Online Machine Learning Experiments in HTML5

Abhinav Dixit[1], Uday Shankar Shanthamallu [1], Andreas Spanias [1], Visar Berisha [1], Mahesh Banavar[2]
[1]SenSIP Center, School of ECEE, ASU, [2]Dept. of ECE, Clarkson University
spanias@asu.edu

***Abstract* – This work in progress paper describes software that enables online machine learning experiments in an undergraduate DSP course. This software operates in HTML5 and embeds several digital signal processing functions. The software can process natural signals such as speech and can extract various features, for machine learning applications. For example in the case of speech processing, LPC coefficients and formant frequencies can be computed. In this paper, we present speech processing, feature extraction and clustering of features using the K-means machine learning algorithm. The primary objective is to provide a machine learning experience to undergraduate students. The functions and simulations described provide a user-friendly visualization of phoneme recognition tasks. These tasks make use of the Levinson-Durbin linear prediction and the K-means machine learning algorithms. The exercise was assigned as a class project in our undergraduate DSP class. The description of the exercise along with assessment results is described.**

Keywords—Machine Learning, Speech recognition, Linear Predictive Coding, Online labs.

I. INTRODUCTION

Machine Learning (ML) and internet-of-things (IoT) applications are becoming mainstream with several products including the Amazon Echo, Google Home, Apple HomePod™ entering households. These products embed advanced signal processing, cloud computing, sensor arrays, and machine learning which enable voice recognition with elevated accuracy. This has motivated introducing some of these advanced concepts, which were previously covered in graduate courses, to undergraduates. We choose to "inject" such advanced topics in Digital signal processing (DSP) courses which are typically offered in the junior or senior level of most undergraduate Electrical and Computer Engineering (ECE) programs [1-9].

The ECE DSP course typically covers signal processing basics including filter design, spectral estimation, Fast Fourier Transform (FFT), multirate signal processing, random signal analysis, etc. For several years now our University embedded online labs [10] in the DSP course which is also part of our online degree program. The online simulations include graphical visualization of various DSP functions. The online labs have been enabled by the Java-DSP software [10]. Because of security and compatibility concerns with Java, this system was recently rebuilt from the ground up for operation in HTML5. This online simulation environment is called JDSP-HTML5 [11]. This software enables online simulations on any modern web browser supporting the latest Web 4.0 HTML5 technologies.

Previous J-DSP work addressed applications and laboratories in several areas and disciplines. Speech processing using J-DSP was presented in [12] and audio processing was described in [13, 14]. A study in earth systems and geological applications was published in [15]. Real-time implementations using DSP chips were described in [16]. Award-winning iPhone (iOS) and Android platform implementations were presented in [17] and [18], respectively. The new implementation in HTML5 provided improved software security, speed, and compatibility with multiple browsers. In this paper, we developed machine learning functions and used them to introduce the concepts of speech and phoneme recognition in the undergraduate DSP course. Using these functions, students can develop simulations of speech processing, linear predictive coding, feature extraction, machine learning training, and speech phoneme classification. The Levinson Durbin algorithm [19-21] is embedded in a block called LPC (Fig. 1).

A speech processing simulation starts with acquiring speech data. The speech files can be accessed and processed frame-by-frame using the SigGen(L) block (Fig. 1). Each frame of speech is processed with a selected window and then passed to the LPC+ block where autocorrelations are estimated and linear prediction coefficients are obtained. Students can visualize the vocal tract spectral envelope and observe its resonant modes (formants [19, 22]). The processing with HTML5 allows longer windows than the previous Java version. A computer project exercise was formed using the LPC and k-means training [23, 24] and clustering functions. The exercise allows the students to train the k-means algorithm and subsequently cluster the formants. Correct clustering should lead to correct classifications of phonemes. Students are able to change the number of LPC parameters, the numbers of clusters, define the frame sequence, and the overlapping of data windows. Students are also able to train, validate and cross-validate the clustering process. They also have opportunities to examine the phoneme classification performance in the presence of additive noise.

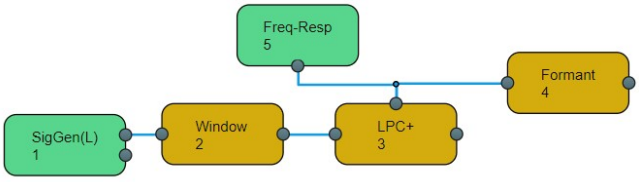

Figure 1: Speech Feature Extraction on a frame by frame basis. Formants of speech are extracted using the LPC block and *z*-plane analysis embedded in the formant block.

