Journal of Theoretical and Applied Information Technology
30th April 2013. Vol. 50 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

DIGITAL SIGNAL PROCESSING TEACHING: A PRAAT BASED APPROACH

[1,2]KURBAN UBUL

[1]Assoc. Prof., School of Information Science and Engineering, Xinjiang University, 830046, China

[2]Researcher, Xinjiang laboratory of multi-language information technology, Urumqi, 830046, China

E-mail: kurbanu@xju.edu.cn

ABSTRACT

Digital Signal Processing (DSP) is a significant and increasing subject area in many disciplines such as, Electrical/Computer Engineering (ECE), Communication Engineering, Computer Science and other Engineering/Science disciplines. With the growing of DSP job market, the DSP course has become very popular among ECE students, and it is still considered to be a difficult and complex one to study. Praat, a speech analysis software, is proposed to apply in signal processing teaching in this paper. More DSP algorithms, even real-time signal processing algorithms are simulated by Praat and its script language that the object of using the software further enlarged to the signal processing area. The authors have enhanced the learning experience for their students by using the software during the teaching, and reduced the difficulty in understanding of DSP theories and algorithms.

Keywords: *Digital Signal Processing (DSP), teaching, computer based educational tool, Praat, Praat script*

1. INTRODUCTION

Digital signal processing (DSP) is one of the extremely important area in science and engineering that has been developed rapidly over the past 5 decades. Now days, DSP is pervasive because it is widely used many important areas such as, communication, space exploration, consumer electronics, robotics, medicine, instrumentation, military, automotive and seismology etc. DSP has enabled the user to remove noisy signals, speed up the communication rate, and store more data, and provides many advantages over its analog processing. Because of these reasons, teaching and learning DSP is becoming an important component in tertiary education.

There are too many sophisticated expressions, formulas, algorithms, waveforms, frequency spectra in the DSP course. It is so hard to understand these in some times. So far, many computer-based educational tools has utilized in DSP education, some of them are conventional text based programming languages such as, Fortran[1], Pascal[2], C[3], C++[4], Matlab [5,6], Mathmatica [7], Java[8] etc. And some of them are graphical programming environments, that is, Simulink[9-10], LabVIEW [11], SystemView [12], and Macromedia Flash [13]. Others are used platforms which software and hardware combined together called firmware, such as, Code Composer Studio (CCS)[14], Filed-Programmable Gate array (FPGA) [15]. Some researchers developed another meaning of tools for DSP teaching mentioned above. Kehtarnavaz et al. [16] designed a system using LabVIEW and TMS320C6000. Korczynski [17] developed virtual harmonic analyzer which operates on simulated and real data is a base for DSP principles. Toral et al. [18] developed a web-based educational tool for DSP teaching.

Nowadays, Simulink and LabVIEW are two graphical programming environments that most widely used for designing DSP systems. They are different with conventional text based programming languages, e.g. C and MATLAB that the graphical programming involves block-based code development, and allowing a more efficient mechanism to build and analyze DSP systems [19]. The firmware has most efficient features among all the computer based tools. Students use Simulink [9-10] to investigate the characteristics of the algorithm and easily design their algorithm with its vast assortment of graphical, DSP, and simulation functions. CCS[14] developed by TI Corporation is a powerful debugging and profiling tool that allows students to write and profile their codes, analyze the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

real-time performance, and even perform real-time scheduling of tasks in the DSP environment.

However, the development of signal processing algorithms for real-time application is still difficult and often requires specialized and lengthy training in a particular programming language for the targeted DSP. Besides, even if the students acquired the programming skills, the process of experiment or project development becomes more tedious and time consuming as the level of complexity increases. Hence, it is becoming increasingly necessary to develop efficient tools for the testing of new DSP algorithms because of the need to shorten the design cycle for new applications becomes critical. This need also meets for conducting a DSP course, where students are required to develop and test various real-time DSP algorithms quickly on actual physical systems. Currently, Matlab and CCS are useful tools for learning effectively DSP over a 16-week, one-semester course. Students must be familiar with DSP algorithm and have some programming ability. But they are inefficient in classroom teaching.

