

# Curriculum Development of an Audio Processing Laboratory Course

Buket D. Barkana  
School of Engineering,  
University of Bridgeport,  
221 University Ave. Bridgeport, CT 06604 USA

**Abstract**—This paper describes the development of an audio processing laboratory curriculum at the graduate level. A real-time speech and audio signal-processing laboratory is set up to enhance speech and multi-media signal processing courses to conduct design projects. The recent fixed-point TMS320C5510 DSP Starter Kit (DSK) from Texas Instruments (TI) is used; a set of courseware is developed. In addition, this paper discusses the instructor's and students' assessments and recommendations in this real-time signal-processing laboratory course.

**Key Words:** audio processing laboratory, fixed-point DSP, TMS320C5510, real-time signal processing.

## Introduction

Audio signal processing is a part of the digital signal processing (DSP) field in science and engineering that has developed rapidly over the past several years. Audio processing has become a part of everyday life in the digital world. Some form of audio processing is used every time you use the phone, cell phone, MP3 player, play a DVD and audio/video media file, or watch a movie. Expertise in audio signal processing is becoming increasingly important for working engineers from diverse disciplines. Audio signals are stored, processed, and transmitted using digital techniques. Creating these technologies requires engineers that understand real-time application issues in these areas.

Many DSP laboratory courses are developed for undergraduate and graduate level engineering education. These courses mostly focus on the general signal processing techniques such as quantization, filter design, Fast Fourier Transformation (FFT), and spectral analysis [1, 2, 3].

Because of the increasing popularity of Web-based education and the advancements in streaming media applications, several web-based DSP Laboratory courses have been designed for distance education [4, 5, 6, 7, 8]. An internet-based signal processing laboratory that provides hands-on learning experiences in distributed learning environments has been developed in [6]. This laboratory is based on an object-oriented Java-tool called Java Digital Signal Processing (JDSP). It facilitates interactive on-line simulations of modern statistical signal and spectral analysis algorithms, filter design tools, QMF banks, and state-of-the-art vocoders. A LabVIEW based Remote DSP Laboratory is developed to carry out laboratory exercises such as Finite Impulse Response, Infinite Impulse Response digital filters and FFT as well as run any executable file developed by the user [8]. Another approach for web-based remote digital signal processing laboratory using the integrated learning methodology (ILM) suggested an approach that is based on integrating on-site and distance learning environments, using web-based communication techniques and remote monitoring of laboratory equipment [7]. Like the others, it carries out

laboratory exercises such as quantization and FIR filters. Although most of the universities and colleges offer DSP laboratory courses that do not provide the needed hands-on experience with real-time audio processing technologies [9, 10, 11, 12, 13].

During the past four years at our school, signal processing courses such as the DSP laboratory, speech signal processing, speech coding, and multimedia signal processing have attracted a steady group of graduate students, mostly without practical knowledge in these fields.

Knowledge in the audio processing field is essential to the understanding of the function of current and future digital audio processing systems and to forming a strong foundation for the learning of newly developed digital devices/systems with applications to audio signals. The main contribution of the proposed course to the education of engineers will be (i) the understanding, through practical applications, of the DSP theory, (ii) student involvement with state-of-art technology, (iii) the development of real-time DSP hardware and software experiences in the music and speech processing fields, (iv) student familiarization of industry development processes, (v) the processing and editing of engineering documentation, and (vi) the development of design and research skills in the audio processing field.

Computer musicians use digital filters in the music they create. Without digital reverberation, it is difficult to get rich, full-bodied sound from the computer. However, reverberation is only the start of the capabilities of digital filters. A digital filter can arbitrarily shape the spectrum of a sound signal. Yet very few musicians are prepared to design the filter they need, even when they know exactly what they want in the way of a spectral modification [14]. At this point, it is very important to teach filter design in audio processing. By doing this, the proposed course can be adapted to the needs of different schools, such as Speech and Audio Technology centers, Departments of Music, Linguistics Departments, communications companies, computer science and engineering schools, and forensic audio engineering.

The following sections take a closer look at some of the tools, hardware, and experimental setups used in conducting this lab course. Section III outlines the course content that includes the possible audio and voice/speech projects using real-time DSK. Section IV discusses the evaluation criteria of this course using the assessments and feedback of the students and the instructor.

