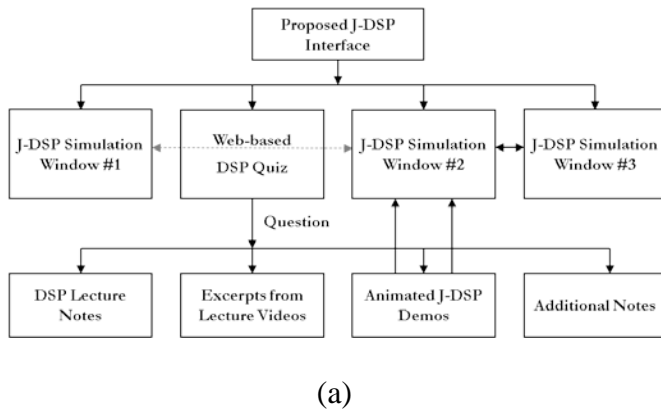
Proposed Interactive Learning Framework

As described earlier, the interface proposed in this paper is built on the idea of integrating a web-based DSP quiz and the existing J-DSP simulation environment. Figure 3(a) illustrates the architectural overview of the learning framework. As it can be observed, the interactive quiz is the central piece that connects to the discrete learning components. After responding to a question in the quiz, students are provided with access to the different modules in order to obtain additional details on a particular concept. In addition, the framework contains multiple simulation windows for the students to create J-DSP block diagrams and to view pre-developed J-DSP animations created by the instructor.

The existing J-DSP infrastructure requires multiple instances of the J-DSP editor to execute multiple simulations in parallel. However, using a new instance for each simulation consumes a lot of memory, and the communication between the simulations is not supported as they run in different browser instances. To address this issue, the number of simulation windows has been increased to three in the proposed framework. This functionality also helps in creating very large simulations that cannot be accommodated in a single J-DSP window and communication can be established between two different simulation windows. Figure 3(b) shows a snapshot of the proposed J-DSP interface. All simulation windows and the quiz are part of a single browser instance and reside in individual tabs.

Web-based Quiz

The proposed interface ensures that the students can easily access the relevant learning resources through the centralized quiz questions. Figure 3(c) shows the quiz window along with its components. The links to all external components reside in the quiz window and these links are initially disabled when the students are presented with the question. After answering a question, students can review the lecture slides or watch relevant excerpts from the lecture video. They can also read through the additional notes embedded in the quiz window itself. Furthermore, instructors can post J-DSP animations to demonstrate concepts related to the quiz question. The instructors can include questions which require: (a) responses that may contain text and numbers, (b) responses that may contain special symbols, and (c) selecting one or more of the pre-specified options. Furthermore, the queries and the pre-specified options can include images and symbols. To ensure that the use of symbols is consistent, users are provided with the list of symbols which are to be used in their responses. Figure 3(d) illustrates the dialog window with the set of valid symbols supported by the quiz engine. To evaluate textual responses, the quiz engine ignores the sentence case, parentheses and



d	:	delta (δ)
W	:	Digital Frequency (angular)(Ω)
w	:	Analog Frequency (angular)(ω)
f	:	frequency
fs	:	Sampling Frequency
f0, f1	:	Input Signal Frequency
pi	:	pi (π)
exp()	:	exponential
sin	:	Sine
cos	:	Cosine
*	:	Multiplication
conv()	:	Convolution
A	:	Power
B	:	Beta
Tau	:	Tau
!=	:	Not equal to
>=	:	Greater than or equal to
conj()	:	Conjugate
mu	:	mean
var	:	variance

Please write the equations with the delay in increasing order
Example: $h(n) = d(n) + 2d(n-1) + d(n-2) + 5$

(d)

Figure 3: Proposed interactive learning interface - (a) Architectural overview, (b) New J-DSP interface with multiple simulation windows, (c) GUI of the DSP quiz, and (d) Notation dialog with the allowed set of symbols for the quiz engine.

empty spaces. However, evaluating questions which require textual or symbolic responses can be very difficult in some cases, due to the possibility of multiple solutions. For example, if the correct response for a particular question is $50(x+y)$, entering $50x+50y$ is also a valid solution. Hence, the instructor is provided with the option of adding more than one correct response in the database.

