

# Time Domain Analysis of Sound Signal

Suchika Keshri

Student M.E. (0205EC14ME17), Electronics & Comm. Dept., S.R.I.T., Jabalpur, M.P, India.

Divyanshu Rao

Assistant Professor, Electronics & Comm. Dept., S.R.I.T. , Jabalpur , M.P, India.

Ravi Mohan

H.O.D PG Courses, Electronics & Comm. Dept., S.R.I.T. , Jabalpur , M.P, India.

**Abstract – Deliver message by voice is the most important, effective and common method of exchange information for mankind. Language is human specific features and human voice is commonly used tool which is also the important way to pass information to each other. The voice has large information capacity. So we can use modern method to study voice processing technology, so that people can easily transmit, store, access and apply the voice. In this paper we list out different properties of sound wave like spectrogram, amplitude spectrum, Probability distribution, Autocorrelation of sound signal. By implementing this, it will become easier for researcher to study and analyst the sound wave.**

**Index Terms – FFT , filter ,voice signal , MATLAB.**

## 1. INTRODUCTION

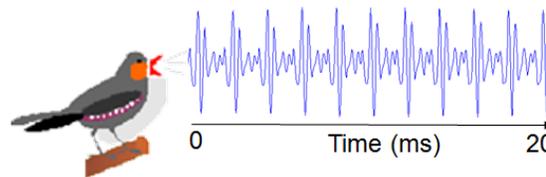
Introduction to spectral analysis

When listening to birdsong, it is immediately apparent that each song has a distinct rhythmic and sometimes even melodic structure. Songs of individual birds in a flock sound similar to each other and differ from those of other flocks. As early as the 18th century, Barrington noted that the songs of cross-fostered birds differed from the species-typical song, suggesting a role for vocal learning. However, until the late 1950s, there had been no objective way of confirming these observations by physical measurements of the songs themselves. The invention of the sound spectrograph (sonogram) at Bell Laboratories was a significant breakthrough for quantitative investigation of animal vocal behavior. The sonogram transforms a transient stream of sound into a simple static visual image revealing the time-frequency structure of each song syllable. Sonogram images can be measured, analyzed, and compared with one another. This allows the researcher to quantify the degree of similarity between different songs by inspecting (or cross-correlating) sonograms and categorizing song syllables into distinct types. Each song is then treated as a string of symbols, corresponding to syllable types, e.g., a, b, c, d..., and song similarity is estimated by the proportion of shared syllable types across the sonograms of the two songs. The procedure is equally useful in comparing the songs of

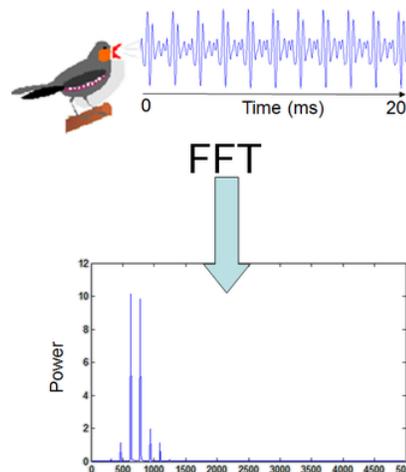
different birds and that of the same bird at different ages or after control and experimental treatments.

Spectral analysis is less than intuitive, and here is a little technical tutorial about how sonograms are computed:

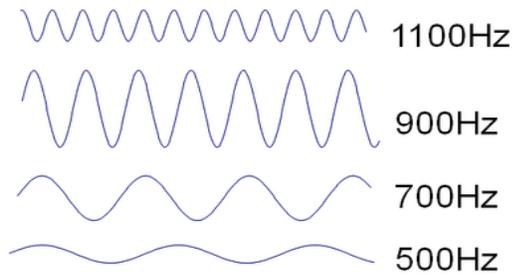
When recording a singing bird, the microphone capture tiny fluctuations in air pressure we call sound waves and turn those into an electrical current, which might look like this over 20 milliseconds:



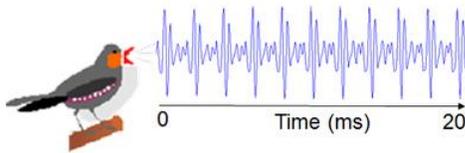
The fast Fourier transform (FFT) is an algorithm to compute the periodic structure in the signal, but representing it by a set of sine waves, called frequencies. Plotting the power of each one of those frequencies gives the Power Spectrum:



Each of the peaks in the Power Spectrum above, corresponds to a sine wave, e.g.,:

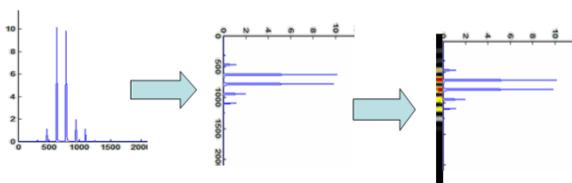


Now comes the interesting part: if we take those waves and add them, we will get a combo wave that looks like this:

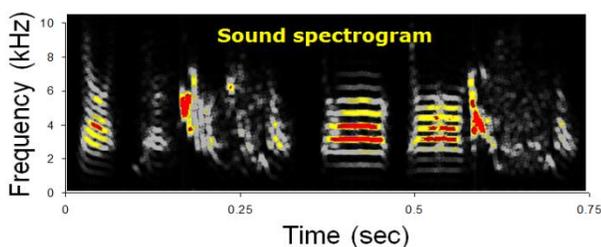


Exactly like the original wave, that is, the Power Spectrum (including phase, that is usually not shown) is a complete representation of the sound. We can look at the time domain (wave form) or frequency domain (the Power spectrum), and it summarizes its periodic structure.

Computing power spectrum works best when the signal is periodic and stationary, which is why in sound it usually makes sense to use short time windows, e.g., 5-20ms. To look at an entire song we have to summarize power spectra of many small time windows. This is how it is done: Take the power spectrum of one window, say from the first 20ms of the song, flip it and replace the graph with a color bar:



and now repeat this procedure for overlapping time windows, i.e., 1-20ms, 2-21ms, 3-22, etc, and stack the color bars next to each other. The outcome might look like this:



This is the famous sonogram! It is a set of vertical strips, each one represent the power spectrum of one time window of sound.

## 2. LITERATURE REVIEW

In paper Analysis And Synthesis Of Speech Using Matlab By Vishv Mohan “The interval of each sound wave has different frequency in its sub-sections. This paper has made an analysis of two matlab functions namely GenerateSpectrogram.m and MatrixToSound.m , in order to analyze and synthesis the speech signals. The first Matlab code section GenerateSpectrogram.m record the user input sound for user (more precisely from the source) defined duration and asks required parameters for computation of spectrogram and returns a matrix with frequency as rows and time as column and corresponding matrix element as amplitude of that frequency. MatrixToSound.m uses the method of additive synthesis of sound to generate sound from the user defined matrix with frequencies as its rows and time as its columns. Sound recording is an electrical or mechanical inscription of sound waves, such as spoken voice, singing, instrumental music, or sound effects. The two main classes of sound recording technology are analog recording and digital recording. Acoustic analog recording is achieved by a small microphone diaphragm that can detect changes in atmospheric pressure (acoustic sound waves) and record them as a graphic representation of the sound waves on a medium such as a phonograph. Digital recording converts the analog sound signal picked up by the microphone to a digital form by a process of digitization, allowing it to be stored and transmitted by a wider variety of media. Digital recording stores audio as a series of binary numbers representing samples of the amplitude of the audio signal at equal time intervals, at a sample rate high enough to convey all sounds capable of being heard. The feature of analysis and synthesis of sound, is applied to create the speech with the help of matrix of elements as frequency or time domain analyzed parameters with specific amplitude.”

The spectra of the sound corresponding to time can be computed using the GenerateTimeVsFrequency.m matlab file and its result matches approximately with that of specgramdemo function of the matlab. Additive synthesis of sound can be simulated with the help of the matlab file created MatrixToSound.m. It approximates the actual sound.

In paper Process and Analysis of Voice Signal by MATLAB by Nan Wu and Bofei Wang” uses Microsoft recorder to collect a voice, and then analyze its time-domain, the frequency spectrum and the characteristics of the voice signal. We use MATLAB’s function to remove the noise which has been added to the voice, further use bilinear transformation method to design a filter which is based on Butterworth simulation and window function and then filter the voice signal which has been added noise. After that we compare the time-domain and frequency-domain of the original voice and noised voice, then

playback the noised voice and de-noising voice and then compare the application of signal processing in FIR filter and IIR filter, especially in the perspectives of the signal filtering de-noising characteristics and applications. According to the comparison, we can determine which filter is the best.”