## **Structure of the audio laboratory**

### *Lab Facilities:*

The laboratory currently consists of eleven stations: 10x Intel Celeron (1.70 GHZ, 512 MB) and 1x P4 (3 GHZ, 512 MB). Each station is equipped with multimedia hardware capabilities (including a real-time DSP board), general software development tools, and real-time development tools. In addition, a headphone, microphone, function generator, and oscilloscope are available for every workstation.

### *Software:*

Each computer is equipped with general software tools for developing labs and projects including MATLAB 7.0, Code Composer Studio (CCS), and Goldwave Digital Audio Editor. The full version of MATLAB 7.0 includes the Signal Processing toolbox and Simulink [6]. MATLAB provides the programming environment for quick development of a theoretical concept. Graduate

students are introduced to MATLAB in their first semester and are familiar with it by the time they take the audio processing laboratory course. Goldwave Digital Audio Editor is a tool to play, edit, mix, and analyze audio signals for pod casting or telephone systems and more. It is present on all workstations in the laboratory.

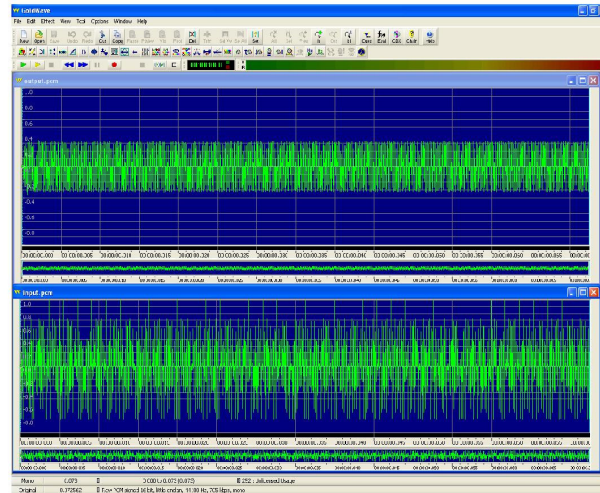


Fig 1. Screenshot of the Goldwave Audio Editor

*Real-time development tool DSK5510: Why is the fixed-point DSP chosen for this course?*

There are two reasons to choose fixed-point DSK for the proposed course. The first reason is that the fixed-point DSP is chosen most often for superior audio because of the following advantages:

1. Less dynamic range
2. Double-precision capable 48-bit processing
3. Programming flexibility that can guarantee proper behavior under the adverse conditions presented by audio signals
4. Lower-power consumption (floating point hardware is more complicated than fixed point; more transistors require more watts)

Looking under the hood of an IEEE 32-bit floating-point processor and a 24-bit fixed-point processor reveals that each DSP design offers the same 24-bit processing precision — therefore, *precision* is not the issue. The defining difference is that the fixed-point implementation offers double precision, while the floating-point device features increased dynamic range. In floating-point processors scaling the data increases dynamic range, but scaling does not improve precision, and in fact degrades performance for audio applications. And it turns out that the strength of the fixed-point approach is the weakness of the floating-point, giving fixed-point a double advantage. The benefit is most obvious in low frequency audio processing. This is important since most of the energy in audio lies in the low-frequency bands (*music and speech have an approximate 1/f spectrum characteristic, i.e., each doubling of frequency results in a halving of amplitude*). The simple truth is that the floating-point technique struggles with large amplitude and low-frequency computations. In fact, building a high-Q, low frequency digital filter is difficult no matter what method you use, but all things considered fixed-point double-precision is superior to floating-point single-precision [16]. More information on the C55x processor can be found at reference 17. The TMS320C5510<sup>TM</sup> DSP which is given in Fig.2. is supported by the CCS application which provides an integrated environment with an editor,

debugger, project manager, profiler, C/C++ compiler, assembly optimizer and linker (code generation tools), simulator, real-time operating system (DSP/BIOS™), real-time data exchange between the host and target (RTDX™), real-time analysis, and visualization. A screenshot of the CCS is shown in Fig.3.

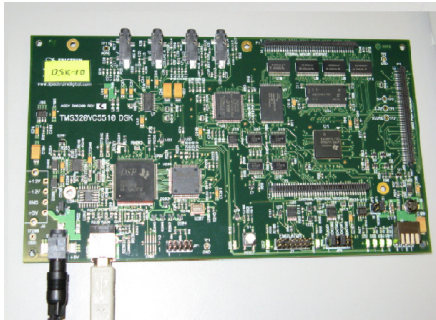


Fig.2. The fixed-point TMS320C5510 DSP Starter Kit (DSK) from Texas Instruments (TI).