In order to provide an improved self-learning experience, students are allowed to have multiple attempts at understanding a concept. In cases where the students fail to answer a question, the activated links to the learning resources will enable them to understand the

concept addressed in the question. The quiz engine then presents the student with another question drawn from the same concept to test their understanding. In addition to providing students with an opportunity to correct their mistakes, it enables the instructors to identify sections of the course material that will need to be emphasized. Furthermore, instructors have the option of allowing multiple attempts to questions that require students to respond with symbolic or numeric expressions. The links to the components will be activated after the first attempt and the students can make another attempt at answering the question for partial credit.

Diverse Learning Components

In this section, we briefly discuss the various learning components included in the proposed framework. It is important to note that this list of resources is not exhaustive and the instructor can enhance the interface with other innovative modules.

Animated J-DSP Demos The inherent scripting functionality in J-DSP has been extended to create animated demos of simulations for demonstration purposes. As soon as the *J-DSP Demo* button is pressed, the relevant animation starts in the J-DSP simulation window. Options are provided to change the speed of the animation or to skip the entire animation. Note that this animation not only allows users to view the simulation and the plots, it also constructs the actual block diagram in the simulation area that can be edited by the users.

Lecture Notes When users click the *Brief Explanation* button, a new tab is opened in a browser, where the relevant slides of the lectures are displayed in PDF format. This is a very important component as students typically refer to the lecture notes more often than any other material and these notes present the theory comprehensively.

Video Excerpts The interface provides an option to access relevant parts of the lecture video, when available. This feature is very helpful as using video lectures with simple examples pertinent to the theory is useful in reinforcing student learning. This also allows students to refresh their concepts by listening to the class lectures again.

Additional Web Links In addition to these components instructors can add additional information, in the *Comments* section, that may not have been covered by the other components. For example, a brief explanation about the animated demo can be provided or external web links can be added.

Utilities for the Course Instructor

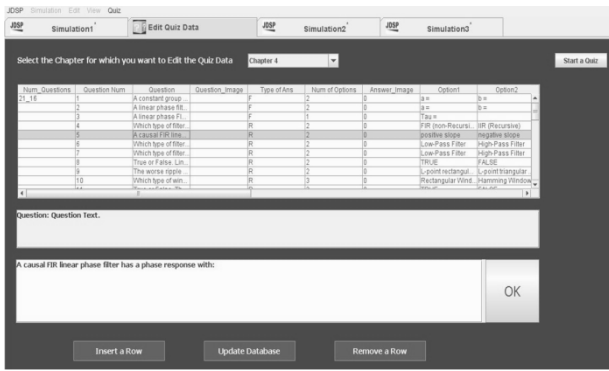
Administrative features have been included in the framework to facilitate instructors to maintain the interface and carry out detailed assessments of the student performance.

Editing the Quiz The interface contains a simple utility that enables the instructors to update the database of the web-based quiz. Since the database can only be viewed or updated by the instructor, administrative privileges are required to perform this task. The proposed framework requires the instructor to log into the J-DSP environment and access a spreadsheet to view and modify the quiz database. This simple and intuitive interface is illustrated in Figure 4(a). The spreadsheet resides as a tabbed window in the proposed interface.

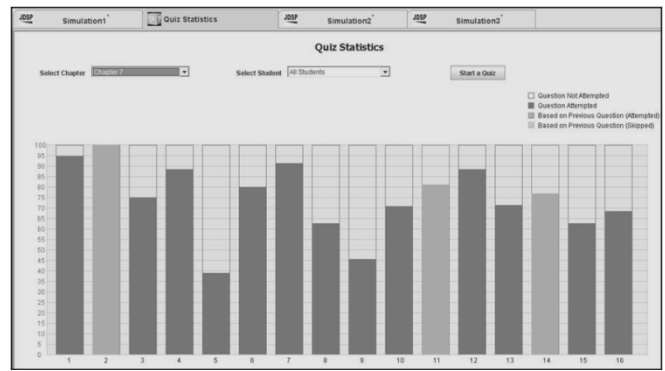
Analyzing the Statistics This feature allows instructors to view the distribution of student performance on the quiz questions. The analysis in a chapter can be viewed specific to a student or the overall performance of the class. This enables instructors to perform the cumbersome task of identifying specific areas of the subject that need more emphasis, with considerable ease. Figure 4(b) illustrates an example statistics of the student performance in a particular chapter.