This chapter will know the best filter among those filters, and our work will be discussed and some shortage will be known. That will help to improve this collector in future. Play the sound in MATLAB by plus the noise processing. Its format is sound(y,Fs), sound(y) and sound(y,Fs,bits). There are big changes between filter. Since the voice we used is at high frequency, the high pass filter is the best. This design chooses a voice signal to be the research object, uses MATLAB to do frequency spectrum, add noise to simulate a noised voice signal. Design FIR and IIR digital filter based on digital signal processing theory, and filter the noised voice signal, then analyze the characteristics of time domain and frequency domain, finally replay the voice signal. Furthermore, create a signal processing interface; complete a system to analysis and process voice signal by MATLAB. Do spectrum analysis of the original voice signal and noised voice signal well, use FFT we get the spectrogram of the signal. In filter part, this project is mainly started with Butterworth filter to design filter, and has achieved the expected de-noising effect well. Finally we create an interface design, barely accomplished record, open voice signal file, show the wave form before and after filtering, but this interface has not used other widgets like list box, pull-downed menu and so on, which has not made full use of the GUI interface design function. From this thesis and design, conclusions can be achieved:

- 1) Using voice card and microphone in laptop which can be a sensor and collector in data collection system.
- 2) Using the strong data processing function in MATLAB, various voice signal analysis and process can be down well.
- 3) Using the filter in MATLAB, noise can be filtered and original sound will reappear.
- 4) Using GUI interface design in MATLAB, a good UI can be designed and we can get a complete system

Of course, there are lots of things which be improved in this design like power spectrum, making a more accurate filter parameter or smarter filter method and also this can be connected to LABVIEW, which optimizes the UI very much.

### 3. PORPOSED MODELLING

#### Software Approach

#### Overview of MATLAB

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems

and solutions are expressed in familiar mathematical notation. Typical uses include

- Math and computation
- Algorithm development
- Data acquisition
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or Fortran. The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most users of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

#### Design filter

#### Design of FIR filter

There are big differences in designing FIR filter and IIR filter. Here is the design step by using window function.

- (1) According to the index of stop-band attenuation and transition-band, choose the type of window functions and estimate the length of the window.
- (2) Construct a function that approaches the frequency response.

- (3) Calculate the transfer function.
- (4) Add the window function and get the result.

#### Design of IIR filter

The steps of design IIR filter are:

- (1) Transfer the technical index of the digital filter to the technical index of simulation low-pass filter.
- (2) According to the technical index of simulation low-pass filter to design the filter.
- (3) Transfer the function of simulation filter to the function of digital filter.

If the designed filter is low-pass filter, the design is finished. If the designed filter is high-pass filter or band-pass filter, there is also step.

- (4) Transfer the technical index of high-pass or band-pass filter to the technical index of simulation low-pass filter. And based on steps design the transfer function of the filter, and then transfer this function to digital filter.

#### Performance index of different filters

- (1) Performance index of low-pass filter

$f_p=1000\text{Hz}$ ,  $f_c=1200\text{Hz}$ ,  $A_s=100\text{dB}$ ,  $A_p=1\text{dB}$

- (2) Performance index of high-pass filter

$f_p=3500\text{Hz}$ ,  $f_c=4000\text{Hz}$ ,  $A_s=100\text{dB}$ ,  $A_p=1\text{dB}$

- (3) Performance index of band-pass filter

$f_{p1}=1200\text{Hz}$ ,  $f_{p2}=3000\text{Hz}$ ,  $f_{c1}=1000\text{Hz}$ ,  $f_{c2}=3200\text{Hz}$ ,  $A_s=100\text{dB}$ ,  $A_p=1\text{dB}$

#### Design of FIR and IIR filter in MATLAB

##### FIR filter

- (1) FIR low pass filter

The boundary frequency in band pass of this filter is 1000Hz, the cutoff frequency in stop band is 1200Hz, sampling frequency is 8000Hz, the maximum attenuation in band pass is 1dB, and the minimum attenuation in stop band is 50dB. Here is the procedure:

- (2) FIR high pass filter

The boundary frequency in band pass of this filter is 4000Hz, the cutoff frequency in stop band is 3500Hz, sampling frequency is 8001Hz, the maximum attenuation in band pass is 1dB, and the minimum attenuation in stop band is 50dB. Here is the procedure:

- (3) FIR band pass filter

The boundary frequency in band pass of this filter is 1200Hz and 3000Hz, the cutoff frequency in stop band is 3500Hz and

3200Hz, sampling frequency is 7000Hz, the maximum attenuation in band pass is 1dB, the minimum attenuation in stop band is 50dB. Here is the procedure:

##### IIR filter

- (1) IIR low pass filter

The boundary frequency in band pass of this filter is 1000Hz, the cutoff frequency in stop band is 1200Hz, sampling frequency is 8000Hz, the maximum attenuation in band pass is 1dB, the minimum attenuation in stop band is 50dB. Here is the procedure:

- (2) IIR high pass filter

The boundary frequency in band pass of this filter is 4000Hz, the cutoff frequency in stop band is 3500Hz, sampling frequency is 8000Hz, the maximum attenuation in band pass is 1dB, and the minimum attenuation in stop band is 50dB. Here is the procedure:

- (3) IIR band pass filter

The boundary frequency in band pass of this filter is 1200Hz and 3000Hz, the cutoff frequency in stop band is 1000Hz and 3200Hz, sampling frequency is 7800Hz, the maximum attenuation in band pass is 1dB, the minimum attenuation in stop band is 50dB. Here is the procedure: We use sin function in order to verify the usability of the designed filter. The results will be shown in next chapter.

##### Filtering

Use the designed filter to filter the voice signal with noise. FIR filter uses function `fftfilter` to filter the signal while IIR filter uses function `filter` to filter the signal. The function `fftfilter` uses overlap and adds operation to achieve the calculation of linear convolution.

##### Design GUI

First we create a new GUI file: Choose New/GUI, and then choose Blank GUI (Default). We can add buttons in the GUI, we can use the buttons indicate the function. When some buttons were set up, double clicked the button, so buttons properties were show in the figure below, then the performance and the value of the buttons will be adjusted easily. The button properties shows blow:

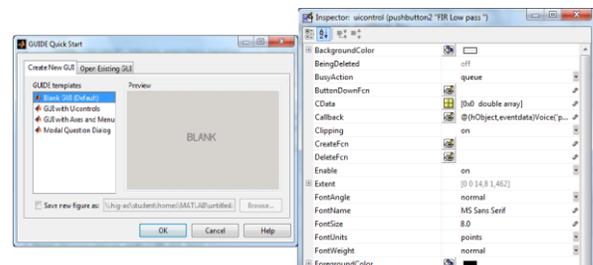
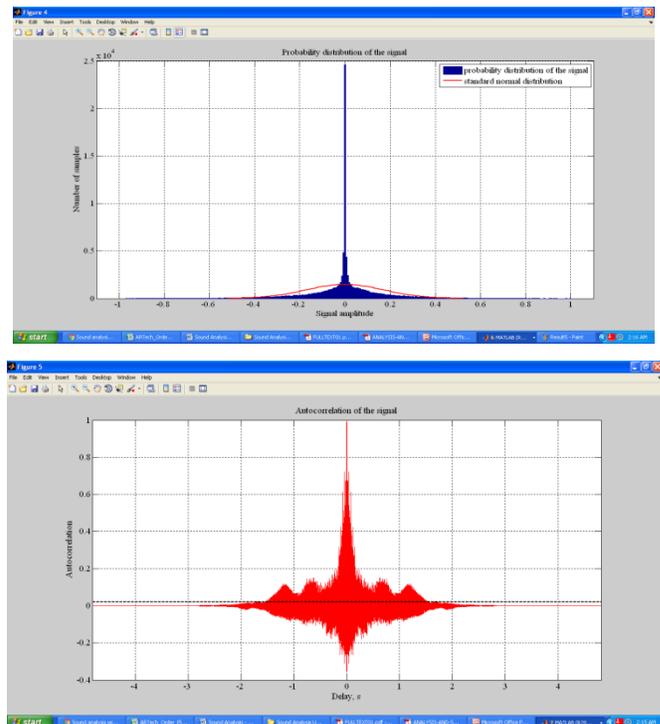
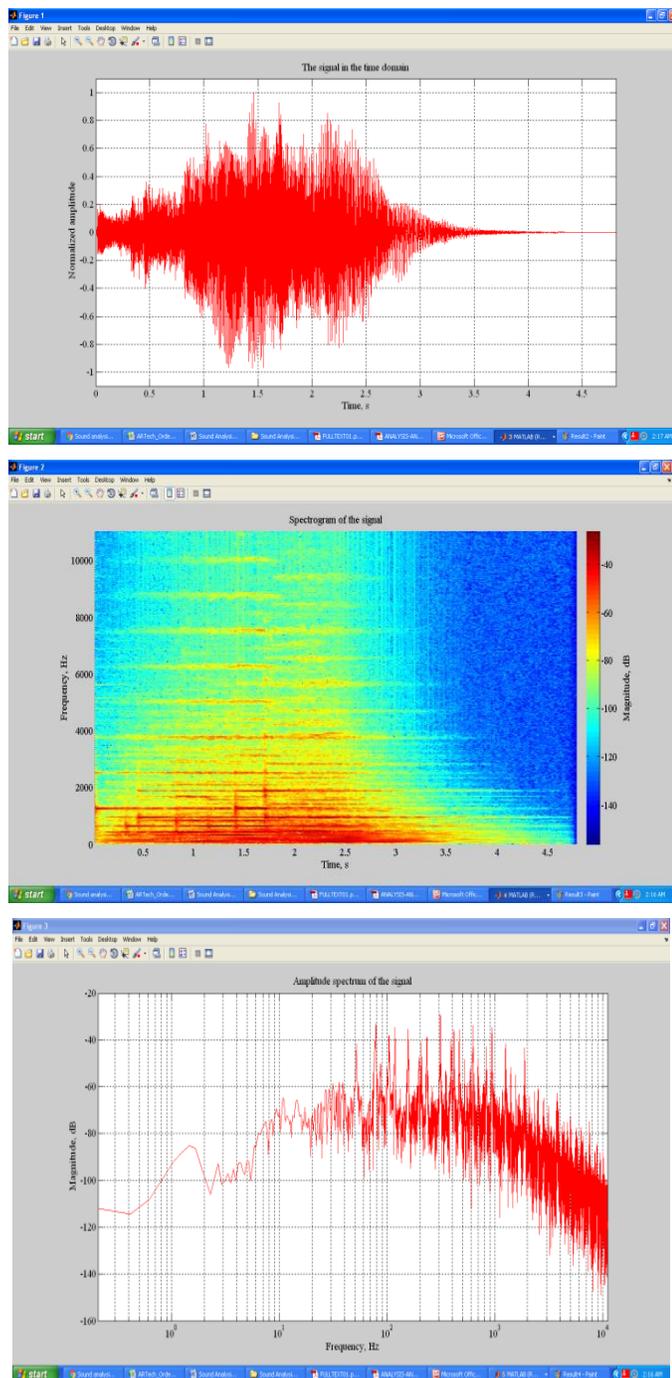