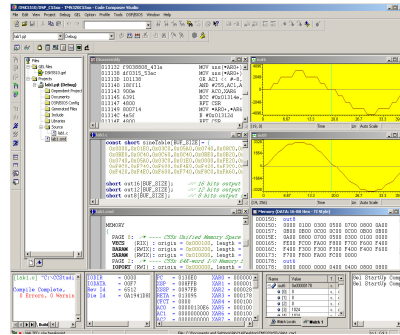


Fig.3. The results from the experiment 1. Part A, captured from the CCS V3.3 screen.

The second reason for choosing the fixed-point DSP is the recommended textbook for the course. The textbook, *Real-Time Digital Signal Processing Implementations and Applications* by Kuo, includes many hands-on experiments using C55x intrinsics with CC Studio version 2.0 based on DSP [18]. Students do not have time to design real-time DSP hardware and software from scratch in the class hours since the proposed course is a 3-cr course. The laboratory exercises are directly drawn from those pertinent to the student's material (the proposed textbook), and students are involved with the development of the target-oriented code.

## Course design and content

### Objective

Our goal is to provide a laboratory course for audio signal processing courses that meets both the hands-on experience and educational needs of graduate engineering students. The objectives of the planned laboratory are:

- Using a combination of the various media of audio analysis, introduce the fundamentals in DSP.
- Introduce students to real-time implementations of signal processing.
- Enhance students' education through hands-on experiments drawn from diverse disciplines.
- Develop an audio signal processing laboratory that can be adopted by other institutions.

### Organization

A 3 credit, one semester lab course has students organized into groups of two students. At the beginning of each class, each group gives a written report to the instructor, and this report includes the results of the previous week's laboratory exercise. Each report is corrected, graded, and returned to the students the next week. Students have an early feedback of their performance, and this is very helpful in subsequent laboratory work.

Each group prepares a hands-on project as a final project. Students start preparing their final project at the middle of the semester. Each student is expected to spend approximately three hours per week in the lab, in addition to the scheduled lab times. Since students have access to the laboratory outside of scheduled lab times, there is plenty of time to finish their final project on time.

*Prerequisite:*

DSP is the only prerequisite course for the proposed course.

*Course Syllabi*

Students are introduced to the DSP hardware and software development tools during the first two weeks of the semester. In addition, lectures will cover background material pertinent to lab in the following areas: the acoustics and acoustic analysis of audio (music and speech), the physiology of speech production, and the perception of audio.

## **Lab experiments**

Most of the laboratory experiments are adopted from the textbook by Kuo et al. [18]. Since students do not have a strong knowledge on C/C++ programming, they are asked to be involved with the development of target-oriented codes from the textbook.

**Lab1:** *Introduction to Real Time Digital Signal Processing.*

**Lab2:** *Introduction to TMS320C55x Digital Signal Processor (Hardware Introduction).*

**Lab3:** *Sampling and quantization of audio signals*

This experiment is used to demonstrate quantization effects, overflow and saturation arithmetic, and to determine the proper fixed-point representations using speech or music signals. The analog-to-digital converter (ADC) converts an analog signal  $x(t)$  to a digital signal  $x(n)$ . The input signal is first sampled to obtain the discrete-time signal  $x(nT)$  with infinite precision. Each  $x(nT)$  value is then encoded using  $B$ -bit wordlength to obtain the digital signal  $x(n)$ . Since the quantizer employs  $B$ -bits [ $B = (M + 1)$ ], the number of the quantization levels available is  $2^B$ . The quantization error  $e(n)$  is the difference between the discrete-time signal  $x(nT)$  and the quantized digital signal  $x(n)$ , and is expressed as

$$e(n) = x(n) - x(nT). \quad (1)$$

Assuming the signals and filter coefficients have been normalized in the range of  $[-1, 1]$  for fixed-point arithmetic, the sum of two  $B$ -bit numbers may fall outside the range of  $[-1, 1]$ . Overflow is a condition that the result of an arithmetic operation exceeds the capacity of the register used to hold that result. Most DSP processors have mechanisms that protect against overflow. Saturation arithmetic prevents overflow by keeping the result at a maximum value. Since saturation arithmetic is a nonlinear operation, it will add undesired nonlinear components. Saturation logic is illustrated in Fig.4 and can be expressed as

$$y = \begin{cases} 1 - 2^{-M}, & x \geq 1 - 2^{-M} \\ x, & -1 \leq x < 1 \\ -1, & x < -1 \end{cases} \quad (2)$$

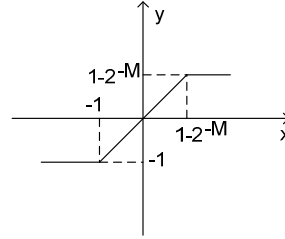


Fig.4. Characteristics of saturation arithmetic

Students find the mean and variance of quantization noise and evaluate the quantization effects of speech and music signals using the DSK for real-time experimentation with different word lengths. The Goldwave audio editor is used to play the quantized signals.

