Real Time Speech Signal De-Noising Using Matlab

Ravi Shankar Kumar[1], Rajeshwar Singh[2]
[1, 2] Dept of Electronics and Communication Engineering
[1]Doaba Institute of Engineering and Technology,Mohali, Punjab
[2]Doaba Khalsa Trust Group of Institutions, SBS Nagar, Punjab

Abstract- *In the era of high speed network and expectation of users for real time quality of sound, transmission in telecommunication and accuracy of measurement systems are badly affected due to noise and distortion in the system. This has lead to major challenges for researchers for noise reduction and removal of distortion in the field of not only in speech processing but also in cellular mobile communication, image processing, radar, sonar, and medical image processing etc. A number of researchers have taken up this daunting task and proposed many solutions to it. However, de-noising of real time speech signal is still a major field of research. In this work, a real time speech signal is recorded using Microphone. Moreover, the idea is to implement the speech signal de-noising techniques such as decomposition, thresholding (soft) and reconstruction in the MATLAB simulation software, and elaborates a comparative analysis based on choice of wavelet transform over Fourier transform. Likewise, for the different level of decomposition, signal to noise (SNR) is estimated. To sum up, in this research, different circumstances are measured to elect best wavelet function and its level, based on its response of signal to noise ratio (SNR) in de-noising speech signal. Also, our signal processing technique recovers signal with a correlation higher than 99%. In analysis for real time speech signal with added Gaussian white noise, while using the technique we obtained a recovered signal with a correlation of 95%. This analysis is very useful to help the researcher understand the know how in removing noise from a signal by using wavelets.*

Keywords- De-noising, Wavelets, Gaussian White Noise, Correlation.

I. INTRODUCTION

Noise and distortion are the two major factors that limit the capacity of data transmission in telecommunications and that they also affect the accuracy of the results in the signal measurement systems, whereas, modeling and removing noise and distortions are at the core of theoretical and practical considerations in communications and signal processing. Another important issue here is that, noise reduction and distortion removal are major daunting challenges in applications such as; mobile communication, speech processing, image processing, medical signal processing, radar, sonar, and any other application where the desired signals cannot be isolated from noise and distortion.
Noise reduction, also called speech enhancement, refers to the process of recovering a speech signal of interest from noisy observations. It has a wide range of applications in voice communications and human-to-machine interfaces. Most early efforts focused on the single-channel case primarily because communication devices at that time were equipped with only a single micro-phone [1, 2].

Wavelet de-noising is considered a non-parametric method. Thus, it is distinct from parametric methods in which parameters must be estimated for a particular model that must be assumed a priori. Assume that the observed data

$$X(t)=S(t) + N(t) \text{ ---------------------(1)}$$

contains the true signal S(t) with additive noise N(t) as functions in time t to be sampled. Let $W(\cdot)$ and $W^{-1}(\cdot)$ denote the forward and inverse wavelet transform operators. Let $D(\cdot,\lambda)$ denote the de-noising operator with soft threshold λ. We intend to wavelet denoise X(t) in order to recover $\hat{S}(t)$ as an estimate of S(t). Then the three steps

$$Y=W(X)\ ,\quad Z=D(Y, \lambda),\ \hat{S}= W^{-1}(Z)$$

summarize the procedure. Of course, this summary of principles does not reveal the details of implementing the operators W or D, or selection of the threshold λ. B.

A continuous-time wavelet transform of f(t) is defined as:

$$CWT_{\Psi} f(a,b) = W_f(b,a) = |a|^{-\frac{1}{2}} \int_{-\infty}^{\infty} f(t) \overset{*}{\Psi}\left(\frac{t-b}{a}\right) dt \quad \text{----(2)}$$

Here a, b $\in$ R, a $\neq$ 0 and they are dilating and translating coefficients, respectively.

Remaining work of this paper of the paper is presented in Four Secctions. Section II is review of literature. In section III research methodology is presented. Section IV and V describe de-noising of sinusoidal and real time speech signal respectively. In section VI, conclusion and future scope is presented.