Developing Customized DSP Functions

In practice, complex signal processing algorithms are typically tested and analyzed using simulation software prior to implementing them in hardware or low-level software. Hence, it is important to allow students to build and simulate DSP algorithms as part of the quiz, in addition to answering theoretical questions. The rich suite of functions in J-DSP allows the users to simulate several basic and advanced algorithms. However, course instructors might be interested in engaging students in implementing a few signal processing functions on their own, in addition to constructing block



(a)



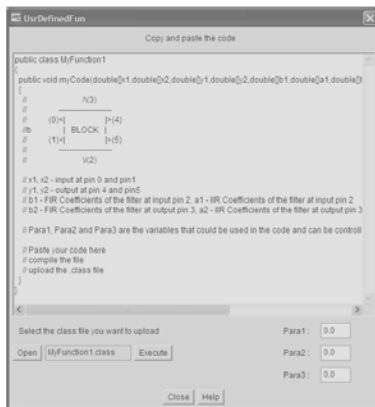
(b)

Figure 4: Utilities for the instructor: (a) Interface for editing the quiz questions and the options for the different learning components, (b) Interface for analyzing the chapter-wise statistics of the student performance.

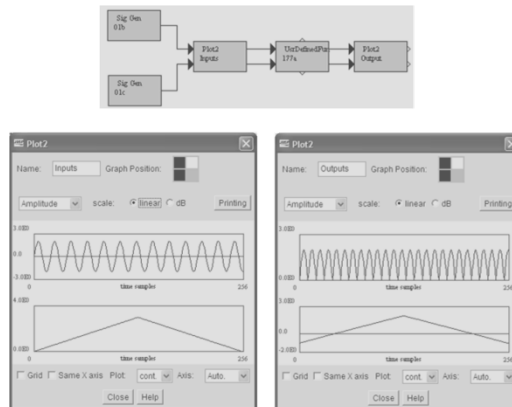
diagrams. In other words, there is a need for students to add new custom functions in J-DSP and interface the new function with the existing blocks for analysis.

As part of the proposed learning framework, we have developed a functionality to allow users to add their own customized functionality as a J-DSP block (*User-defined block*). This function is similar to other J-DSP blocks, except that it allows users to load their Java class files during runtime. Figure 5(a) illustrates the dialog window corresponding to the *User-defined block*. As it can be observed, this block specifies the users with the input and output pins to which the input and output signal arrays can be associated. The text area in the dialog window

provides the prototype of the Java class (*MyFunction1*), with which users can develop their code. The block accepts two signal arrays as inputs, implements the user-defined function, and generates the corresponding outputs. There is also an option to provide filter coefficients as inputs. Furthermore, users can also include a maximum of three additional parameter variables namely, *Para1*, *Para2* and *Para3* in their code and the values of these parameters could be controlled from the dialog window dynamically. The code for the custom function needs to be written in Java using the template provided (*MyFunction1.java*) and compiled locally to generate the *.class* file (Figure 5(b)). Since J-DSP software does not contain an built-in Java compiler, it can accept only



(a)



(b)

```

MyFunction1.java
/*
      ^ (3)
      (0) < BLOCK > (4)
      (1) < > (5) // BASIC BLOCK MODEL
      v (2)
*/
// x1, x2 - input at pin 0 and pin1
// y1, y2 - output at pin 4 and pin5
{
  for(int i=0; i<256; i++)
  {
    y1[i] = Math.abs(x1[i]);
  }
  for(int i=0; i<256; i++)
  {
    y2[i] = x2[i] - para1;
  }
}

```

Figure 5: Proposed functionality to interface customized Java code to the J-DSP simulation environment: (a) Dialog window corresponding to the User-defined block, (b) An example demonstration and the sample Java code developed using the template provided in J-DSP.

compiled *.class* files. The *Open* button in the dialog window allows users to locate the compiled Java (*.class*) file and pressing the *Execute* button loads the file into the J-DSP simulation. As soon as the file is loaded, it is executed dynamically, and the signals at the output pins change suitably.

Assessments

In this section, we describe the student assessments of the J-DSP software, and the proposed learning framework obtained from an undergraduate signal processing class. This course is offered online as well, where the web-based self learning is particularly useful. The goal of this evaluation was to identify the impact of integrating several learning components, in to the existing J-DSP environment, in student understanding. These assessments were used to determine if the proposed framework could enhance student learning, particularly in topics that students find challenging. Furthermore, general assessments about the aesthetics and the usability were used to obtain an overall subjective opinion about the environment. The pedagogy adopted for the use of the J-DSP interface included the following activities: lecture on the pertinent signal processing concepts, an online laboratory using the J-DSP programming environment and a web-based quiz with access to relevant learning resources.