**Lab4: Design and Implementation of Finite Impulse Response (FIR)**

This experiment aims to overcome filtering and filter design issues using audio signals. Students are asked to remove unwanted frequency components on speech and music signals with different order FIR filters. In addition, the audio signal sampling rate is converted from 32 to 24 kHz. Students first interpolate the signal sampled at 32 kHz to 96 kHz, and then decimate it to 24 kHz.

Filtering is one of the most useful signal processing operations. An FIR filter operates on discrete-time signals and can be implemented with a DSP chip such as the TMS320C5510. This involves use of an ADC to capture an external input signal, processing the input samples, and sending the resulting output through a DAC. The filter design consists of the approximation of a transfer function with a resulting set of coefficients. The difference equation of an FIR filter is given as

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n - k), \quad (3)$$

where  $y(n)$  is the output and  $N$  is the number of the coefficients. Low-pass and band-pass FIR filters are designed using DSK5510. Coefficients are calculated with MATLAB. Lowpass FIR filtering is the key process for sampling rate conversion which is necessary for interconnecting DSP systems operating at different sampling rates. This digital sampling-rate conversion can be reached by first performing interpolation of a factor  $U$ , and then decimating the signal by a factor  $D$ . See Fig.5.

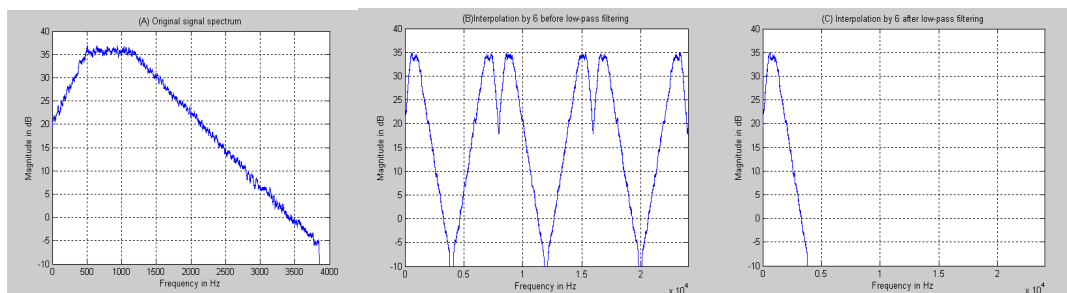


Fig.5. Interpolation by an integer operation: (a) original signal spectrum; (b) interpolation by 6 before lowpass filtering; (c) interpolation after lowpass filtering. (Kuo [18])

**Lab5: Design and Implementation of Infinite Impulse Response (IIR) Filters**

Students focus on the design, realization, implementation, and application of digital IIR filters. The transfer function of the IIR filter is defined in Eq.(4).

$$H(z) = \frac{\sum_{l=0}^{L-1} b_l z^{-l}}{1 + \sum_{m=1}^M a_m z^{-m}}, \quad (4)$$

where L is the filter order. To perform IIR filter design, find the coefficients  $b_l$  and  $a_m$  in order to satisfy the given specifications. The IIR filter can be realized by the difference (input/output) equation

$$y(n) = \sum_{l=0}^{L-1} b_l x(n-l) - \sum_{m=1}^M a_m y(n-m). \quad (5)$$

There are several types of structures or realizations including direct-form, cascade-form, lattice, and lattice-ladder structures in the case of IIR and FIR systems. Direct Form I and Direct Form II realization of IIR filter are implemented with this experiment. Direct-form realizations are obtained directly from the system function  $H(z)$  without any arrangement of it. More theoretical details regarding the direct form I and II structures can be found in many signal processing textbooks.

**Lab6: Adaptive Filter Design**

Modern mobile communications technology requires knowledge of adaptive filter design. Audio signals and background noise are often random in nature. In this experiment, the properties of the random processes are reviewed. Students implement the filter design in Figure 6.