Praat is a widely used software that doing phonetics and acoustic by computer [20-23]. It is presented as a supplementary tool for DSP teaching by taking a few examples in our previous report [24]. More DSP algorithms, even real-time signal processing algorithms are simulated by Praat and its script language that the object of used by Praat are further enlarged to the signal processing area in this paper. This section presented the various kinds of computer based educational tools applied in DSP area is presented in this section. Section 2 described the main content of DSP course. The application Praat and its scripts are indicated in section 3 and section 4 respectively. Conclusion and future works are indicated in section 5.

2. THE MAIN CONTENT OF DSP COURSE

The object of this course is to indicate students the role of DSP that mathematical system theory can play in the development of computer applications/products such as multi-media systems. It has taken “hands-on” teaching style the course with computers being used as an integral part of the classroom and laboratory environment. Generally, traditional classroom lectures present the underlying theory of signals and systems, but the teaching method of the course has been to supplement every lecture with a computer demonstration or simulation that relates the theory to real-world signals, especially speech signal processing and their applications. Furthermore, weekly laboratory assignments are assigned for students to explore the signals in greater depth. The outline of DSP course given below is close to the final definition of the course, but it has been continued to evolve based on the different instructors teaching programs and teaching styles. The topic list of the course would need to be expanded for a semester-length course:

1. Definition and classification of signals and systems.

2. Introduction to MATLAB programming, review vector/matrix notation.

3. Complex numbers: represent sinusoids with phasors.

4. Sinusoidal signals: amplitude, phase and frequency

5. Synthesizing sounds with general classes of sinusoids.

6. Frequency content: harmonics, amplitude modulation and frequency modulation signals.

7. Sampling of continuous-time signals, aliasing and reconstruction.

8. Linear filtering: the concept of smoothing data.

9. Block level description of systems.

10. Infinite Impulse Response (IIR) filter design.

11. Finite Impulse Response (FIR) filter design.

12. Recursive filtering: difference equations with feedback.

13. Frequency response of IIR and FIR filters.

14. Simulation of dynamic time response; impulse response.

15. Z-transform analysis: rational transfer functions; the inverse Z transform; Chrip-z transform.

16. Synthesizing sounds with narrowband recursive filters.

17. Fourier Spectrum (including Fast Fourier Spectrum) analysis: spectrograms and windowing.

18. Finite word length effect: Number representation, quantization of filter coefficient, analog (A)/ digital (D) and D/A conversion.

19. Multi-rate signal processing: Sampling rate conversion, implementation of multi-rate system, filter design for multi-rate system.

Journal of Theoretical and Applied Information Technology
30th April 2013. Vol. 50 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

20. Random signal processing: Spectral analysis of stationary process, optimum linear filters.

The main goals of setting these contents are to explore the theoretical DSP concepts by implementing them on actual hardware or software for real time execution. During the process, students will increase their abilities of operation with equipment commonly used in industry, such as oscilloscopes, spectrum analyzers, signal/function generators, DSP chips, and signal converters.

3. USING PRAAT IN DSP TEACHING

Before clarifying what can be done for the DSP course indicated in section 2 by Praat, the main function of the software is indicated briefly.

3.1 The Main Function of Praat

The main function of Praat are summarized as the following:

1. Sound recording: It can record mono and stereo sounds with different sampling frequency (8KHz~192MHz) in different file formats, such as wav format, aifc format, nist format etc.

2. Sound creating: It can create sounds from formula. Tone complex, gamma tone, Shepard tone and vowel editor separately. The signal parameters (e.g. Sampling frequency, amplitude, duration, etc) can be set arbitrarily by the user.

3. Analyses: Many speech parameters such as, pitch, formant, intensity and spectral can be analyzed by the software.

4. Annotation: It can annotate the speech signal and save the result to TextGrid file.

5. Manipulation: It can conduct various operations to the speech signal via processing speech parameters such as, pitch, duration, intensity, formant etc.

In addition to these general functions, Pratt also has some specialized functions such as, voice analysis, filtering, synthesis, listening experiments, learning and statistics etc. 字体不同

These functions of Praat can be implemented by selecting related buttons in main menu of the software. They can realize or simulate many algorithms and theories in DSP course, if the speech signal is used as an import. This approach which uses Praat to process speech signals with related algorithms can improve students' understanding about DSP theories and algorithms. Some examples are indicated in the following section.

3.2 Praat Application in DSP Teaching

Some algorithms in DSP course have simulated using Praat in our previous report [24]. They include sampling theory, Fourier transform and filtering. The filtering algorithms used before are presented in Table1.