The motivation for this exercise was to provide multiple advanced signal processing experiences to undergraduate students including: a) computing LPC parameters and their significance, b) extracting the formants from LPC parameters, c) associating formants with phonemes, d) using machine learning to train an algorithm to associate formants with phonemes, and e) clustering formants and automatically recognize phonemes. To achieve this, we developed new speech processing functions

978-1-5386-1174-6/18/$31.00 ©2018 IEEE

in JDSP-HTML5. These functions support a computer project exercise assigned to students in DSP course at ASU. The exercise and student knowledge gained was assessed using a pre-quiz and post-quiz. The assessment consisted of several multiple-choice questions on LPC and ML. We provide details about the exercise and assessment in the subsequent sections. The rest of the paper is organized as follows: Section II describes the new functions of JDSP-HTML5. Section III describes the k-means ML algorithm, Section IV describes the new blocks, Section V describes the computer exercise, Section VI describes the assessment and Section VII provides concluding remarks.

II. FORMANTS AND SPEECH RECOGNITION

The first block shown in Figure 1 is the signal generator block. The dialog for signal generator block in shown in Fig. 2. Students can use this block to process data frame-by-frame. The user can select among several stored natural signals and visualize specific time frame ranges in the signal. A time-domain plot of the current frame enables students to define the range of the signal to be examined. The new blocks developed in JDSP-HTML5 are the formant block and the k-means block. Within the k-means block we included a convergence plot of Mean-Square Error (MSE) Vs Iteration.

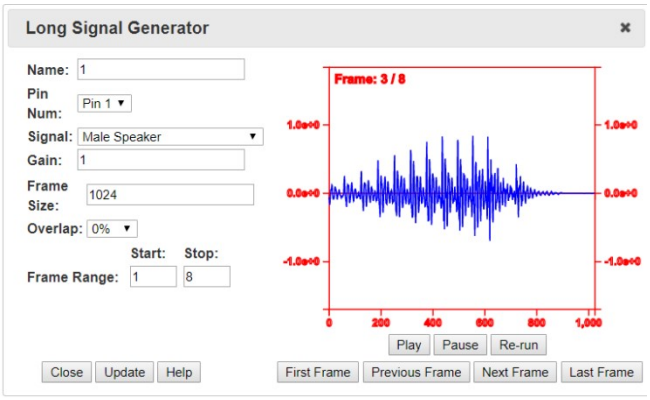

Figure 2: Frame-by-frame processing of speech and other signals using the SigGen(L) block in JDSP-HTML5.

Students using the signal generator can define the frame size according to the signal stationarity range. Typically, in speech processing, 20ms frames for 8 kHz sample speech are used though there are also other signal lengths that have been defined before. Users can also choose start and stop frames for processing. For example, one can focus on examining frames over the range of a phoneme or a word depending on the application. Overlapping windows can also be programmed – a process that is typical in many speech processing applications.

To implement speech recognition in JDSP-HTML5, we developed DSP functions to compute linear prediction coefficients and extract formant frequencies from the speech signal [25, 26]. The formant frequencies are the resonant frequencies of the vocal tract which associate with phonemes. The block diagram for speech recognition that includes the k means ML algorithm is shown in Fig 3.

The signal generator in Fig. 2 contains audio signals including phonemes, sentences, noise etc. In this example, we chose phoneme 'a' for processing. The speech signal is processed frame-by-frame with a frame size of 160 samples. Each speech frame is passed through the window block where it is windowed using a Hamming window. The windowed signal is then passed to the LPC+ block for estimation of the LPC coefficients (the autocorrelation computation is embedded in the LPC+ block).

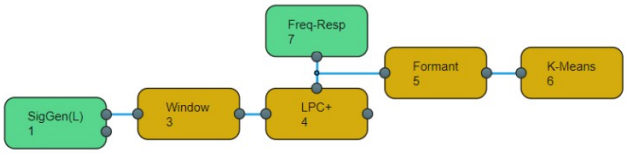

Figure 3: Block diagram for LPC and formant extraction with k-means clustering.