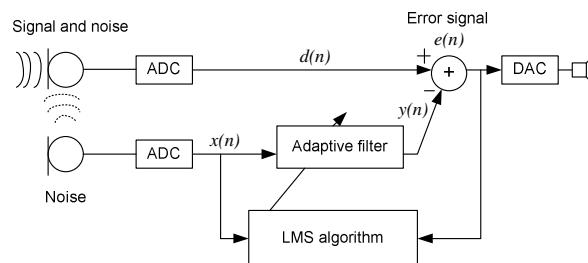


Fig.6. Noise canceler using a one-tap adaptive filter [19]

There are two ADC channels. The first microphone captures the noisy speech (speech + noise),  $d(n)$  while the second microphone resides where it picks up only the correlated noise and feeds the noise reference  $x(n)$  to the adaptive filter. The adaptive filter uses LMS (least mean squares) algorithm to adjust its coefficients to produce the best estimate of the noise  $y(n)$ . The output of the error signal is expected to be a best estimate of the clean speech. In FIR filter design, the difference equation (the relationship between the input and output signal) is computed as

$$y(n) = \sum_{l=0}^{L-1} w_l(n)x(n-l), \quad (6)$$

where the filter coefficients  $w_l(n)$  are time varying and updated by the adaptive algorithms. The input vector at time  $n$  is defined as

$$x(n) \equiv [x(n)x(n-1)\dots x(n-L+1)]^T, \quad (7)$$

and the weight vector at time  $n$  is

$$w(n) \equiv [w_0(n)w_1(n)\dots w_{L-1}(n)]^T. \quad (8)$$

The difference equation (6) can be expressed in vector form as

$$y(n) = w^T(n)x(n) = x(n)w(n). \quad (9)$$

The filter output  $y(n)$  is compared with the desired response  $d(n)$  to obtain the error signal

$$e(n) = d(n) - y(n) = d(n) - w^T(n)x(n). \quad (10)$$

After substituting this gradient estimate into the steepest-descent algorithm, a well-known LMS algorithm (the stochastic gradient algorithm) can be found as

$$w(n+1) = w(n) - \mu x(n)e(n). \quad (11)$$

As the step size increases, it goes faster and the error increases. The step size controls stability and the rate of convergence. The upper bound on step size is inversely proportional to filter length  $L$ . The convergence of the LMS algorithm must satisfy

$$0 < \mu < \frac{2}{LE[x^2(n)]}. \quad (12)$$

Students are asked to adjust the step size of the LMS algorithm and the length of the filter to minimize the error using the TMS320C5510. The Goldwave audio player and MATLAB are used to watch the convergence/divergence of the output signal.

### **Lab7: Echo Cancellation**

This experiment focuses on adaptive voice echo cancellers for long-distance networks. Cancelling the voice echo in long distance signal transmission and acoustic echo in hands-free speakerphones is very important. There are many possible signal paths in a telephone network. A two-wire circuit is bidirectional and carries signals both ways. A four-wire circuit uses two separate unidirectional paths for signal transmission in both directions. See Fig.7. It is very difficult to provide a perfect impedance match in every case since the length of the subscriber lines varies. The resulting imbalance causes a large portion of the incoming signal from the distance talker to appear in the transmit path, and it is returned to the talker as an echo.



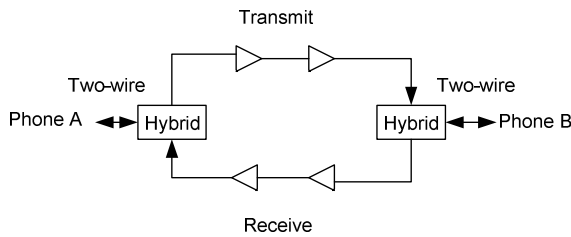


Fig.7. General 2/4-wire interconnection scheme

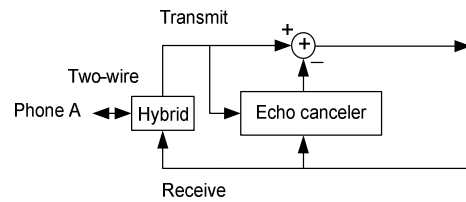


Fig.8. Echo cancellation scheme

The implementation of a fixed-point acoustic echo canceller uses the leaky NLMS (nonlinear LMS) algorithm as an echo canceller in Fig.8. Transmission delay between the echo canceller and the hybrid is estimated using the cross-correlation function in MATLAB.

**Lab8: Audio Enhancement Techniques**

Excessive noise degrades the performance of audio signal processing techniques. Speech enhancement (noise reduction) algorithms become increasingly important to improve voice quality in noisy environments for hands-free applications.