Table 1: The different filtering methods used in Praat

Filtering	Filter Type
Time domain filtering	Pass Hann band filter, Stop Hann band filter, Formula filter
Frequency domain filtering	One formant filter, Pre-emphasis filter, De-emphsis filter

The application of Praat in DSP is further enlarged by taking other algorithms in simulation here. The basic operations of signals, e. g., time shifting, time scaling, time reversal, override sampling, and to finite the signal by different window functions are presented in this paper. Two of them are taken as example.

3.2.1 Time reversal

This operation is the reversal of the horizontal (or time) axis, or flips the signal about the vertical (or y) axis, as indicated in the following Figure 1.

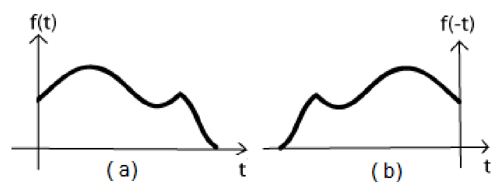
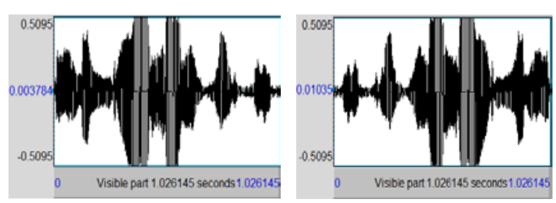


Figure 1: Reverse the time axis

In Figure 1, f(t) is the original signal and the signal performed time reversal is f(-t). The operation is easily implemented by Praat, that is, open a speech signal firstly, then click the "Reverse" button from the "Modify" section, and the reversed signal is appeared in the "Object" window. An example of time reversal is presented as the following Figure 2. The original and reversed signal is indicated in Figure2 (a) and (b) separately.



(a) (b)

Figure 2: An example of time reversal of speech signal

Journal of Theoretical and Applied Information Technology
30th April 2013. Vol. 50 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3.2.2 Window function

It is a mathematical function that is zero-valued outside of some chosen interval. When another function or a signal (data) is multiplied by a window function, the product is also zero-valued outside the interval: all that left is the part where they overlap [25]. It is applied in FIR filter design in DSP subject. There are many kinds of window functions, such as, rectangular window, triangular window, Hanning window, Hamming window, Gaussian window and Kaiser window etc. The application window function in Praat is indicated by taking triangular window and Hamming window examples.

a) Triangular window: Triangular window w(n) is mathematically defined:

$$w(n) = \frac{2}{N+1} \cdot \left(\frac{N+1}{2} - \left|n - \frac{N-1}{2}\right|\right) \qquad (1)$$

where, N is length of window. Triangular window is indicated as the following Figure3.

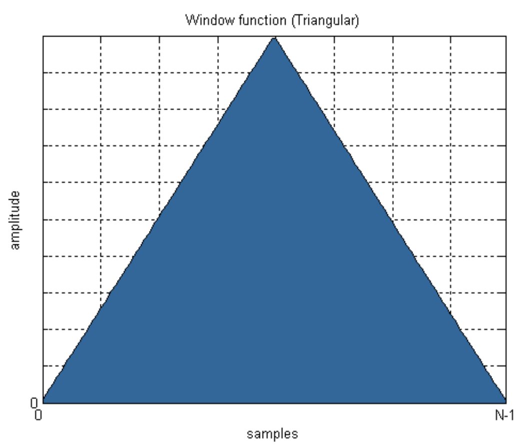
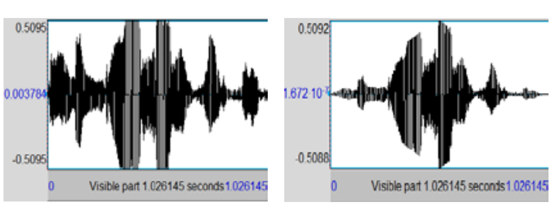


Figure 3: Triangular window

The speech signal is multiplied by triangular window in Praat. Open the speech signal at first, click the “Multiply by window” button from the “Modify” section; then click “Window shape” button and select “Triangular window”; at last the speech signal multiplied by triangular window is appeared in the “Object” window. An example of this is given in Figure4.



(a) (b)

Figure 4: Speech signal multiplied by triangular window

In Figure4, (a) is the original speech signal, (b) is the handled signal multiplied by triangular window.

b)Hamming window: Hamming window w(n) is mathematically defined:

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right) \qquad (2)$$

where, N is length of window. Triangular window is indicated as the following Figure5.

