Development of Signal Processing Online Labs using HTML5 and Mobile platforms

Abhinav Dixit[1], Sameeksha Katoch[1], Photini Spanias[2], Mahesh Banavar[3], Huan Song[1], Andreas Spanias[1]

[1]SenSIP Center, School of ECEE, [2]MLFTC, ASU, [3]Dept. of ECE, Clarkson University

spanias@asu.edu

***Abstract* - Several web-based signal processing simulation packages for education have been developed in a Java environment. Although this environment has provided convenience and accessibility using standard browser technology, it has recently become vulnerable to cyber-attacks and is no longer compatible with secure browsers. In this paper, we describe our efforts to transform our award-winning J-DSP online laboratory by rebuilding it on an HTML5 framework. Along with a new simulation environment, we have redesigned the interface to enable several new functionalities and an entirely new educational experience. These new features include functions that enable real-time interfaces with sensor boards and mobile phones. The Web 4.0 HTML5 technology departs from older Java interfaces and provides an interactive graphical user interface (GUI) enabling seamless connectivity and both software and hardware experiences for students in DSP classes.**

Keywords—Signal Processing, HTML5, Web 4.0 Engineering Education, sensor interfaces, real-time online labs.

I. INTRODUCTION

At Arizona State University (ASU), DSP-related courses are part of the accredited online engineering degree offered by the ASU Online/EdPlus network and have worldwide reach. These programs have created new needs for online lab access and enable enriched education experiences with supplemental materials including streaming video and interactive simulations. In order to provide on-line laboratory experiences to distance learners, we have previously developed and tested the award-winning Java-DSP (J-DSP) tool [1-2] for use in DSP and signals and systems classes. With the development of web-based technologies over the last two decades, several of web-based signal processing simulation packages for education [3-4, 12-15, 17-20] have been proposed. The Java environment used to develop these technologies provided convenience and accessibility using standard browser technologies for a long period. However, the limited compatibility of browsers with Java has had a negative effect in disseminating Java courseware.

Our award winning J-DSP online laboratory [6-10.33], which was developed as a JAVA applet, provided a valuable online resource for DSP simulation for several years. However, several browsers do not support JAVA applets. Elevated requirements for online education content and limited access in older applet technologies have motivated rebuilding and enhancing online simulation tools in an entirely new and secure framework. We have developed, from the ground up, a new online tool based on Web 4.0 HTML5 technologies. HTML5 is compatible with most or all modern browsers and enables connectivity and access to several new interactive tools [30]. Several aspects of this new HTML5 J-DSP education environment are shown in Figure 2.

HTML5 is a markup language used for structuring and presenting content on the World Wide Web. The new software now can be opened in the browser as any other webpage and it doesn't need any dependencies on external applications, making our software browser friendly. Being a browser-based environment, HTML5 provides compatibility with JavaScript and Cascading Style Sheet (CSS). Using these techniques and the opportunity to renovate the software we came up with a better visual and user-friendly environment.

With no browser connectivity limitations and with an environment similar to a web-site, our interactive software can be securely connected to the Internet. We plan to develop interfaces in the software to interact with hardware platforms such as Arduino™, Raspberry Pi, and NXP sensor boards, and also with mobile devices such as Android phones and iPhones. This feature will give the software a new dimension where a user can interact perform real-time experiments with hardware. Data acquisition can further be developed to enable running machine learning algorithms. The ability to track parameters and the performance of algorithms as applied to data gathered from remote devices will provide students experience in new areas involving Internet of Things (IoT) applications.

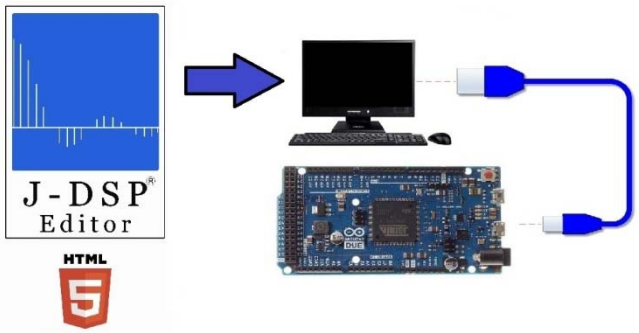

Figure 1. HTML5 based DSP software interface with hardware platforms.