II. REVIEW OF LITERATURE

In research paper [1], authors demonstrated the application of the Bionic Wavelet Transform (BWT), an adaptive wavelet transform derived from a non-linear auditory model of the cochlea, to enhance speech signal. Authors in research paper [2] proposed a new speech enhancement method based on time and scale adaptation of wavelet thresholds. Also, proposed a method based on critical-band decomposition, which converts a noisy signal into wavelet coefficients (WCs), and enhanced the WCs by subtracting a threshold from noisy WCs in each sub band. Additionally, they proposed a gain factor in each wavelet sub band subject to a perceptual constraint. Authors in [3] have proposed new method for time–frequency analysis of speech signals. The analysis basis of the proposed Short-Time Fan-Chirp Transform (FChT) defined univocally by the analysis window length and by the frequency variation rate, that parameter predicted from the last computed spectral segments. Researcher also proposed an audio de-noising algorithm based on adaptive wavelet soft-threshold, based on the gain factor of linear filter system in the wavelet domain and the wavelet coefficients teaser energy operator in order to progress the effect of the content-based songs retrieval system. In paper [4], author has proposed a speech de-noising algorithm for white noise environment based on perceptual weighting filter, which united the spectrum subtraction and adopted auditory perception properties in the traditional Wiener filter. This paper [5] presents how the non-local means (NLM) algorithm may be involved in speech signals de-noising. This algorithm can be implemented in real time using a digital signal processor (DSP) such as ADSP2181 from Analog Devices. The performance of NLM algorithm is evaluated and compared with a wavelet based de-noising algorithm. Implementation issues are discussed in order to develop a real time system. In this paper [6] authors have come out with a useful de-noising or noise cancellation mechanism. For testing their idea and proposed mechanism, they used MATLAB for simulating the algorithms. The approaches they adopted in this paper has been implemented successfully and giving satisfactory results. The noise cancellation mechanism using mean power calculation (MPC) is working very efficiently according to test results and giving satisfactory performance. The experiment and testing whatever made in this paper are in the form of simulation and done using MATLAB. They have tested algorithms only with voice or speech signals. They thought of doing modification in algorithms for testing them with video signals and images also. They have also thought of implementing some de-noising or noise cancellation techniques in hardware circuitry, compare with adopted techniques, and make a comparative study between them in future. Authors do not analyze using time or frequency methods instead they propose de-noising technique based on mean power calculation. The average power of each speech frame is calculated. Then those frames are accepted which exhibits average power above certain threshold level. This paper [7] presented a two-step approach to the problem of multichannel noise reduction in the frequency domain. In the first step, a Householder transformation is constructed. In the second step, a Wiener filter is formed by using the noise-only or noise dominated component to estimate the noise in the speech-plus-noise component, thereby achieving noise reduction. The construction of the Householder transformation in each frequency bin requires the knowledge of the DOA information. If this information is not known *apriori,* which is often the case in practice, they derived a method that combines the noise reduction and DOA estimation together into one process. Simulations were conducted to validate the performance of the developed approach in both non-reverberant and reverberant environments. The results showed that it can improve the SNR significantly and this improvement increases with the number of microphones regardless whether there is reverberation or not. But, authors in this paper carried out research for multi-channel de-noising only

.

III. RESEARCH METHOLODY

Fourier transform based spectral analysis is the dominant analytical tool for frequency domain analysis. However, Fourier transform cannot provide any information of the spectrum changes with respect to time. Fourier transform assumes the signal is stationary, but speech signal is always non-stationary. To overcome this deficiency, a modified method-short time Fourier transform allows to represent the signal in both time and frequency domain through time windowing function. The window length determines a constant time and frequency resolution. Thus, a shorter time windowing is used in order to capture the transient behavior of a signal; we sacrifice the frequency resolution. The nature of the real speech signals is non-periodic and transient; such signals cannot easily be analyzed by conventional transforms. So, an alternative mathematical tool – wavelet transform must be selected to extract the relevant time-amplitude information from a signal.

We will describe a simple method for wavelet transform of a given signal. We choose a wavelet function, which is the mother wavelet, as well as determining the value of the signal wavelet scale, using the command wname. In this example we use a wavelet coif5 level 10. wname = 'coif5'; lev = 10;

In most of the signals of low frequency components that give the signal most of its information, while the high frequency components are responsible for incorporating particular features, that is why we subdivide the components of a signal into two categories: approaches (low frequencies) and details (high frequencies). To perform this decomposition the wpdec command is used which is responsible for creating a tree that consists of a vector in this case it is the signal to break down "y" the level to be decomposed lev = 10 and the type of wavelet used, which in this case is coiflet 5 and it is stored as a tree in the tree variable:

tree = wpdec(y,lev,wname);