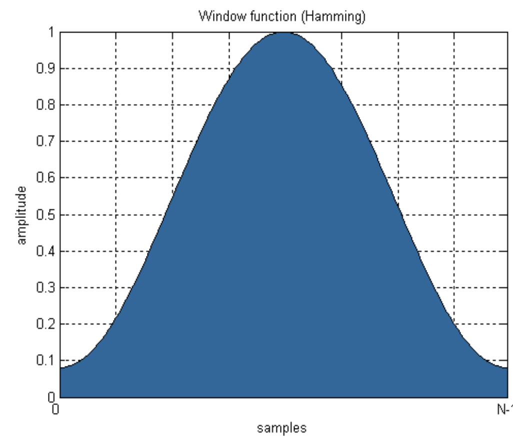
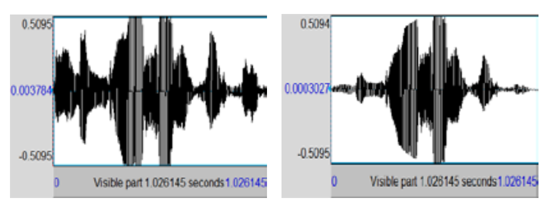


Figure5. Hamming window

Since the process of multiply hamming window to the speech signal is similar to the method of triangular window’s, so it is not explained again. An example for the speech signal multiplied by hamming window is indicated as the following Figure 6. The original signal and the processed signal multiplied by Hamming window are indicated in Figure 6 (a) and Figure 6 (b) respectively.



(a) (b)

Figure 6: Speech signal multiplied by Hamming window

4. USING PRAAT SCRIPT IN DSP

A Praat script is a text that controls the actions of a program. The format of this script text must confirm to certain syntax rules as other programming languages.

4.1 Praat Scripting Language

Praat script has its own format that must confirm to certain syntax rules. Usually, a Praat script

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

includes many elements and some of them are indicated as the Table 2 in appendix.

A script is a text that consists of Praat menu commands and Praat action commands. The text of the script will be interpreted and the corresponding actions will be performed when a Praat script is run. Like other programming languages, Praat has its interpreter which reads and interprets the script text and then initiates these actions. The interpreter runs Praat script in brief. A Praat script is able to solve different problems related with speech signal processing. It can be useful for situations [27]:

1. To automate repetitive operations. The user can fulfill it automatically in relaxed way and save his time with Praat script, if there are some tasks to do the same series of analyses on a large database.

2. To log operations. The script serves as a guide, if the user wants to repeat what he has done.

3. To make drawings in the picture window. Nearly all kinds of drawings about signal processing can be produced with a script.

4. To add a new button in the menu. For instance, the user may script these actions and define a new button in the dynamic menu, if there is a series of actions on a selected speech signal that have to be performed in a prescribed order. So every time he has a speech signal selected and he click that button, the actions in the associated script will be carried out in the prescribed order.

5. To share results with others. The user can communicate to other people what he has done and how they may achieve the same results. There are many examples accompanied by a script that everyone can download.

Theoretically, nearly all the problems about speech signal processing can be solved by Praat scripts. Many researchers have developed various kinds of Praat scripts for speech signal processing. The user can get and modify them easily based on his needs, even develop new scripts. This paper is given 2 examples of Praat script for using in DSP.

Table 2： The elements of Praat script

Elements	Description
Comments	Comments lines start with symbols: #, !, ; (# preferred)
White space	All white spaces (and tabs) at line beginnings are ignored
Continuation lines	Continuation lines start with three dots (...).
Variables	Type: numeric variables, string variables, array variables
Formula	It can modify existing sound, matrix etc.
Loops	Types: while loops, for loops, repeat until loops
Functions	It is include mathematical functions and string functions

4.2 Praat Script in DSP Teaching

Although it has indicated a strong function in speech signal processing with the constant development of new versions of Praat, but there are many problems of signal processing that can't be solved just by the software and its menus. So, the user needs to develop Praat scripts to solve these problems. In general, the Praat scripts are used to process speech signals with many algorithms, such as, sampling, analyzing, labeling, transforming and filtering etc. The user can access the script editing window when he click the "New Praat script" or "Open Praat script" button of the "Control" menu in the Praat main window. The signal addition algorithms and generation of white noise with Praat script are indicated here.