Linear predictive coding is used in speech compression and other applications. Linear prediction is a process where the difference (error) between the current sample of the data is estimated by a linear combination of past samples. The error is minimized in the mean square sense and the prediction coefficients are estimated by solving a system of autocorrelation equations. In time series theory, these are called the Yule-Walker equations. Typical LPC order for compression applications is 10, though in the GSM telephony standard algorithm the order is 8. For other applications, the order may vary from 8 to 14. Introducing LPC in a DSP class offers several opportunities that have pedagogical value including learning about: a) inverse filters. b) spectral envelopes, c) sensitivity and stability of filters, d) lattice filter structures, e) poles of LPC synthesis filters and association with formants, e) relation between the prediction residual and the glottal pulse, and f) parametric models for speech analysis, synthesis, recognition, and compression. In Figure 4, we show the JDSP-HTML5 plot of the LPC spectral envelope magnitude and phase.

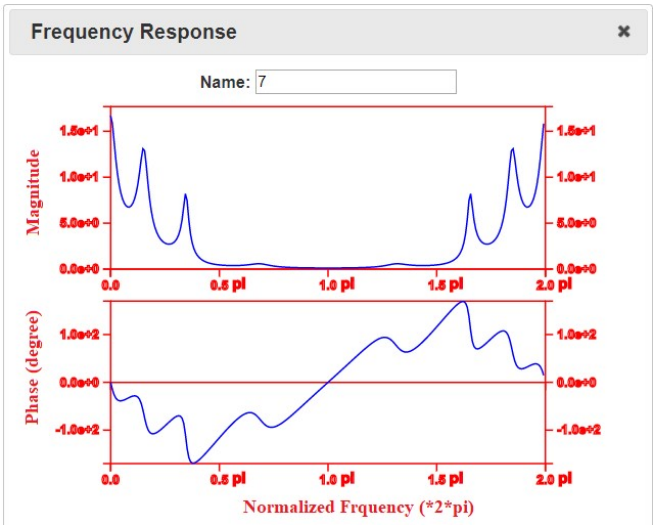

Figure 4: LPC spectral envelope of the 50th frame for phoneme 'a' speech signal.

The peaks of the magnitude envelope are estimates of the formants of speech which can be used to synthesize or recognize phonemes. The LPC coefficients estimated from the LPC block can be used to determine the formant frequencies using the Formant block in JDSP-HTML5. The formant block calculates the first two formant frequencies of the speech signal and stores them. The formants are calculated using z plane calculations to identify the two highest peaks in the spectral envelope. The frequencies corresponding to the two highest peaks are estimates of the first and second formants respectively. The formants are different for each phoneme and formant frequencies for female speakers are slightly higher. The average formant frequencies for a typical male speaker are given in Table I [28].

Phoneme	Formant 1	Formant 2
'ɑ'	560	920
'**a**'	560	1480
'**i**'	280	2620
'**u**'	320	920

Table I: Average (ideal) formant frequencies for males.

III. THE K-MEANS ALGORITHM

Previous education studies on machine learning have appeared in the literature, namely [31-34]. In this section, we provide a general overview of the K-means algorithm implemented in JDSP-HTML5. The k-means algorithm is an unsupervised machine learning algorithm. The k-means algorithm is an intuitive algorithm that uses a similarity metric to group data. The Euclidean distance is often used as a similarity metric. The algorithm identifies k-clusters in the data. In Figure 5, we show an example with 2-d vectors (x1, x2) in three different data clusters and their centroids formed using the k means algorithms. The data that will be used here is speech and the k-means will be used to form identifiable formant clusters representing different phonemes.

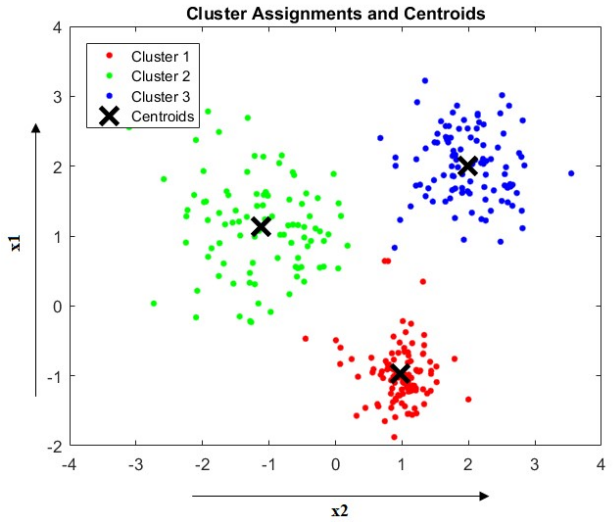

Figure 5: Clustering data consisting of 3 different distributions using the k-means algorithm. The means of the three clusters are (2, 2), (-1, 1) and (1, -1) respectively. The variances of the three clusters are 0.5, 0.8 and 0.3 respectively.