Using J-DSP for Signal Analysis Laboratories

The undergraduate DSP course at Arizona State University covers topics pertinent to LTI system analysis, *z*-Transforms, FFT based spectral analysis, digital filter design, and multirate signal processing. In order to facilitate the understanding of the complex theoretical concepts, a set of J-DSP laboratory exercises has been developed. These hands-on 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 either in class or in textbooks. The computational nature of the J-DSP laboratories and the inclusion of real-life signals make J-DSP particularly useful in providing engineering intuition and valuable hands-on experience to the class participants.

A set of concept-specific and general evaluation forms have been developed to understand the impact of using J-DSP for performing laboratories and understanding DSP fundamentals. The general assessment questionnaire enables the assessment of the ergonomics, and the usefulness of J-DSP software. Some of the aspects evaluated include the software capabilities, academic standing, browser compatibilities, expediency of the user manual etc. Student responses revealed that the user interface and the easy access to J-DSP were received positively. Most students found building J-DSP simulations highly intuitive and the comprehensive function set very useful. 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. Some of the statistics based on the general and concept-specific assessments can be found in Table 1. Furthermore, students found adapting to the J-DSP environment quite easy, and as illustrated by Figure 6 about 70% of the students took less than 30 minutes to learn the environment.

Evaluation of the Proposed Integrated Learning Framework

The first set of assessment results were generated by analyzing the student response to the quiz questions on digital filter design, that allowed multiple attempts, and to questions drawn from the same concept which the student failed to answer during the first attempt. Figure 7(a) illustrates the student performance on questions that allowed multiple attempts. For each question, a maximum of three attempts were allowed and it can be easily observed that after accessing the relevant resources, students

Table 1: Statistics based on general and concept-specific assessments on using J-DSP for performing undergraduate signal analysis laboratories.

Evaluation Question	Strongly Agree (%)	Agree (%)	Neutral (%)	Disagree (%)	Strongly Disagree (%)
Establishing and connecting blocks in J-DSP is easy	53	39	7	1	0
User interface of J-DSP is intuitive and user-friendly	31	63	5	1	0
With the help of a simple manual, it is easy to learn the basics of DSP using J-DSP	86	0	0	14	0
The online lab concept is beneficial for distance learning	97	0	0	3	0
Enough information is available on the Help screens and dialog windows	49	33.5	0	13.5	0
J-DSP online labs helped me understand the concepts of z-Transform	50	46	3	1	0
Understanding of FIR and IIR filter design is enhanced by the J-DSP exercises	42	47	8	3	0
Using FFT in signal analysis is clear by performing a J-DSP simulation	24	61	13	2	0

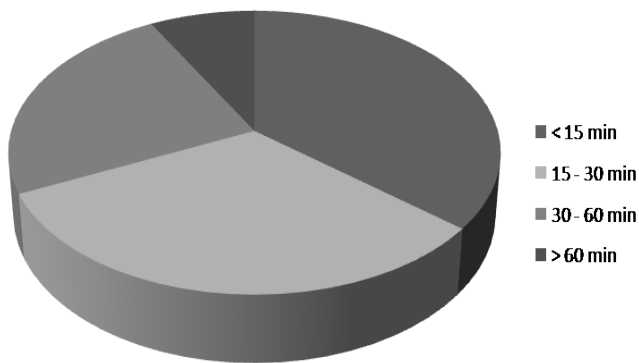
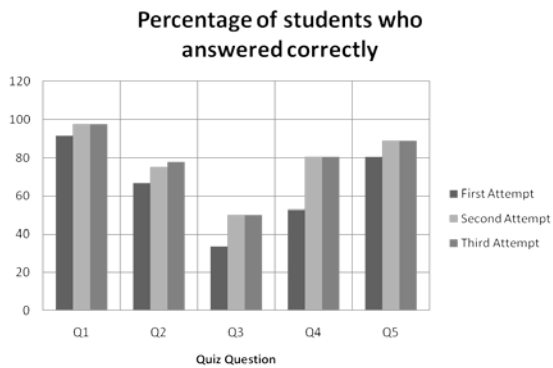


Figure 6: Time taken by the students to learn the simulation environment of J-DSP.