978-1-5090-5920-1/17/$31.00 ©2017 IEEE

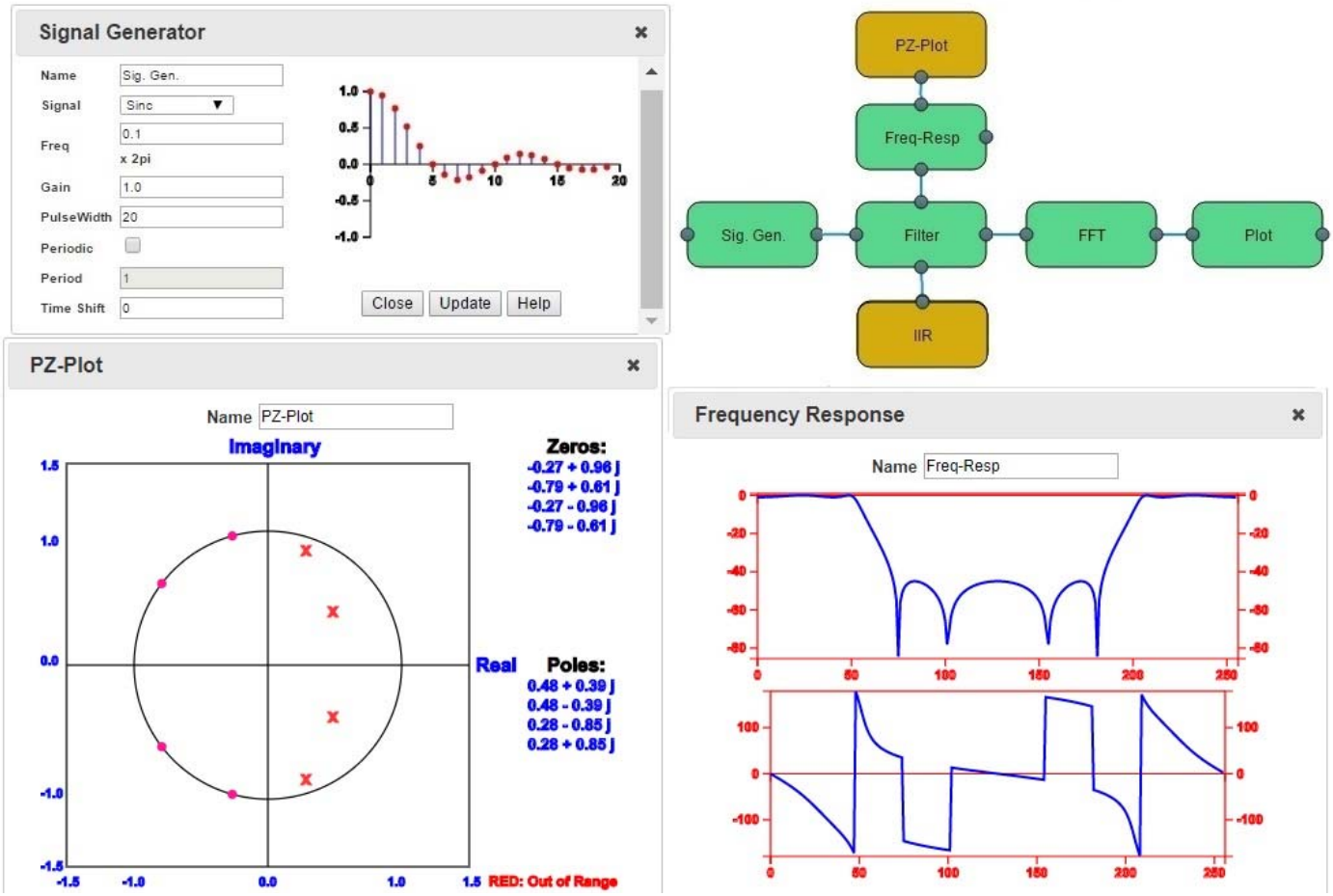

Figure 2. Implementation of Pole-Zero and Frequency response block in HTML5 based J-DSP