IV. JDSP FORMANT AND K-MEANS BLOCKS

The formant block in JDSP-HTML5 accumulates the formant pairs for each frame that has been formed using the signal generator block. The formant values can be seen in the formant block dialog. A sub-set of the formants of 50 consecutive frames of the phoneme 'ɑ' are shown in Figure 6. These accumulated formants can then be passed to the K-means block and can be further processed for clustering.

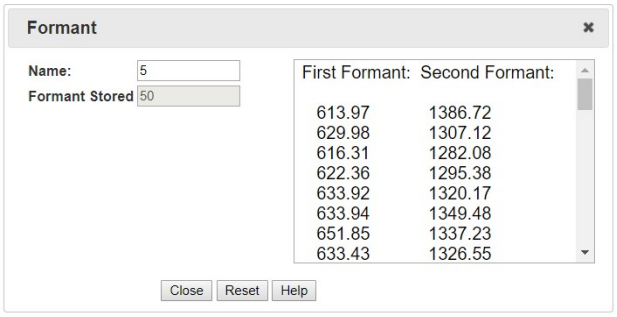

Figure 6: First and second formant frequencies stored within formant block and JDSP-HTML5

Figure 7 shows the K-means block result distinguishing 4 phonemes i.e. 'ɑ', '**a**', '**i**' and '**u**'. The K-means block clusters the calculated formants from 4 different speech signals into 4 clusters and provides the within-cluster sum of squares convergence rate. The normalized MSE is basically a measure of variance. The centroid values that are calculated by the K-means algorithm is an approximation of the means of the corresponding formant.

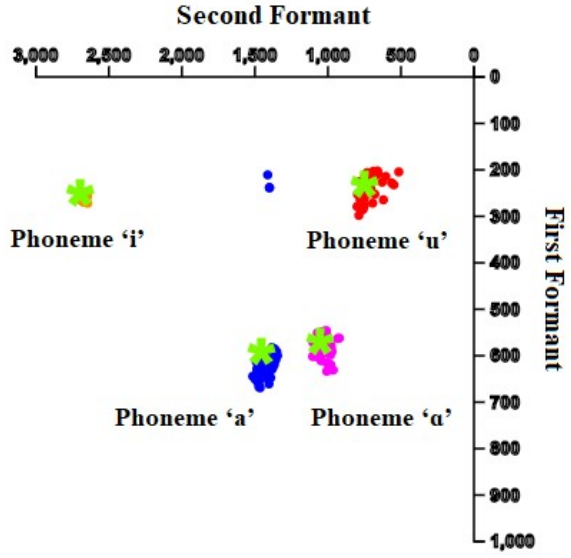

Figure 7: K-means clustering 4 phonemes: 'ɑ', 'a**', '**i**' '**u**'. The means are stated later in Fig. 8.**

The K-means block also shows the final centroid values for all clusters i.e. mean formants pairs for each phoneme as shown in Figure 8. It is to be noted that the calculated centroid values are close to the true values of the vowel formants in Table I.

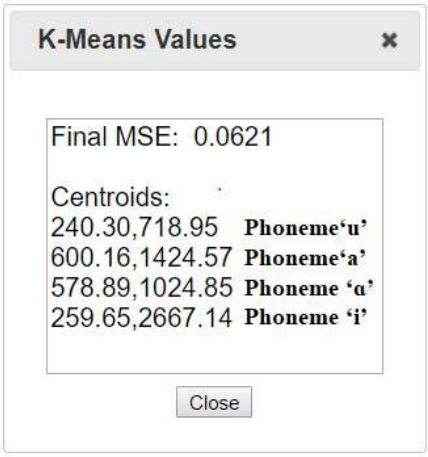

Figure 8: Formant centroids calculated by K-means. The values associated with the means in Fig. 7

V. CLUSTERING EXERCISE DESCRIPTION

We developed a training exercise using the JDSP-HTML5 in which we perform linear predictive analysis to estimate the spectral envelope of speech. We then calculate formants and finally, we cluster the formants and obtain their centroids. We assigned this exercise in two parts as a class project. The first part of the exercise familiarizes the students with the basics of LPC and speech formant extraction. Students learn important concepts in speech processing, such as setting the frame size, use of Hamming window, and determining the LPC coefficients. Using the PZ-Plot block, students also learn the relations of the poles and the formants.