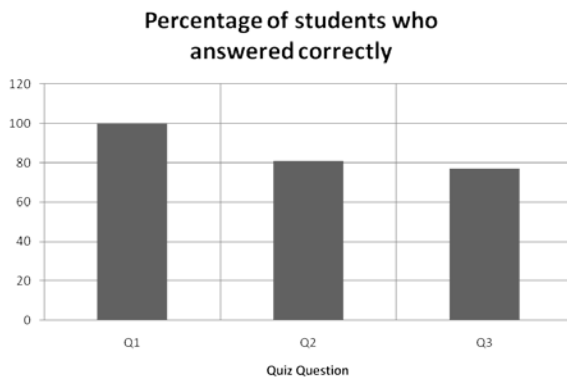
demonstrated improvements in their understanding. For example, only 52% of the students correctly answered question number 4 correctly in their first attempt. However around 80% answered it correctly in the subsequent attempt. Figure 7(b) shows the percentage of students that answered different questions drawn from the same concept, after they failed

to answer the previous question. It is important to note that these questions allow only one attempt. The results indicate that all students performed quite well on these questions, which reflects the value of the additional resources in understanding the concepts.

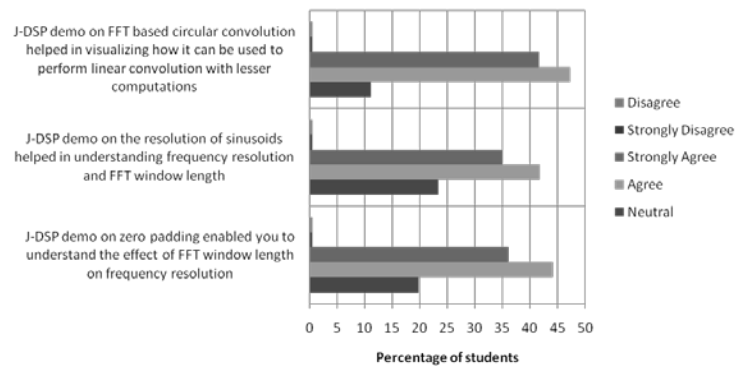
The next set of assessment results were generated by asking the students specific questions about their experience with the new interface. In particular, we evaluated the usefulness of the links to the other learning components such as the animated demos, lecture videos and lecture notes. Figure 7(c) shows the statistics obtained based on the responses of the students about the usefulness of the J-DSP animations, while Figure 7(d) evaluates the use of lecture videos and relevant notes. More than 80% of the students agreed that the J-DSP demos provided a clear visualization of the concept presented in the question. Furthermore, 85% of the students felt that it was very



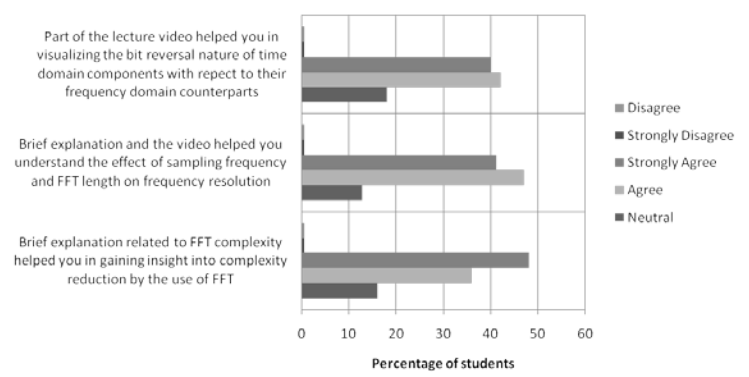
(a)



(b)



(c)



(d)

Figure 7: Assessment results - (a) Student performance on questions that allowed multiple attempts, (b) Performance on questions drawn from the same concept which the students failed to answer previously, (c) Evaluation of the impact of J-DSP animations, (d) Assessment of the use of links to lecture video and relevant notes.

convenient to have the links to the lecture slides and video excerpts in order to review the fundamentals. The final set of assessment questions was presented to identify student opinion about the organization and aesthetics of the proposed interface. More than 90% of the students reacted positively towards the integration of web-based quiz and simulation capabilities under a single framework. Furthermore, we have incorporated some of the changes that the students recommended regarding the appearance of the quiz interface.