II. ARCHITECTURE

The front end of the software is completely developed inHTML5. This gives the software a user-friendly and interactive environment when compared to designing the software in other languages. The application we have developed is similar to a web-site interface and all the processing is done at each instance and then discarded when the program is closed. All the data signals generated, processed or displayed are present in the browser interface until they are required and discarded as soon as they are no longer used. Therefore, there is no dependence on any other external application such as an applet. The maximum size of the array that can be generated in Java applets is 256. Using HTML5 gives the software an ability to generate signals with samples in the range of thousands and perform complex DSP operations. Therefore, DSP operations such as the FFT and the frequency response function can be executed with high resolution. This will lead to more accurate calculations, and make it easier for the students to understand and interpret spectral representations of the signals. Fig. 2 describes the flow of the code and the architecture used to develop this HTML 5 DSP software package. HTML5 gives us the freedom to interact with different structures such as JavaScript, jQuery, and Cascade Style Sheets (CSS) [30]. CSS provides a simple and appealing GUI and produces animations that make the software more interactive and user-friendly. On the other hand, JavaScript and jQuery gives a powerful environment to perform complex DSP operations. The back-end, that is, the processing part of our new software has been primarily developed in JavaScript.

JavaScript is a high-level, dynamic, and interpreted run-time language. It has been standardized in the ECMA Script language specification. Alongside HTML and CSS, JavaScript is one of the three core technologies of World Wide Web content production; a majority of websites employ it, and all modern Web browsers support it without the need for plug-ins [11]. Since there are no plug-in dependencies, the software is an independent website that can work on its own without interruption and external application requirements. The object-oriented nature of HTML5 also provides us an ability to interact with the large set of libraries already available. Just by importing a specific library, its functionalities and features can be used in our application. This can be used to develop classes that represent a particular type of object. We can then use this object to perform DSP operations on the data it contains. The blocks in our software are individual objects and although being of the same type, *i.e.*, a block, they can be used to perform different operations.

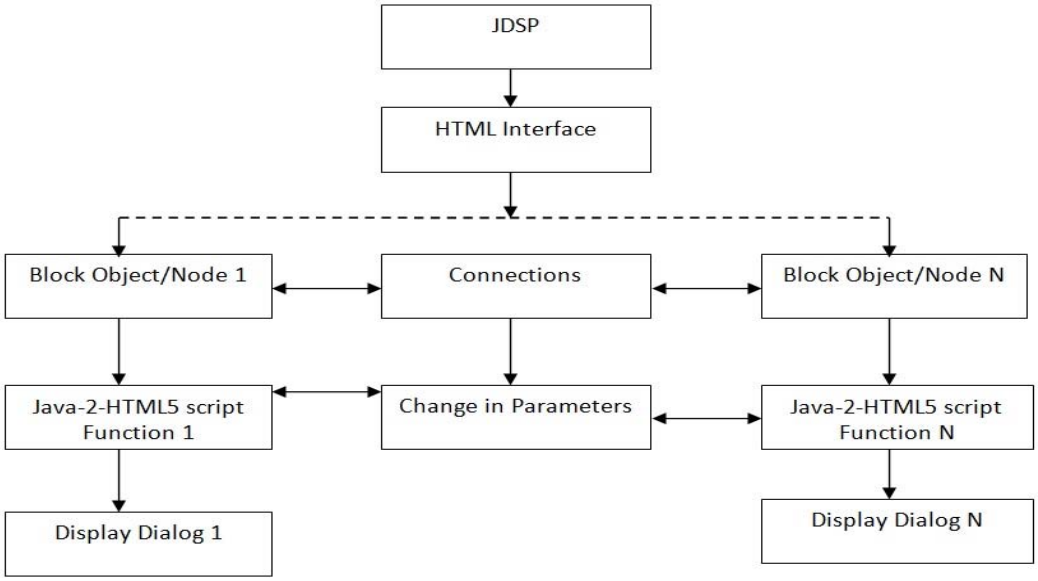

Figure 3. HTML5 DSP software code flow and architecture.

The blocks can be connected in different configurations in our software and therefore used for several DSP operations. Furthermore, the memory utilization is optimized, which makes the processing of complex algorithms faster.

III. ENVISIONED SPEECH AND AUDIO PROCESSING FUNCTIONS

By using HTML5, large sets of data can be processed at a time. Due to this advantage, we plan on using a function in our new software that can acquire speech samples in real-time using mobile devices like Android phones and iPhones [21-25]. This will give our software an ability to perform real-time operations with real-life speech signals [26,34,35]. Through this function, we'll be able to capture and track the changes in parameters of the speech.