The second part of the exercise focuses on machine learning aspects of speech recognition. The first two formant frequencies extracted for each frame and for different phonemes are accumulated in the formant block. This becomes the data for training in the K-means block. In this part, students learn to choose an appropriate value of k depending on the number of phonemes to be clustered. Through the project students also learn that even for sustained phonations, formant frequencies vary from one frame to another and from one speaker to the next. The exercise also included noisy speech phonemes so that students can understand the effects of noise in speech formant extraction and classification.

VI. ASSESSMENT

In this section, we discuss the assessments of the exercise and the software and their impact on student learning [27]. With the help of the exercise and online software we have developed, students are able to relate class concepts to practical engineering applications. We assigned pre- and post-quizzes to a class of 48 senior level students. The exercise is available at [29]. The pre-exercise quiz was assigned before the start of the project and the post-quiz after the exercise. The duration between the pre-quiz and post-quiz was 3 weeks. The responses were collected anonymously. In nearly all questions we observed improvements which we continue to analyze. A total of 21 questions were posed. First 12 questions were based on LPC, Windowing, and FFT use in speech processing and formant extraction. Next 9 questions were based on the K-means machine learning algorithm, training, clustering, effects of noise, etc. Figure 9 show graphs for the pre-quiz and post-quiz from the JDSP-based exercise. Comparing the results of the pre-quiz and post-quiz, there was an improvement of 31.46% in speech problems and 17.81% in machine learning problems. The full quiz can be found at [30].

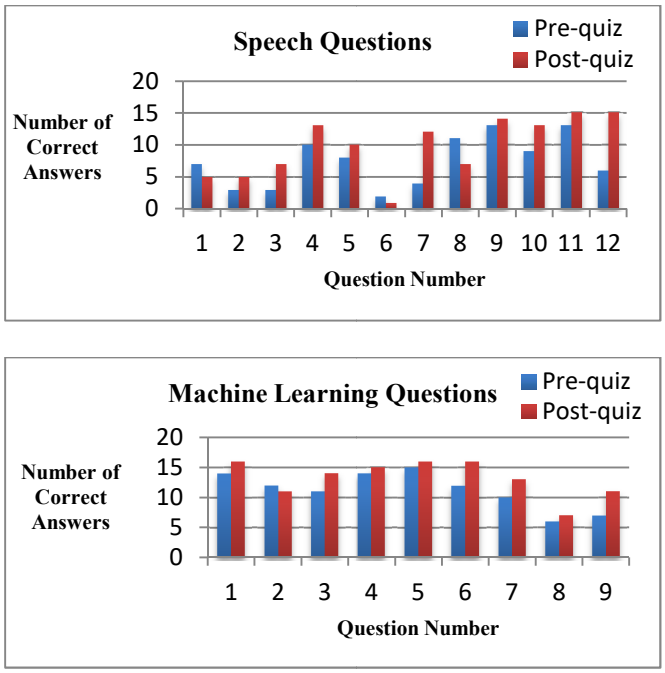

Figure 9: Pre- and Post-quiz results for speech results.

VII. CONCLUSIONS

In this paper, we presented the development of online machine learning software along with an exercise in speech processing for use in the DSP class. A series of HTML5 interactive blocks were developed and used to support a computer exercise that provided learning experiences in linear predictive coding of speech and in using machine learning for phoneme recognition. The software and exercise were assessed using quizzes and interviews. We interviewed 10 senior level students after the post-quiz and obtained comments on ease of use of the new functions and reflect on what they learned. Our assessment results were encouraging and demonstrated learning of several new concepts in the undergraduate class and association of these concepts with UG DSP theory. The exercise and the associated interactive graphical user interface energized the students many of whom discussed their results in class and posed several "what if" questions. We plan to enhance the software and repeat the exercise in class for multiple phonemes and with experiments containing co-channel noise. The full results of the exercise, software and their assessment will be presented at the conference.