Students are given speech signals that are recorded in a noisy environment and are asked to use voice activity detection algorithm and various filtering ideas to reduce the background noise using DSK5510. This experiment is another example of the combination of the usage of MATLAB, the Goldwave Audio Editor, and DSK5510.

**Lab9: Modified Discrete Cosine Transform**

Speech coding is a process that leads to the representation of analog waveforms by sequences of binary digits. Although high-bandwidth channels and networks are becoming more viable, speech coding for bit rate reduction has retained its importance. Here, a basic audio coding algorithm in the frequency domain, the Modified Discrete Cosine transform (MDCT), is introduced. Students gain knowledge and experience (i) on the use of filter banks, (ii) on converting time-domain signals to frequency domain coefficients, (iii) on developing and using psychoacoustic models, (iv) on quantization/dequantization, and (v) on aliasing cancellation in the time domain and frequency domain.

**Lab10-11: Final Project**

**Samples of the Final Projects:**

Project 1:

One of the projects prepared during the Spring 2008 semester proposed a voiced/unvoiced classification algorithm using the short-time zero-crossings rate and energy of the speech signal. In the frame-by-frame processing stage, the speech signal is segmented into a non-overlapping frame of samples.

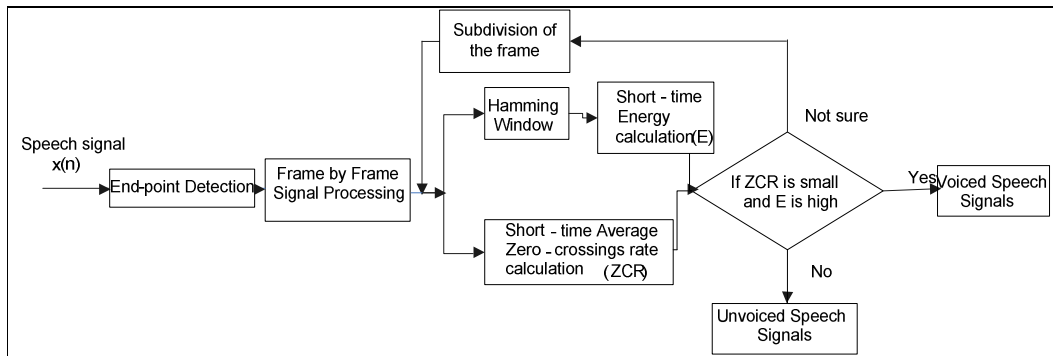


Fig.9. Block diagram of the voiced/unvoiced classification algorithm

TABLE I. VOICED/UNVOICED DECISIONS FOR THE WORD “FOUR” USING THE MODEL

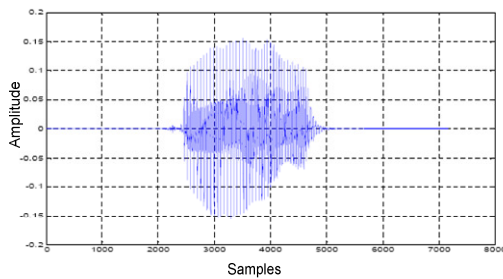


Fig.10. Original speech signal for the word “four.”

| Frames                     | ZCR | Energy   | Decision |
|----------------------------|-----|----------|----------|
| Frame 1 (400 samples)      | 152 | 0.0018   | unvoiced |
| Frame 2(1) (200 samples)   | 52  | 0.0543   | unvoiced |
| Frame 2(2) (200 samples)   | 19  | 21.1189  | voiced   |
| Frame 3 (400 samples)      | 41  | 186.6628 | voiced   |
| Frame 4 (400 samples)      | 41  | 230.5772 | voiced   |
| Frame 5 (400 samples)      | 43  | 252.98   | voiced   |
| Frame 6 (400 samples)      | 56  | 193.70   | voiced   |
| Frame 7(1) (200 samples)   | 31  | 27.2842  | voiced   |
| Frame 7(2) (200 samples)   | 30  | 25.960   | voiced   |
| Frame 8(1_1) (100 samples) | 24  | 3.4214   | voiced   |
| Frame 8(1_2) (100 samples) | 11  | 0.4765   | unvoiced |
| Frame 8(2) (200 samples)   | 19  | 0.166    | unvoiced |
| Frame 9 (400 samples)      | 89  | 0.0054   | unvoiced |

Block diagram of the voiced/unvoiced classification is given in Fig.9. Table 1 includes the voiced/unvoiced decisions for the word “four” which is given in Fig.10. It has 3600 samples at an 8000Hz sampling rate. At the start, the frame size is set as 400 samples. At the end of the algorithm if the decision is not clear, the energy and zero-crossing rate is recalculated by dividing the related frame size into two frames. This phenomenon can be seen for Frame 2, 7, and 8 in Table I. This project won the second place of the best student paper award at the ASEE Regional Conference, West Point, USA, 2008 [20].