We can also use the speech data acquired using our software with different functions in our software to extract useful signal information. For example, the PZ-Placement block [2] in JDSP provided a unique feature where one can add or remove poles or zeros on the *z*-plane and observe the change in the frequency response. We use this feature to manipulate real-time signals and audio files. For example, the changes in the filter using the pole-zero placement function in HTML 5 can now be observed in conjunction with audio processing on a hardware platform in real-time. The hardware platform will not require any dedicated code as our software provides the necessary interfaces.

IV. ENVISIONED MACHINE LEARNING FUNCTIONS

Another planned feature is to introduce in the software a learning model. The software will learn the habits and requirements of the users and make suggestions according to their personal learning styles. Moreover, HTML5 allows for enhanced analytics, better graphics, low power video streaming, and enables several new methods for tracking student performance.

Our software now has the ability to acquire data from remote devices. It can also process more data at the same time. This ability can be used for complex machine learning algorithms. The data acquired by the remote devices or hardware can be used to train a model using machine learning algorithms [27] such as *K*-means clustering. This trained model can then be used to perform real-time operations over the new data acquired from the users. This will make our software a real-time application, where we can gather sensor data from mobile phones using a mobile app installed on users' mobile phones.

We aim to use this ability to perform classification over the data acquired by habits of users and monitor health activities and performance of the user. Our software will now be able to detect any abnormalities in the activities and report that to the user.

V. ENVISIONED HTML5 ENABLED LABS AND IOT FEATURES

In addition to improved aesthetics, HTML5 J-DSP can interface to various hardware platforms such as the Arduino™, Raspberry Pi, and the NXP sensor boards [28]. The Web 4.0 environment that we have used to build the HTML5 online DSP lab provides several opportunities to add new interfaces and several other modern features. The software can now communicate with other devices including computers, Android phones, and iPhones. With this interface, we can create a channel through which learners can visualize

and experiment with real life signals and implement real-time DSP applications using phones and laptops. Figure 3 shows the capability to connect HTML5 DSP software to remote devices such as laptops, Android phones, and iPhones.

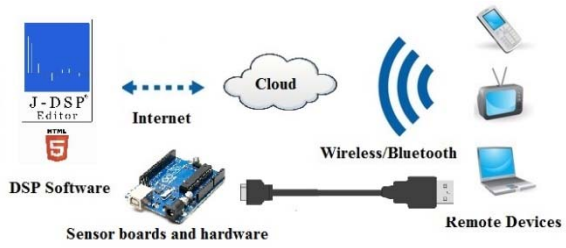

Figure 4. The interaction between DSP software and remote devices using wireless communications (Internet, Cloud, and Bluetooth) and sensor boards.

An additional enhancement over JAVA applets includes the ability to seamlessly work with expanded signal arrays and matrices of large dimensionality. With the Java constraints removed, we are able to process more than 20000 samples at a time. These expanded signal analysis capabilities allow us to demonstrate more clearly, frequency resolution and spectral leakage in Fourier analysis and more complex time-frequency graphics such as spectrograms, which were previously limited with JAVA applets. In addition, certain advanced functions such as array processing [31], GPS [32], and Kalman filtering [29] can be demonstrated with the expanded data processing capability.

VI. PLANNED ASSESSMENTS

J-DSP assessment will be made through online forms and feedback from the students using the software in courses. The online exercises and labs performed on J-DSP software will automatically store securely and privately qualitative as well as quantitative data over the network. Feedback forms from students are first collected for general assessment. They assess the learning experience while using the software and specific lab functions. Additional assessment based on pre- and post- exercises aim to determine student learning of specific topics.

The evaluation forms can be accessed securely with passwords through the J-DSP website. The users fill out and submit such forms on the Internet. The feedback data can also be accessed by links provided on-line.

VII. REMARKS

The HTML5 DSP software allows users to use a secure user-friendly environment, designed with latest web technologies. This gives students new opportunities to interact with various hardware and software platforms available for data processing or acquisition. Since connectivity with the Internet is of vital importance, the capability of our software to interact securely with several browsers make it more accessible and easy to use. The ability to interact with hardware and sensor boards provides opportunities for real-time hardware experiments. The new interface will also enable learning student habits and adapt the simulation environment. The data from the activities of the users can be used to track and assess student learning.