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] 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.

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

[3] 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. Lecture Notes in Computer Science, vol 3583. Springer, Berlin, ICWL 2005.

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

[5] 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, pp. 265-271, 2010.

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

[7] 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.

[8] 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.

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

[10] 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, pp. 735-749, Nov. 2005

[11] A. Dixit, S. Katoch, P. Spanias, M. Banavar, H. Song, A. Spanias, "Development of Signal Processing Online Labs using HTML5 and Mobile platforms," *IEEE FIE 2017*, Indianapolis, Oct., 2017.

[12] Atti V. and Spanias A., "On-line Simulation Modules for Teaching Speech and Audio Compression ," *33rd ASEE/IEEE FIE-03*, Boulder, November 2003.

[13] 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.

[14] 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.

[15] 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.

[16] A. Spanias, K. Huang+, R. Ferzli, H. Kwon+, V. Atti+, V. Berisha+, L. Iasemides, H. Krishnamoorthi+, P. Spanias, S. Misra+, M. Banavar+, K. Tsakalis, S. Haag, "Interfacing Java-DSP with a TI DSK board," ASEE *COE Journal*, Vol. XVII, No. 3, Invited, July-Sep. 2007.

[17] 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.

[18] S. Ranganath, JJ Thiagarajan, KN Ramamurthy, S. Hu, M. Banavar, A. Spanias, "Undergraduate Signal Processing Laboratories for the Android Operating System," AC 2012-4755, *ASEE Conf.* , June 2012.

[19] J. Makhoul, "Linear Prediction: A Tutorial Review," Proc. IEEE, pp. 561-580, April 1975.

[20] A. Spanias, "Speech Coding: A Tutorial Review," Proc. IEEE, Vol. 82, No. 10, pp. 1441-1582, October 1994.

[21] A. Spanias, T. Painter, V. Atti, Audio Signal Processing and Coding, Wiley, March 2007.

[22] L.Rabiner and R.Schafer,Theory and Applications of Digital Speech Processing, Pearson, 2011.

[23] U. Shanthamallu, A. Spanias, C. Tepedelenlioglu, M. Stanley, "A Brief Survey of Machine Learning Methods and their Sensor and IoT Applications," *Proc. 8th Int. Conf. on Information, Intelligence, Systems and Applications (IEEE IISA 2017)*, Larnaca, August 2017.

[24] V. Berisha, A. Wisler, A. Hero, A. Spanias, "Empirically Estimable Classification Bounds Based on a Nonparametric Divergence Measure," *IEEE Trans. on Signal Processing,* vol. 64, pp.580-591, Feb. 2016.

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

[26] Li Deng, Douglas O'Shaughnessy. *Speech processing: a dynamic and optimization-oriented approach*. Marcel Dekker. pp. 41–48, 2003.

[27] A. Spanias and J. Blain Christen , "A STEM REU Site On The Integrated Design of Sensor Devices and Signal Processing Algorithms'," *Proc. IEEE ICASSP 2018,* Calgary, April 2018.

[28] A study of the formants of the pure vowels of British English. University of London, 1962.

[29] http://jdsp.engineering.asu.edu/MLExAp18/ MLEx.pdf

[30] http://jdsp.engineering.asu.edu/MLExAp18/ MLQuiz.pdf

[31] D. Petkovic *et al*., "Work in progress: A machine learning approach for assessment and prediction of teamwork effectiveness in software engineering education," *Frontiers in Education Conference Proceedings*, pp. 1-3. Seattle, WA, 2012,

[32] M. Pantic and R. Zwitserloot, "Active Learning of Introductory Machine Learning," *Proceedings. Frontiers in Education. 36th Annual Conference*, pp. 1-6, San Diego, CA, 2006.

[33] Z. Markov, I. Russell, T. Neller and S. Coleman, "Enhancing undergraduate AI courses through machine learning projects," *Proceedings Frontiers in Education 35th Annual Conference*, pp. T3E-21, Indianopolis, IN, 2005.

[34] Y. Shibberu, " *Introduction to Deep Learning: A First Course in Machine Learning",* paper presented at *2017 ASEE Annual Conference & Exposition, Columbus*, Ohio, June, 2017.