We generate the coefficients of the decomposition of the signal using the wpcoef command where needed as the variable information and the number of tree node that has a better performance in the tree that was generated, which in this case is two. det1 = wpcoef(tree,2);

We determine noise threshold by using wpbmpen command that returns a global threshold THR for de-noising. THR obtained by a wavelet packet coefficients selection rule by means of using a penalization method provided by Birge-Massart. T (in our case tree) is a wavelet packet tree corresponding to the wavelet packet decomposition of the signal or image to be de-noised. SIGMA is the standard deviation of the zero mean Gaussian white noise in the de-noising model. alpha is a tuning parameter for the penalty term, which must be a real number greater than 1. The sparsity of the wavelet packet representation of the de-noised signal or image grows with alpha. Typically alpha = 2.

sigma = median(abs(det1))/0.6745;
alpha = 2;
thr = wpbmpen(tree,sigma,alpha);

We use command wpdencmp that performs a de-noising or compression process of a signal, when using wavelet packet. The ideas and the procedures for de-noising and compression by using wavelet packet decomposition are the same as those used in the wavelets framework. Wpdencmp (TREE,SORH,CRIT,PAR,KEEPAPP). It has the same output arguments, using the same options as above, but which were obtained directly from the input wavelet packet tree decomposition TREE of the signal to be de-noised or compressed. SORH ('s' or 'h') stands for soft or hard thresholding. Best decomposition performed using entropy criterion defined by string CRIT and parameter PAR. Threshold parameter is also PAR. In addition, if CRIT = 'nobest' no optimization is done, and the current decomposition is thresholded. If KEEPAPP= 1, approximation coefficients cannot be thresholded; otherwise, they can be:

keepapp = 1; xd = wpdencmp(tree,'s','nobest',thr,keepapp); All those parameters help us to de-noise our signals by using wavelets.

IV. DE-NOSING SINE WAVE SIGNAL

With MATLAB, it is possible to process noisy signals containing certain information, such as an audio one, in order to reduce the quantity of noise contained in it. Non-periodicity characterizes an audio signal, which is composed by a large number of different frequencies signals. This feature allows the use of conventional methods such as digital filters to eliminate noise mixed into the signal. In this section, we will introduce the treatment of noise in audio signals by using wavelet transforms, which are included as a tool in MATLAB. To do this we will start by using a sine wave signal, which is periodic and presents a well-known behavior. First, it is necessary to define the time interval k, and to define the angular frequency w.

k = 0:9.0703e-005:5; w=500*pi;

The variable h represents the argument of the sine function x in the next two instructions h=w.*k; x = sin(h);

Once we have defined the signal to be treated x, it is necessary to generate the noise source in order to be mixed with the original signal. One of the most common sources of noise is Gaussian white noise, which contains all frequencies. y = awgn(x,0,'measured'); wname = 'coif5'; lev = 10; tree = wpdec(y,lev,wname); det1 = wpcoef(tree,2); sigma = median(abs(det1))/0.6745; alpha = 2; thr = wpbmpen(tree,sigma,alpha); keepapp = 1; xd = wpdencmp(tree,'s','nobest',thr,keepapp);

Correlation between both signals, original and filtered one, is the parameter to compare them. We used the command crosscorr, which computes and plots the sample crosscorrelation function XCF between two univariate, stochastic time series.

The sinusoidal waveform x represents the signal to which is added white Gaussian noise; this will subsequently be reduced to recover the sine wave.

We choose cross correlation to evaluate the best performance; the cross correlation number that we select is the maximum at zero.

We add white Gaussian noise to sine waveform and the resulting graph is the plot shown in Fig.1. (a)