VIII. ACKNOWLEGEMENTS

The work at Arizona State University is supported in part by the NSF DUE award 1525716 and the SenSIP Center. The work at Clarkson University is supported in part by the NSF DUE award 1525224.

REFERENCES

[1] A. Spanias et al, "Development and Evaluation of a Web-Based Signal and Speech Processing Laboratory for Distance Learning", *Proc IEEE ICASSP-2000*, Istanbul, Vol. 6, pp. 3534-3537, June 2000.

[2] Huang, C.; Thiagarajan, J. J.; Spanias, A.; Pattichis, C.;, "A Java-DSP interface for analysis of the MP3 algorithm," *Proc. of the IEEE DSP/SPE Workshop*, pp.168-173, 4-7 Jan. 2011.

[3] Thomas A. Baran, Richard G. Baraniuk, Alan V. Oppenheim, Paolo Prandoni, and Martin Vetterli, "MOOC adventures in signal processing: Bringing DSP to the era of massive open online courses," *IEEE Signal Processing Magazine,* vol. 33, no. 4, pp. 62–83, 2016.

[4] "Ruby on Rails," http://rubyonrails.org.

[5] Garvit Juniwal, Alexandre Donze´, Jeff C. Jensen, and Sanjit A. Seshia, "CPSGrader: Synthesizing temporal logic testers for auto-grading an embedded systems laboratory," in *Proceedings of the 14th International Conference on Embedded Software - EMSOFT '14.* ACM Press, 2014

[6] A. Spanias and V. Atti, "Interactive On-line Undergraduate Laboratories Using J-DSP," *IEEE Trans. on Education Special Issue on Web-based Instruction*, vol. 48, no. 4, pp. 735-749, Nov. 2005.

[7] Karthikeyan Natesan Ramamurthy, Linda A. Hinnov, and Andreas S. Spanias (2014) Teaching Earth Signals Analysis Using the Java-DSP Earth Systems Edition: Modern and Past Climate Change. *Journal of Geoscience Education:* Vol. 62, No. 4, pp. 621-630, November 2014.

[8] S. Mehta, A. Spanias, J.J. Thiagarajan, M.K. Banavar, K.N. Ramamurthy, R. Santucci, C. Pattichis, P. Spanias, H. Krishnamoorthi, "An Integrated Graphical Environment for Web-based Learning," *Computers in Education Journal*, vol. 5, no. 1, , January-March 2014.

[9] A. Spanias, Digital Signal Processing; An Interactive Approach – 2nd Edition, 403 pages, Textbook with JAVA exercises, ISBN 978-1-4675-9892-7, Lulu Press On-demand Publishers Morrisville, NC, May 2014.

[10] B. Santucci and A. Spanias, "Java-DSP Functions for Power Amplifier Linearization Techniques," *ASEE Computers in Education Journal*, Winner of the Harden-Simons Prize Award 2014.

[11] *Flanagan, David (2011). JavaScript: The Definitive Guide (6th ed.).* O'Reilly & Associates. ISBN 978-0-596-80552-4 *Guide (6th ed.).* O'Reilly & Associates. ISBN 978-0-596-80552-4.

[12] R. Owston, "The World Wide Web: A Technology to

Enhance Teaching and Learning?", *Educational Research*, vol. 26, no. 2, pp. 27-33, March 1997.

[13] V. L. Stonick, "Teaching Signals and Systems Using the Virtual Laboratory Environment at ECE at CMU", *Proc. 1993 IEEE Int. Conf. on Acoustics Speech and Signal Processing*, pp. 36-39, 1993-April.

[14] T. B. Welch, C. H. G. Wright, and M. G. Morrow, "Experiences in offering a DSP-based communication laboratory," in Proceedings of the 11th IEEE Digital Signal Processing Workshop and the 3rd IEEE Signal Processing Education Workshop, (Taos, NM (USA)), Aug. 2004.

[15] C. H. G. Wright, T. B. Welch, D. M. Etter, and M. G. Morrow, "A systematic model for teaching DSP," in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. IV, pp. 4140-4143, May 2002.

[16] E. A. Lee and P. Varaiya. Signals and Systems. Addison Wesley, 1st edition, 2003.