Project 2:

Another project prepared during the Fall 2008 semester proposed a Dual Architecture Implementation of Acoustic Echo Cancellation using TMS320C5510 DSP. An external-switched algorithm of a dual architecture implementation is developed. Using the orthogonality property of adaptive algorithms to detect convergence, two complete adaptive filters can be run in parallel to take advantage of each filters particular configuration. By configuring one filter for fast adaptation and the second for minimizing the steady-state error, a system is designed with the advantages of both without suffering from increased computational cost. The block diagram of the system is given in Figure 11.

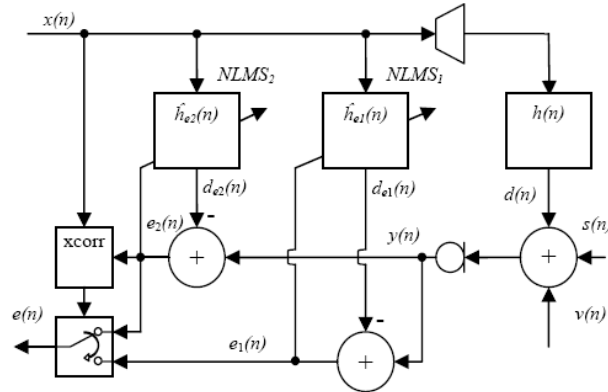


Fig.11. Dual Architecture AEC System

This project received an honorable mention award in the student paper competition at the NE ASEE Regional Conference, Bridgeport, USA, 2009 [21].

## Evaluation and Assessment

Informal discussions of perceptions of laboratory work were held with individuals and groups of students for two years. Although these were not audio taped, notes were written up immediately afterwards. Based on the teaching experience from the 2007-2009 academic years and the results obtained with Audio Lab, students considered their experience with the lab to be very good. A high percentage of the students consider the Audio Lab useful in helping them to understand the concepts of DSP in general and of music and speech signals in particular. At the same time, they found it very challenging with their initial weak background in software. Most of the students showed great interest in the final project. Some of the projects were presented at national conferences as student papers or posters.

Most of the engineering graduate students are from overseas, coming to America and to the University of Bridgeport. Although they have a good mathematical background, they usually do not have any experience with software tools such as MATLAB. This issue causes a big challenge to students and the instructor of this course. To overcome this challenge, mini MATLAB courses are offered at the beginning of the semester. The lab assistant teaches the Digital Signal Processing Toolbox in MATLAB. The students who have a software background perform much better than the others who do not have any software background.

During the past semester, students were organized into groups of three. The instructor noticed that one student was always left out. For this reason, the capacity of the course is reduced by half. The students may elect to work alone or with a lab partner.

The students have three hours per week to complete the listed experiments. We noticed that most of the students needed more time to finish each lab. With the lab assistance's guidance, they were allowed to work in the laboratory using necessary hardware and software tools outside of the normal lab/lecture hours.

## Conclusion

An audio processing laboratory has been presented to enhance the material and concepts covered in DSP, speech, and multimedia signal processing courses. This course is integrated in the EE

curriculum at the School of Engineering, University of Bridgeport, CT, USA. Hands-on real-time experiments are based on the TMS320C5510 DSK and MATLAB. Students learn industry-style design with the TMS320C5510 DSK. This experience helps the students when they work as a professional engineer in audio processing. Students have gained experience on these topics: research experience, industrially relevant experience, MATLAB programming, real-time DSP, designing and implementing projects, writing and presentation skills, all of which they will need later when working in industry.

Our experience with the laboratory has been very positive. The students like the experiments, and they feel that they are useful in solidifying concepts covered in lecture. They have a better sense of the power of the algorithms used in the signal processing of music and speech. To overcome their weak software background, students receive a set of pre-laboratory assignments in many cases, and these provide theoretical and practical background in conducting experiments.

## Acknowledgment

The author is indebted to Dr. Lawrence V. Hmurcik who has provided valuable suggestions and discussions and wishes to thank the students who took the course ELEG-459 Audio Signal Processing Laboratory at the University of Bridgeport.