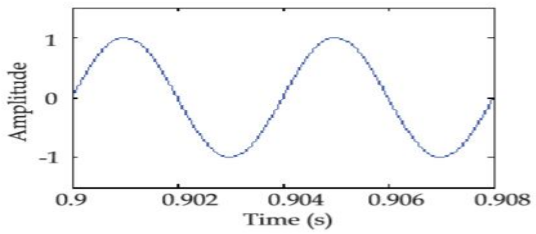

Fig.1(a): Original Sine waveform

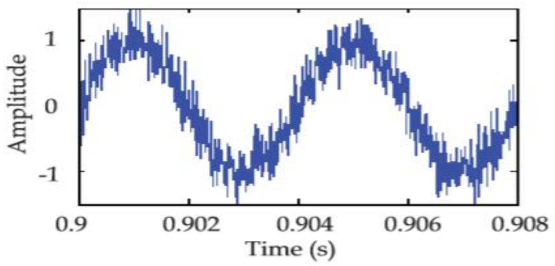

Fig 1(b): Sine waveform with Additive White Gaussian noise

Once we have defined the signal to be treated x, it is necessary to generate the noise source in order to be mixed with the original signal. One of the most common sources of noise is Gaussian white noise, which contains all frequencies. y = awgn(x,0,'measured'); wname = 'coif5'; lev = 10; tree = wpdec(y,lev,wname); det1 = wpcoef(tree,2); sigma = median(abs(det1))/0.6745; alpha = 2; thr = wpbmpen(tree,sigma,alpha); keepapp = 1; xd = wpdencmp(tree,'s','nobest',thr,keepapp);

To plot the XCF sequence without the confidence bounds, set nSTDs = 0.

D=crosscorr(x,xd);
Finally, we plot three figures.
figure(1) plot(k,xd)
hold on plot(k,y,'k')
hold on plot(k,x,'g')
legend('Denoise signal','Signal with AWGN','Original signal');
figure(2) plot(z,D)
legend('Correlation @ 0');

Sine wave signal and AWGN combined signal is shown in Fig. 2.

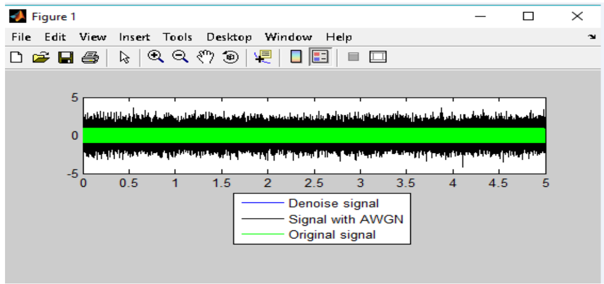

Fig.2: Sine waveform on application of addition of noise

The cross correlation between sine wave signal and de-noise signal is shown in Fig.3

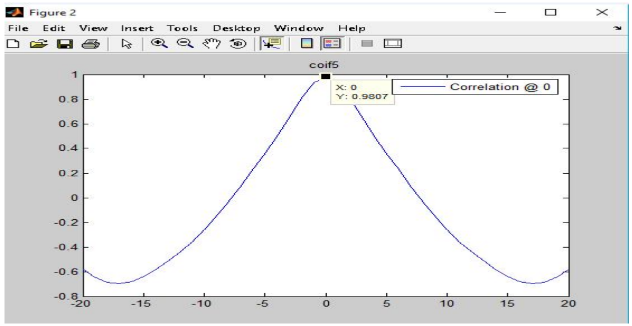

Fig.3: Cross correlation between original signal and de-noised signal

V. DE-NOISING OF REAL TIME SPEECH SIGNAL

In this case, a real time speech signal is recorded using the audio toolbox available in MATLAB Simulink. It is shown in Fig 4.

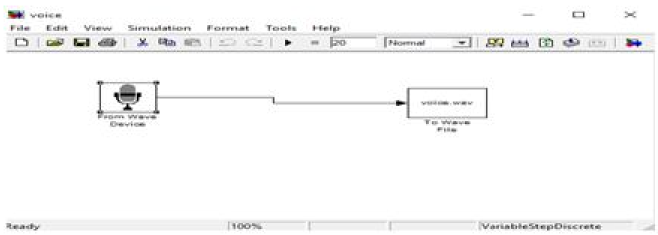

Fig.4: Recording of audio signal using MATLAB Simulink

Two useful methods for improving simulation throughput rates are increasing the signal frame size and compiling the simulation into native code: Increase frame sizes (and convert sample-based signals to frame-based signals) throughout the model to reduce the amount of block-to-block communication overhead. This can drastically increase throughput rates in many cases. However, larger frame sizes generally result in greater model latency due to initial buffering operations. Generate executable code with Real Time Workshop. Native code runs much faster than

Simulink, and should provide rates adequate for real-time audio processing. The dialog box is shown in Fig:.5.