[17] Wang J., Liu L., Jia W. (2005) The Design and Implementation of Digital Signal Processing Virtual Lab Based on Components. In: Lau R.W.H., Li Q., Cheung R., Liu W. (eds) Advances in Web-Based Learning – ICWL 2005. ICWL 2005. Lecture Notes in Computer Science, vol 3583. Springer, Berlin, Heidelberg.

[18] A. Kalantzopoulos, E. Zigouris, "Online Laboratory Sessions in System Design with DSPs using the R-DSP Lab", *iJOE*, vol. 10, no. 4, 2014.

[19] N. Sousa, G.R. Alves, M.G. Gericota, "An Integrated Reusable Remote Laboratory to Complement Electronics Teaching", Learning Technologies *IEEE Transactions on education,* vol. 3, no. 3, pp. 265-271, 2010.

[20] Z. Dvir, "Web-based remote digital signal processing (DSP) laboratory using the Integrated Learning Methodology (ILM)", *IEEE*, 2003.

[21] Andreas Spanias, Ted Painter, Venkatraman Atti, Audio Signal Processing and Coding, Hardcover 544 pages, ISBN: 9780471791478 and 0-471-79147-4, Wiley, Textbook with theory, problems, and MATLAB computer exercises. March 2007.

[22] Ted Painter and Andreas S. Spanias, "Perceptual Coding of Digital Audio," *Proceedings of the IEEE*, pp. 451-513, Vol. 88, No.4, April 2000.

[23] Makhoul J. Linear prediction: a tutorial review. Proc. IEEE 63:561-80, 1975. IBolt Beranek and Newman Inc., Cambridge, MA)

[24] L. R. Rabiner, R. W. Schafer, Digital Speech Processing, NJ, Englewood Cliffs: Prentice Hall, 1978.

[25] C. H. G. Wright, T. B. Welch and M. G. Morrow, "Signal processing concepts help teach optical engineering," *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6275-6279. Shanghai, 2016.

[26] A.S. Spanias, "Speech Coding: A Tutorial Review," Proceedings of the IEEE, Vol. 82, No. 10, pp. 1441-1582, Oct. 1994

[27] V. Berisha, A. Wisler, A. Hero, A. Spanias, "Empirically Estimable Classification Bounds Based on a Nonparametric Divergence Measure," I*EEE Transactions on Signal Processing,* vol. 64, no. 3, pp.580-591, Feb. 2016.

[28] J. Lee, M. Stanley, A. Spanias, C. Tepedelenlioglu, "Integrating machine learning in embedded sensor systems for Internet-of-Things applications," *DOI: 10.1109/ISSPIT.2016.7886051IEEE ISSPIT*, Limassol, Dec. 2016.

[29] Narayan Kovvali, Mahesh Banavar, A. Spanias, An Introduction to Kalman Filtering with MATLAB Examples, Synthesis Lectures on Signal Processing, , Morgan & Claypool Publishers, Editor Jose Mura, vol. 6, no. 2, pp. 1-81, ISBN 13: 9781627051392, September 2013. [30] "HTML5 Differences from HTML4 – APIs". World Wide Web Consortium.

[31] Jeff Foutz, A. Spanias, M. Banavar, Narrowband Direction of Arrival Estimation for Antenna Arrays, Synthesis Lectures on Antennas, Morgan & Claypool Publishers, Research Monograph, ISBN-13: 978-1598296501, Aug. 2008.

[32] Steve Miller, Xue (Sophia) Zhang, A. Spanias, Multipath Effects in GPS Receivers, Synthesis Lectures on Communications, Morgan & Claypool Publishers, ISBN 978-1627059312, 70 pages, Ed. William Tranter, Vol. 8, No. 1 , Pages 1-70, December 2015.

[33] Jinru Liu; Banavar, M.K.; Thiagarajan, J.J.; Ramamurthy, K.N.; Shuang Hu; Xue Zhang; "WIP-Interactive signal-processing labs and simulations on iOS devices, *IEEE FIE 2011*, Oct, 2011.

[34] V. Atti, Algorithms and Software for Predictive and Perceptual Modeling of Speech, Morgan & Claypool Publishers, March 2011.

[35] Visar Berisha, Steve Sandoval, Julie Liss, Bandwidth Extension of Speech Using Perceptual Criteria, Morgan & Claypool Publishers, ISBN: 9781627053136, October 2013.