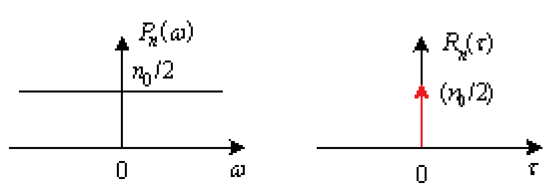
1) Generating white noise: White noise is a random signal with a flat power spectral density in common communication systems. There are several kinds of white noise, and a continuous time, infinite-bandwidth white noise signal is introduced in DSP course. The power spectral density of the white noise satisfies the following:

$$P_n(\omega) = \frac{n_0}{2} \qquad (-\infty < \omega < +\infty) \qquad (3)$$

where, n_0 is a constant. Its autocorrelation function implies:

$$R_n(\tau) = \frac{1}{2\pi}\int_{-\infty}^{+\infty}\left(\frac{n_0}{2}\right)e^{j\omega\tau}d\omega = \frac{n_0}{2}\delta(\tau) \qquad (4)$$

The figure of white noise's power spectral density and autocorrelation function is indicated as the following Figure7.



Journal of Theoretical and Applied Information Technology
30th April 2013. Vol. 50 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Figure 7: The power spectral density and autocorrelation function of white noise.

Praat script of white noise indicated as the following Figure 8. The scripts include elements comments (which begin with #), white spaces, variables (e.g. i), formula (e.g. self/2^`n times`) and loops (for loops). The duration of the signal is set 1 second using "Positive duration_(s) 1.0".

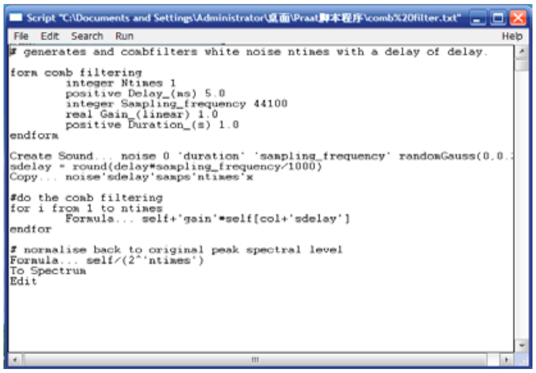


Figure 8: Praat script of white noise

The script is run after clicking the "Run" button of the "Control" menu in the Praat main window as indicated the following Figure9.

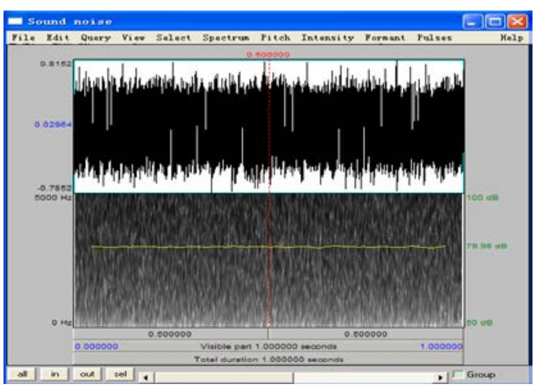


Figure 9: The figure of white noise

In Figure 9, the above figure is indicated white noise in time domain, and the under figure indicating the spectra of white noise. The students can see clearly the figure of white noise in time and frequency domain, even the details of the signal by clicking " in" button. They also can hear the "sound" of white noise so that it is helpful for understanding their knowledge about white noise. Of course, the teacher can further explain the nature of white noise (or random signal) via extracting different parameters of the white noise by the software.

2) Signal addition algorithms: The signal addition is a basic algorithm in signal processing. The addition of two signals is:

$$z(t)=x(t)+y(t) \tag{5}$$

where, x(t) and y(t) is two signals, z(t) is sum of them. In special circumstances, one of the signal is a constant (or x(t)=c, in which c is constant) in signal addition. The user can add a constant to the signal by clicking related button of Praat. But it is need to write a Praat script to realize the addition algorithm of two signals. The part of the script for signal addition algorithm indicated as the Figure10.

```
#find out how many Sounds have been selected
numberOfSounds = numberOfSelected ("Sound")

#set up arrays with names and IDs of selected Sounds
for ifile from 1 to numberOfSounds
   sound$ = selected$("Sound",'ifile')
   soundID = selected("Sound",'ifile')
   ids'ifile' = soundID
   names'ifile'$ = sound$
endfor
```

Figure 10: Part of Praat script for signal addition algorithm.