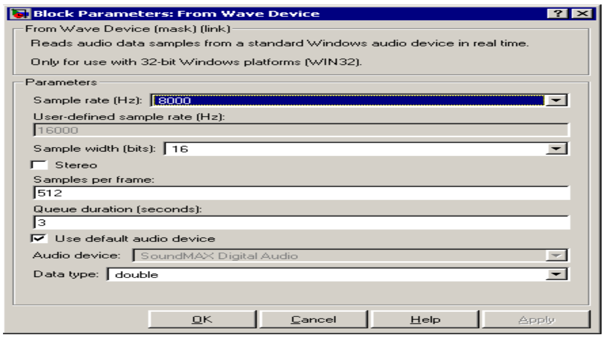

Fig.5: Dialog box to set the parameters of block ‘ from wave device’

The following functions are used from Communication Toolbox

Add white Gaussian noise to signal
y = awgn(x,snr)
y = awgn(x,snr,sigpower)
y = awgn(x,snr,'measured')
y = awgn(x,snr,sigpower,state)
y = awgn(x,snr,'measured',state)
y = awgn(...,powertype)
wavread - Read Microsoft WAVE (.wav) sound file
y = wavread('filename')
[y,Fs,bits] = wavread('filename')
[...] = wavread('filename',N)
[...] = wavread('filename',[N1 N2])
[...] = wavread('filename','size')
wavwrite - Write a Microsoft WAVE (.wav) sound file
wavwrite(y,'filename')
wavwrite(y,Fs,'filename')
wavwrite(y,Fs,N,'filename')
wpdec- (Wavelet Toolbox)- Wavelet packet decomposition 1-D
T = wpdec(X,N,'wname',E,P)
T = wpdec(X,N,'wname')
wpcoef (Wavelet Toolbox)- Wavelet packet coefficients
X = wpcoef(T,N)
X = wpcoef(T)
wpbmpen (Wavelet Toolbox)-Penalized threshold for wavelet packet de-noising
THR = wpbmpen(T,SIGMA,ALPHA)
THR = wpbmpen(T,SIGMA,ALPHA,ARG)
wpdencmp (Wavelet toolbox)- De-noising or compression using wavelet packets
[XD,TREED,PERF0,PERFL2] = wpdencmp(X,SORH,N,'wname',CRIT,PAR,KEEPAPP)
[XD,TREED,PERF0,PERFL2] = wpdencmp(TREE,SORH,CRIT,PAR,KEEPAPP)
crosscorr (GARCH toolbox)- Plot or return computed sample crosscorrelation function
crosscorr(Series1,Series2,nLags,nSTDs)
[XCF,Lags,Bounds] = crosscorr(Series1,Series2,nLags,nSTDs)

crosscorr(Series1,Series2,nLags,nSTDs) computes and plots the sample crosscorrelation function (XCF) between two univariate, stochastic time series. To plot the XCF sequence without the confidence bounds, set nSTDs = 0.
[XCF,Lags,Bounds] = crosscorr(Series1,Series2,nLags,nSTDs) computes and returns the XCF sequence.

Input Arguments:

Series1 - Column vector of observations of the first univariate time series for which crosscorr computes or plots the sample crosscorrelation function (XCF). The last row of Series1 contains the most recent observation.
Series2- Column vector of observations of the second univariate time series for which crosscorr computes or plots the sample XCF. The last row of Series2 contains the most recent observation.

nLagsPositive- scalar integer indicating the number of lags of the XCF to compute. If nLags = [] or is not specified, crosscorr computes the XCF at lags , where = min([20,min([length(Series1),length(Series2)])-1]).

nSTDsPositive - scalar indicating the number of standard deviations of the sample XCF estimation error to compute, if Series1 and Series2 are uncorrelated. If nSTDs = [] or is not specified, the default is 2 (i.e., approximate 95 percent confidence interval)

Output Arguments :

XCF- Sample crosscorrelation function between Series1 and Series2. XCF is a vector of length 2(nLags)+1, which corresponds to lags . The center element of XCF contains the 0th lag cross correlation.

Lags-Vector of lags corresponding to XCF(-nLags, ..., +nLags).

Bounds- Two-element vector indicating the approximate upper and lower confidence bounds, assuming that Series1 and Series2 are completely uncorrelated

The recorded speech signal is read using given command. The audio signal is saved as voice.wav

[x,Fs,nbits]= wavread ('voice'); Fig.6 shows original signal, signal with noise and de-noised signal respectively.