Conclusions

In this paper, we described a framework for integrated learning using the J-DSP simulation environment. The main advantages of this interface are its interactivity and integration with several learning components and online

simulations, which perfectly aligns with our goal of enhancing the self learning experience. The web-based quiz has a number of attractive features with the main focus on the improvement of student learning rather than just testing their understanding of DSP concepts. This enables students to organize their learning and to access related course material easily. The proposed framework emphasizes the need for simulating complex DSP algorithms by allowing students to build and execute custom Java functionality in a J-DSP block diagram. The proposed interface was assessed by the students of an undergraduate DSP course at Arizona State University. The assessment results are very encouraging, and show that students find the infrastructure very helpful and easy to use. Their feedback indicates that the interface facilitated the understanding of several DSP concepts, which they found challenging.

Though the focus of the paper was on building a framework for DSP education, this interface can be adapted to other disciplines as well. Finally we note that work in J-DSP continues with extensions of the software to smart phones and tablets [22,23], computer exercises [24] and workshops to assess them [25].

References

1. R. Sims, "Interactivity: A forgotten art?" *Computers in Human Behavior*, vol. 13, no. 2, pp. 157–180, 1997.
2. C. Chou, "Interactivity and interactive functions in web-based learning systems: A technical framework for designers." *British Journal of Educational Technology*, vol. 34, no. 3, pp. 265–279, 2003.
3. I. Humar, A. R. Sinigoj, J. Bester, and M. O. Hagler, "Integrated component web-based interactive learning systems for engineering," *IEEE Transactions on Education*, vol. 48, no. 4, pp. 664–675, nov 2005.
4. U. Drogenik and J. W. Kolar, "Interactive Power Electronics Seminar (iPES)-a web-based introductory power electronics course employing Java-applets," in *IEEE Power Electronics Specialists Conference*, vol. 2, 2002, pp. 443–448.
5. M. F. Iskander, "Technology-based electromagnetic education," *IEEE Transactions on Microwave Theory and Techniques*, vol. 50, no. 3, pp. 1015–1020, mar 2002.
6. I. Humar, E. Bulic, and A. R. Sinigoj, "Solved problems of fundamentals of electromagnetics on the web: System usage evaluation," *Electrotech. Rev.*, vol. 71, no. 1-2, pp. 40–44, 2004.
7. D. Millard and G. Burnham, "Innovative interactive media for electrical engineering education," in *Proceedings of IEEE Frontiers in Education*, vol. 3, 2001, pp. S3C–17–22.
8. Anon., "The Infinity Project: Engineering education for today's classroom," <http://www.infinity-project.org/infinity/> (accessed August 2012).
9. D. J. Brown, M. Covington, and M. L. Swafford, "Mallard: An educational tool for digital signal processing," in *Proceedings of Asilomar Conference on Signals, Systems and Computers*, vol. 1, 1996, pp. 231–235.
10. M. J. Jackson, D. I. Laurenson, and B. Mulgrew, "Developing and evaluating Java-based educational tools," in *IEEE International Symposium on Engineering Education: Innovations in Teaching, Learning and Assessment*, vol. 2, 2001, pp. 26/1–26/6.
11. S. L. T. Marin et.al., "Implementation of a web-based educational tool for digital signal processing teaching using the technological acceptance model," *IEEE Transactions on Education*, vol. 48, no. 4, pp. 632–641, nov 2005.
12. J. M. Wilson and W. C. Jennings, "Studio courses: How information technology is changing the way we teach, on campus and off," *Proceedings of the IEEE*, vol. 88, no. 1, pp. 72–80, jan 2000.
13. A. Spanias and V. Atti, "Interactive online undergraduate laboratories using J-DSP," *IEEE Transactions on Education*, vol. 48, no. 4, pp. 735–749, nov 2005.
14. V. Atti and A. Spanias, "Online simulation modules for teaching speech and audio compression techniques," in

Proceedings of IEEE Frontiers in Education, vol. 1, nov 2003, pp. T4E–17–22.