## References

- [1] I. M. Abdel-Qader, B. J. Bazuin, Real-Time Digital Signal Processing in the Undergraduate Curriculum, *IEEE Transactions on Education*, vol.46, no.1, February, 2003
- [2] D. Jacoby and R. Saint-Nom, Nice experiences teaching SP in Argentina, *Acoustics, Speech, and Signal Processing, IEEE International Conference on acoustics, Speech, and Signal Processing*, 2689-2692 vol.5, 2001
- [3] J. P. Frantz, H. Choi, R. Baraniuk, DSP Education at Rice University,
- [4] S. Gallardo, F. Barrero, S. L. Toral, M. J. Durán, *eDSPlab: A remote-accessed instrumentation laboratory for digital signal processors training based on the Internet*, *IEEE Industrial Electronics, IECON 2006 - 32nd Annual Conference on*, 6-10 Nov. 2006
- [5] A. Spanias, and V. Atti, Interactive Online Undergraduate Laboratories Using J-DSP, *IEEE Transactions on Education*, vol.48, no.4, November 2005
- [6] A. Spanias, S. Urban, A. Constantinou, M. Tampi, A. Clausen, X. Zhang, J. Foutz and G. Styliano, Development and Evaluation of a Web-Based Signal and Speech Processing Laboratory for Distance Learning, *Computers in Education Journal*, April-June 2000
- [7] Z. Dvir, Web-Based Remote Digital Signal Processing (DSP) Laboratory Using The Integrated Learning Methodology (ILM), 2006 International Conference on Information Technology: Research and Education , 2006
- [8] A. Kalantzopoulos, D. Karageorgopoulos, E. Zigouris, A LabVIEW based Remote DSP Laboratory, *International Journal of Online Engineering (iJOE)*, vol.4, 2008
- [9] L. Moreno, J.F. Sigut, J.J. Merino, J.I. Estevez, J.L. Sanchez, A. Brito, "Digital Signal Processors for a Signal Processing Laboratory", *IEEE Trans. Educ.*, vol.42, no.3, pp.192-199, Aug. 1999
- [10] S.M. Kuo, G.D. Miller, "An Innovative Course Emphasizing Real-Time Digital Signal Processing Applications", *IEEE Trans. Educ.*, vol.39, no.2, pp.109-113, May 1996
- [11] M.J. Caputi, "Developing Real-Time Digital Audio Effects for Electric Guitar in an Introductory Digital Signal Processing Class", *IEEE Trans. Educ.*, vol.41, no.4, Nov. 1998
- [12] W.S. Gan "Teaching and learning the how's and why's of real-time digital signal processing", *IEEE Trans. Educ.* vol.45, no.5, pp.336-443, Nov.2002
- [13] A.J.S. Ferreira and F.J.O. Restivo, "Grasping the potential of digital signal processing through real-time DSP laboratory experiments", in *Proc. IEEE Signal Processing Society 2<sup>nd</sup> Signal Processing Education Workshop*, Pine Mountain, GA, pp.286-291, Oct.2002
- [14] Smith, J.O., "Introduction to Digital Filters", in *Introduction to Digital Filters with Audio Applications*, W3K Publishing, <http://books.w3k.org/>, 2007, ISBN 978-0-9745607-1-7.
- [15] *MATLAB Version 7.0 User Manual*, Math Works, Inc., Natick, MA

- [16] Greg Duckett, Terry Pennington, Rane Corporation, RaneNote 153, Revised 2005
- [17] *TMS32055x Code Composer Studio: Tutorial*, Texas Instruments, <http://www.ti.com/>
- [18] S.M. Kuo, B.H. Lee, and W. Tian, *Real-Time Digital Signal Processing Implementations and Applications*, pp.49-50 England: Wiley, 2007
- [19] Tan L., *Digital Signal Processing Fundamentals and Applications*: Elsevier, 2008
- [20] Bachu R., Adapa B.K., Kopparthi S., Barkana B.D., "Separation of Voiced and Unvoiced Speech Signals using Energy and Zero Crossing Rate", ASEE Regional Conference, West Point, March 2008. (Best Student Paper Award, Second Place)
- [21] Stark B., "Dual Architecture Implementation of Acoustic Echo Cancellation", Advisor: Barkana B.D., Proceedings of the 2009 ASEE NE American Society for Engineering Education Conference, April 3-4, 2009, (Best Student Paper Honorable Mention Award)