Figure shows the quick way to build a GUI interface in MATLAB, the left figure shows how to create GUI interface and the right one is button properties. Push save button to save the GUI into a FIG file, and an M-file will be created. Use the code that we wrote; just plug the code into the corresponding push buttons.

#### 4. RESULTS



#### 5. CONCLUSION

The spectra of the sound corresponding to time can be computed using the Sound\_Analysis.m matlab file and its result matches approximately with that of specgramdemo function of the matlab.

#### REFERENCES

- [1] Oppenheim, A.V., and R.W. Schaffer, *Discrete-Time Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1989, pp.713-718.
- [2] Rabiner, L.R., and R.W. Schaffer, *Digital Processing of Speech Signals*, Prentice-Hall, Englewood Cliffs, NJ, 1978.
- [3] P.I. Kattan and G.Z. Voyiadjis *MATLAB Guide to Finite Elements: An Interactive Approach*. 1st ed. New York: Springer Berlin Heidelberg, 2010.
- [4] L.R.Rabiner and R.W.Schafer, "Introduction to Digital Speech Processing", *Journal of Foundations and Trends in Signal Processing*, vol 1, no.1, pp:1-149, Jan. 2007. [3]Y. LI and M. GUO, "Audio Data Acquisition System Based on MATLAB", *Journal of Audio Engineering*, vol31, no.1, pp: 38-48, May 2007.
- [5] X. Chen, J. Liu and C.Wu, "Long-distance Data Acquisition System Design Based on MATLAB", *Journal of New Technology & New Process*, vol 10, no.1, pp:108-210. August 2007.
- [6] A.H. VanDoren, "Data Acquisition Systems", *Journal of Automatic data collection systems*, vol 1, no.1, p291, Jan.1982.
- [7] H. Wang, G. Wang, X Jin and N Bai "Application of data acquisition system based on sound card and MATLAB in Young modulus measurement", *Journal of Physics Experimentation*, vol 4, no.1, pp:205-246, April 2005
- [8] M. D. Lutovac, D.V. Tošić and B. L. Evans. *Filter Design for Signal Processing Using MATLAB and Mathematica*. 4sted. Beijing: House of Electronics Industry, 2004.
- [9] K. Feher, *Telecommunications Measurements, Analysis, and Instrumentation*, 1sted. New York: Englewood,1997.