15. A. Spanias, C. Panayiotou, T. Thrasyvoulou, and V. Atti, “Java-DSP interface with MATLAB and its use in engineering education,” in *Proceedings of ASEE Annual Conference and Exposition*, jun 2004.
16. A. Spanias, K. Ramamurthy, J. J. Thiagarajan, M. Banavar, and C. Huang, “Using JDSP and LabVIEW to perform undergraduate labs,” in *Proceedings of ASEE Annual Conference and Exposition*, jun 2007.
17. A. Spanias, V. Berisha, H. Kwon, C. Huang, A. Natarajan, and R. Ferzli, “Using the Java-DSP real-time hardware interface in undergraduate classes,” in *Proceedings of IEEE Frontiers in Education*, oct 2006, pp. 12–17.
18. H. Kwon, V. Berisha, V. Atti, and A. Spanias, “Experiments with sensor notes and Java-DSP,” *IEEE Transactions on Education*, vol. 52, no. 2, pp. 257–262, may 2009.
19. A. Spanias, N. Chakravarthy, Y. Song, and L. Iasemidis, “Teaching genomics and bioinformatics to undergraduates using J-DSP,” in *Proceedings of ASEE Annual Conference and Exposition*, June 2004.
20. K. Ramamurthy, A. Spanias, L. Hinnov, and J. Ogg, “On the use of J-DSP in earth systems,” in *Proceedings of ASEE Annual Conference and Exposition*, June 2008.
21. R. Santucci, T. Gupta, M. Shah, and A. Spanias, “Advanced functions of Java-DSP for use in electrical and computer engineering courses,” in *Proceedings of ASEE Annual Conference and Exposition*, June 2010.
22. J. Liu, S. Hu, J. J. Thiagarajan, X. Zhang, S. Ranganath, K. N. Ramamurthy, M. Banavar, and A. Spanias, “Interactive DSP laboratories on mobile phones and tablets,” Proc. of IEEE ICASSP 2012, pp. 2761 – 2764, Kyoto, March 2012.
23. A. Spanias, Plenary session given at IEEE FIE 2012 on Premier Award for iJDSP Software development, <http://more.engineering.asu.edu/ijdsp/>, Seattle, Oct. 2012.
24. A. Spanias, DSP – An Interactive Approach, Digital Signal Processing: An Interactive Approach, ISBN: 978-1-4243-2524-5, September 2007
25. iJDSP Assessment Workshops, http://jdsp.engineering.asu.edu/ASU_UCY_Workshop_2013/index.html
26. T. Painter, A.S. Spanias, —Perceptual Coding of Digital Audio, Proceedings of the IEEE, pp. 451-513, Vol. 88, No.4, April 2000.
27. A.S. Spanias, "Speech Coding: A Tutorial Review," Proc. IEEE, Vol. 82, No. 10, pp. 1441-1582, October 1994.
28. K. N. Ramamurthy, A. S. Spanias, MATLAB® Software for the Code Excited Linear Prediction Algorithm: The Federal Standard-1016, Morgan and Claypool Publishers, Vol. 2, No. 1, ISBN 1608453847, Jan 2010.
29. J. J. Thiagarajan, Andreas Spanias, Analysis of the MPEG-1 Layer III (MP3) Algorithm Using MATLAB, Morgan and Claypool Publishers, Vol. 3, No. 3, ISBN-10: 1608458016, ISBN-13: 978-1608458011, November 2011.

30. Andreas Spanias, Ted Painter, Venkatraman Atti, Audio Signal Processing and Coding, Hardcover 544 pages, ISBN: 0-471-79147-4, Wiley, 2007.

Acknowledgement

This work was funded in part by NSF TUES Phase 3 grant 0817596 and by the ASU SenSIP Center.

Biographical Information

Shalin Mehta received the B.E. degree in Electronics and Communications Engineering from Sardar Patel University, Gujarat, India in 2005 and the M.S. degree in Electrical Engineering from Arizona State University, Tempe, in 2010. His major concentration was in the areas of Signal and Image processing. Work on J-DSP software gave him a perfect opportunity to pursue his passion of developing educational software for students in the area of signal processing. Presently, he works at Intel Corporation developing multi-media software drivers for Intel integrated graphics chips.

Andreas Spanias is a Professor in the School of Electrical, Computer, and Energy Engineering at Arizona State University (ASU). He is also the founder and director of the SenSIP industry consortium (NSF I/UCRC). His research interests are in the areas of adaptive signal processing, speech processing, and audio sensing. He and his student team developed the computer simulation software Java-DSP (J-DSP - ISBN 0-9724984-0-0). His team received the PREMIER award at 2012 FIE for their app iJDSP. He is the author of two text books: Audio Processing and Coding by Wiley and DSP; An Interactive Approach. He and his team co-authored several research monographs with the Morgan and Claypool publishers. He served as Associate Editor of the IEEE Transactions on Signal Processing and as General Co-chair of IEEE ICASSP-99. He also served as the IEEE Signal Processing Vice-President for Conferences. He is co-recipient of the 2002

IEEE Donald G. Fink paper prize award and was elected Fellow of the IEEE in 2003. He served as Distinguished lecturer for the IEEE Signal processing society in 2004.