To note that it is important to choose the mode of addition before run the Praat script. Because, there are two modes, in which one is "Point-by-point values" mode the suitable for the discrete signal addition, the other is "Real time" mode that used for signal addition across different time domain and sampling rates. The real time mode is selected here, and the original two signals and the results of their addition are illustrated as the following Figure 11.

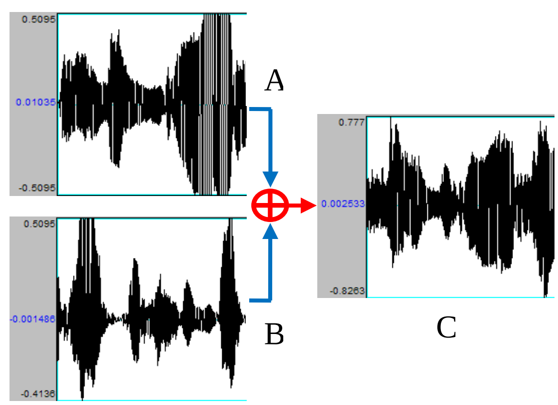


Figure 11: The addition charts of two signals.

In Figure5, the amplitude range of signal A and B is [-0.5095, 0.5095] and [-0.4138, 0.5095] respectively. C is the result of the two signals that its amplitude is ranging from -0.8263 to 0.777. The user can hear the effect of signal addition through playing the signal C by Praat.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

5. CONCLUSION AND FUTURE WORK

In this paper, a Praat based teaching approach for the discrete /real-time signal processing; especially for digital signal processing methodology is proposed. Praat and its script language show great effectiveness during the DSP teaching, so that students can see processed figures and can hear processed voice by Praat. This approach helps students to improve their understanding of some DSP algorithms and theories, even upgrade their programming ability. Furthermore, Praat is safe, easy-learned, standards compliant, and small-volume software that it would become one of the important computer based tools in DSP teaching.

In the future work, we would develop more and more Praat script programs to enhance its role in the signal processing field, especially in DSP area. We will broaden Praat's utilization from DSP teaching to the whole signal processing teaching.

ACKNOWLEDGEMENTS

This work is supported by the National Natural Science Foundation of China (No. 61163028), the Open Project of Xinjiang Laboratory of Multi-language Information Technology (No. 049807, 2013 years') and Second period fund of Educational Reform Engineering toward 21st Century Higher Education of Xinjiang University (No. XJU2008JGY11). And the author thanks to Professor Paul Boersma and Dr. David Weenink for developing the software, Praat.

REFRENCES:

[1] K. Steiglitz, *An Introduction to Discrete Systems*, Jhon-Wiley & Sons Inc. New York, 1974.

[2] H .Gethoffer, A. Lacroix, and R. Reiss, "A Unique Hardware and Software Approach for Digital Signal Processing", *Proceeding of IEEE International Conference on Acoustics, Speech, and Signal Processing(ICASSP)*, Hartford, Connecticut (USA), May 9-11, 1977, pp 151-154.

[3] A. Singh, "An Innovative Course on Real-Time Digital Signal Processing Applications", *Proceedings of the 29th Asilomar Conference on Signals, Systems and Computers(ASILOMAR)*, Pacific Grove, CA, USA, Oct. 30-Nov. 1, 1995. pp 88-92.

[4] S. Kim, K. I. Kum and Wonyong S, "Fixed-point optimization utility for C and C++ based digital signal processing programs", *IEEE Transactions on Circuits and Systems—II: Analog and Digital Signal Processing*, Vol. 45, No. 11, November 1998, pp. 1455 – 1464.

[5] S. K. Mitra, *Digital Signal Processing: A Computer-Based Approach*, McGraw-Hill Science Publisher, Boston, 2001.

[6] V. K. Ingle and J. G. Proakis, *Digital Signal Processing Using MATLAB*, Global Engineering Publisher, Stamford, 2011.

[7] S. Wolfram, *Mathematica: A System for Doing Mathematics by Computer*, Addison-Wesley publisher, Boston, USA, 1988.

[8] D. A. Lyon, H. V. Rao, D. Lyon, H. Rao, *Java Digital Signal Processing*, M & T Books publisher, New York, 1997.