- [10] M. D. Lutovac, D.V. Tošić and B. L. Evans. *Filter Design for Signal Processing Using MATLAB and Mathematica*, 4sted. Beijing: House of Electronics Industry, 2004.
- [11] L.D. Paarmann, *Design and Analysis of Analog Filters: A Signal Processing Perspective*, 3rded. America: Springer, 2001.
- [12] B. Venkataramani, *Digital Signal Processors*, 2nded. New Delhi: Tata McGraw Hill Education Private Limited, 2010.
- [13] B.A. Sheno, *Introduction to Digital Signal Processing and Filter Design*, 1sted. Canada: John Wiley & Sons, 2006.
- [14] E. Lai, *Practical Digital Signal Processing*, 1sted. America: Newnes, 2003. 28
- [15] D. Patrick, S. Fardo, *Understanding AC Circuits*, 2nded. America: Newnes, 1999
- [16] J.D. Foley and A.V. Dam, *Fundamentals of Interactive Computer Graphics*, 1sted. California: Wesley Publishing Company, 2005.
- [17] A. R. A. Sovijarvi, J. Vanderschoot, and J. E. Earis, "Standardization of Computerized Respiratory Sound Analysis," *Eur Repair Rev*, vol 10 (77), pp. 58, 2000.
- [18] J.E. Earis, B. M. G. Cheetham, "Current Methods used for Computerized Respiratory Sound Analysis," *Eur Repair Rev*; vol 10 (77), pp. 585-590, 2000.
- [19] Z. M. K. Moussavi, "Respiratory Sound Analysis," *IEEE Engineering in Medicine & Biology Magazine* 0739-5175, 2007.
- [20] Z. M. K. Moussavi, "Separating heart sound from lungs sounds," *IEEE Engineering in Medicine & Biology Magazine*, 2007.
- [21] Z. M. K. Moussavi, "Fundamentals of Respiratory Sound Analysis," Morgan & Claypool, First Edition.
- [22] W. F. Ganong, "Review of Medical Physiology (Seventeenth Edition), New Jersey, Prentice Hall, 1995.
- [23] Guyton & Hall, "A textbook of Medical Physiology", Tenth Edition, printed in India at Gopsons Paper Ltd., Noida 201 301.
- [24] Johnson, L.R., "Essential Medical Physiology (Second Edition). Philadelphia, Lippincott Williams & Wilkins, 1998.
- [25] MedicineNet.com, article.htm.
- [26] T.E. Ayoob Khan, Dr. P. Vijayakumar, "Separating Heart sound from lung sound using LavView", *International Journal of Computer and Electrical Engineering*, Vol. 2, No. 3, June, 2010 1793-8163.
- [27] Rahman Jamal, Mike Cerna, John Hanks *Designing Filters Using the Digital Filter Design Toolkit*.
- [28] Emmanuel. C. Ifeakor, Barrie. W. Jervis, *Digital Signal Processing*
- [29] Edward P. Cunningham, *Digital Filtering An Introduction*, John Wiley & Sons, Inc. 1995.
- [30] Haykin S, *Adaptive Filter Theory*, 2nd Edition, United States of America: Prentice Hall, Inc. 1991
- [31] Krister Landernas, *Implementation of Digital Filters Using Digital-Series Arithmetic's*, Department of electronics, Licentiate thesis No. 23, 2004
- [32] <http://www.dsptutor.freeuk.com/dfilt2.htm> Accessed on 2008- 11-22
- [33] <http://www.intersil.com/data/AN/an9603.pdf> Accessed on 2008-11-21
- [34] Nicholas John Loy, *An Engineer's Guide to FIR Digital Filters*, Prentice Hall, 1988
- [35] Cedric Keip, *Design of a Finite Impulse Response Filter Coefficients Computation tool using Evolutionary Strategies and Extension to Filter Bank Design*, 2004
- [36] Terrell, Trevor J. *Introduction to Digital Filters*. Second Edition, London: Macmillan, 1980
- [37] Thomas J. Cavicchi, John, *Digital Signal Processing*, Wiley & Sons, Inc 2000
- [38] <http://dualist.stanford.edu/~ee265/www>  
[Sept2006/docs/digital\\_filters.pdf](http://dualist.stanford.edu/~ee265/www/Sept2006/docs/digital_filters.pdf) Accessed on 2008-11-22