Jayaraman J. Thiagarajan received his Masters and Doctorate degrees in Electrical Engineering from Arizona State University. He is currently a postdoctoral researcher in the Center for Applied Scientific Computing at Lawrence Livermore National Laboratory. His research interests are in the areas of machine learning, sparse modeling, and computer vision. He has worked extensively on the MPEG-1 Layer-III audio coding standard and has published a monograph on this topic. In addition, he has contributed to the Java-DSP software package used in undergraduate DSP education. He has also served as a reviewer for several IEEE and Elsevier journals and conferences.

Mahesh K. Banavar (S'08–M'11) received the B.E. degree in telecommunications engineering from Visvesvaraya Technological University, Karnataka, India, in 2005 and the M.S. and Ph.D. degrees, both in electrical engineering, from Arizona State University, Tempe, in 2007 and 2010, respectively. He is currently a postdoctoral research associate at Arizona State University, specializing in Signal Processing and Communications, and doing research in wireless communications and sensor networks. Dr. Banavar is a recipient of the Teaching Excellence Award from the Graduate and Professional Student Association at Arizona State University. He is also a member of MENSA and the Eta Kappa Nu honor society.

Karthikeyan Natesan Ramamurthy completed his M.S. and Ph.D. degrees in Electrical Engineering at Arizona State University. Currently, he is a research staff member in the Business Analytics and Mathematical Sciences department at the IBM Thomas J. Watson Research Center in NY. His research interests are in the areas of machine learning, signal modeling, data analytics and computer vision. He has also worked on developing a Java toolbox for signal processing and spectral

analysis in Earth systems and geological sciences. He has served as a reviewer for several IEEE and Elsevier journals and conferences.

Robert Santucci obtained his Bachelor of Science degrees in both electrical engineering and in computer science from Embry-Riddle Aeronautical University in 2000. He obtained a Master of Science degree in electrical engineering from Arizona State University in 2008. He is currently a Ph.D. candidate at Arizona State University in the School of Electrical, Computer, and Energy Engineering. He has worked in industry doing wireless transceiver and power-management integrated circuit design and verification for cellular products. He has also worked in high-speed serial interconnect design for servers. His current research interests include amplifier linearization, wireless transceivers, and distributed estimation.

Constantinos S. Pattichis is currently a Professor with the Department of Computer Science of the University of Cyprus. His research interests include ehealth and mhealth, medical imaging, biosignal analysis, life sciences informatics, and intelligent systems. He has published 80 refereed journal and 186 conference papers, and 27 chapters in books in these areas. He is Co-Editor of the books *M-Health: Emerging Mobile Health Systems*, and of the *Ultrasound and Carotid Bifurcation Atherosclerosis*, published by Springer in 2006, and 2012 respectively. He was Guest Co-Editor of 11 journal Special Issues including the more recent ones on *Atherosclerotic Cardiovascular Health Informatics*, and *Citizen Centered e-Health Systems in a Global Health-care Environment*, of the IEEE Trans. on Information Technology in Biomedicine. He was General Co-Chairman of the *IEEE 12th International Conference on BioInformatics and BioEngineering (BIBE2012)*, and the *IEEE Information Technology in Biomedicine (ITAB09)*. Moreover, he serves as Distinguished Lecturer of the IEEE EMBS, an Associate Editor of the IEEE Journal of Biomedical and Health Informatics, and on the Editorial Board

of the Journal of Biomedical Signal Processing and Control. He is a Fellow of IET, and Senior Member of IEEE.

Photini Spanias completed her Education Doctorate at Arizona State University in 1996. She is currently a Senior Lecturer in the Division of Teacher Preparation at the Mary Lou Fulton Teachers College (MLFTC) at Arizona State University. Her research interests are in the areas of Math Methods. She teaches undergraduate and graduate courses in Mathematics Education, Algebraic and Geometric thinking and assessment. She is a member of NCTM. She consults with Engineering and Mathematics education specialists in the preparation of education and assessment materials and web content.

Harish Krishnamoorthi received his M.S. and Ph.D. degrees in electrical engineering from Arizona State University in 2007 and 2011 respectively. From May 2008 to August 2008, he was a visiting researcher at Eindhoven University of Technology, Eindhoven, The Netherlands. Since 2011, he has been working for the R & D division of Cochlear Americas developing efficient audio processing algorithms. His primary research interests are in the areas of psychoacoustics, auditory models, speech and audio coding, speech enhancement and auditory scene analysis.