[9] K. H. Hong, W. S. Gan, Y. K. Chong, K. K. Chew, C. M. Lee, T.Y. Koh, "*An integrated environment for rapid prototyping of DSP Algorithms using MATLAB and Texas instruments' TMS320C30*", *Microprocessors and Microsystems - Embedded Hardware Design* ,Vol. 24, No. 7, November 2000, pp. 349-363.

[10] C. S. Burrus, J. H. McClellan, A. V. Oppenheim, and T. W. Parks, *Computer-based Exercises for Signal Processing Using Matlab*, Prentice-Hall Publisher, Englewood Cliff, 1994.

[11] M. A. Yoder, B. A. Black, "Teaching DSP First with LabVIEW", *12th Digital Signal Processing Workshop, & 4th Signal Processing Education Workshop*, Wyoming (USA), Sept. 24-27, 2006, pp. 278-280.

[12] T. Bigg, J. Owen, R. W. Stewart, et al, "Rapid Prototyping Library for Adaptive Signal Processing Applications", *Proceedings of 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing* (*ICASSP*) , Phoenix, Arizona (USA), 1999, Vol. 4, pp. 2171 – 2174.

[13] K. Ubul, G. Ubul, and A. Aysa, "Macromedia Flash -Based Animations for Teaching of the Digital Signal processing principles", *Advanced Material Research*, Vol. 219-220, 2011, pp. 1518-1522.

[14] R. Chassaing, *Dsp Applications Using C and the Tms320C6X Dsk*, Jhon-Wiley& Sons Inc. New York, 2002.

[15] T. S. Hall, D. V. Anderson, "A Framework for Teaching Real-Time Digital Signal Processing With Field-Programmable Gate Arrays", *IEEE Trans. on Education*, Vol. 48, No. 3, August 2005, pp. 551-558.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[16] N. Kehtarnavaz, N. Kim, and I. Panahi, "Digital Signal Processing System Design: Using LabVIEW and TMS320C6000", *Proceeding of IEEE 11th Digital Signal Processing Workshop & 3rd IEEE Signal Processing Education Workshop*, Taos Ski Valley, New Mexico (USA), 2004, pp, 10-14.

[17] M. J. Korczynski, A. Hetman, and A. Hlobaz, "Virtual Laboratory a Key for Teaching Principles of Digital Signal Processing", *Proceeding of 2005 International Conference on Instrumentation and Measurement Technology Conference*, Ottawa (Canada), 2005 pp. 1222-1225.

[18] S. L. Toral, F. Barrero, M. R. Martı´nez-Torres, "Analysis of Utility and Use of a Web-based Tool for Digital Signal Processing Teaching by Means of a Technological Acceptance Model", *Computers & Education* , Vol. 49, No. 4, December 2007, pp. 957–975.

[19] N. Kehtarnavaz and C. Gope, "DSP System Design using LabVIEW and Simulink: A Comparative Evaluation", *Proceedings of 2006 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Toulouse (France), Vol. 2,2006, pp. 981-985.

[20] V. M. Ramesh and H. V. Sahasrabuddhe, "Exploring Data Analysis in Music using tool praat", *Proceeding of First International Conference on Emerging Trends in Engineering and Technology*, Nagpur (India), 2008, pp. 508-509.

[21] C. P. Moura, L. M. Cunha, H. Vilarinho, M. J. Cunha, D. Freitas, et. al, "Voice Parameters in Children With Down Syndrome", *Journal of Voice*, Vol. 22, No. 1, January 2008, pp. 34-42.

[22] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, *Numerical recipes in C: The Art of Scientific Computing*. Cambridge University Press, New York, 1992.

[23] D. Deliyski, M. K. Evans and H. S. Shaw, "Influence of data acquisition environment on accuracy of acoustic voice quality measurements", *Journal of Voice*, Vol. 19, No. 2, June 2005, pp. 176–186.

[24] K. Ubul, A. Hamdulla, and A. Aysa, "A Digital Signal Processing Teaching Methodology Using Praat", *Proceedings of 4th International Conference on Computer Science & Education (ICCSE),* Nanning (China), July 25-28, 2009, pp. 1804-1809.

[25] C. K. Campbell, *Surface Acoustic Wave Devices for Mobile and Wireless Communications*, Academic Press, New York, 1998.

[26] D. Weenink, Speech Signal Processing with Praat,*http://www.speechminded.comunpublished*, January 2013.