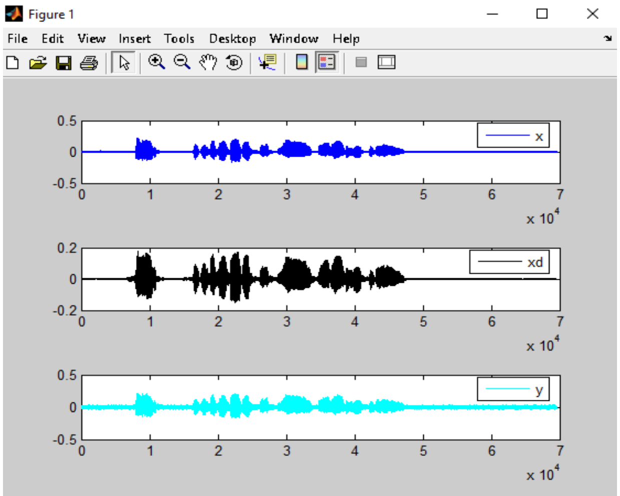

Fig.6: Original Audio signal, signal with noise and de-noised

To the recorded speech signal White Gaussian noise is added having AWGN as 10.

y = awgn(x,10,'measured'); wavwrite(y,Fs,'noisyvoice');

xd is the signal with noise.

xd = wpdencmp(tree,'s','nobest',thr,keepapp);

The Cross correlation between the original signal and de-noised signal is drawn and shown in Fig. 7 using D=crosscorr(x,xd);

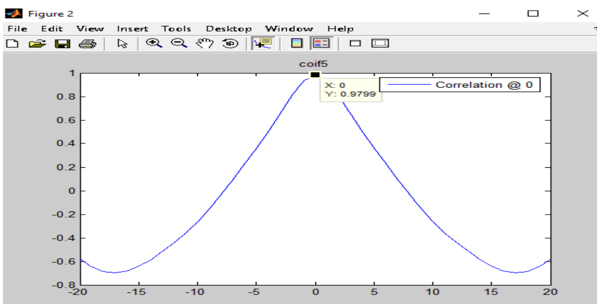

Fig.7: Cross correlation between original signal and de-noised signal

VI. CONSCLUSION

We provide a practical approach in how to put into practice wavelets in noisy audio data to improve clarity and signal retrieval. Also, MATLAB code is used to reduce the Gaussian white noise in periodic signals (sine function) and in speech signals (composed of several frequencies) using wavelet analysis. In order to de-noise real time speech signal, we take up a microphone speech signal. Moreover, the idea is to implement the speech signal de-noising techniques such as decomposition, thresholding (soft) and reconstruction in the MATLAB simulation software, and elaborates a comparative analysis based on choice of wavelet transform over Fourier transform. Likewise, for the different level of decomposition, signal to noise (SNR) is estimated. To sum up, in this research, different circumstances is measured to elect best wavelet function and its level, based on its response of signal to noise ratio (SNR) in de-noising speech signal.

Our signal processing technique recovers signal with a correlation higher than 99%. In analysis for audio signal with added Gaussian white noise, while using the technique we obtained a recovered signal with a correlation of 95%. This analysis is very useful to help the researcher understand the know how in removing noise from a signal by using wavelets. For further investigation, a uniform linear microphone array consisting of a number of omni- directional microphones will be used. Similarly, other types of noise signals can be added and signal can be processed to recover the original signal. Using other wavelet families, correlation can be observed between original signal and recovered signal and wavelet technique for higher correlation can be observed.

REFERENCES

[1] Johnson M. T, Yuan X and Ren Y, "Speech Signal Enhancement through Adaptive Wavelet Thresholding, Speech Communications", Vol. 49, No. 2, pp: 123-133, ISSN: 0167-6393, 2007.

[2] Bahoura M & Rouat J,"Wavelet speech enhancement based on time–scale adaptation, Speech Communication", Vol. 48, No. 12, pp: 1620-1637. ISSN: 0167-6393, 2006.

[3] Képesia M & Weruaga L, "Adaptive chirp-based time–frequency analysis of speech signals, Speech Communication", Vol. 48, No. 5, pp: 474-492. ISSN: 0167-6393, 2006.

[4] Dong E & Pu X. , "Speech denoising based on perceptual weighting filter", Proceedings of 9th IEE International Conference on Signal Processing, pp: 705-708, October 26-29, Beijing. Print ISBN: 978-1-4244-2178-7, 2008.

[5] Sorin Zoican, "Speech De-noising System with Non Local Means Algorithm" 9th International Symposium on Electronics and Telecommunications, IEEE, pp 315 – 318, , 2010.

[6] Arnab Pramanik , Rajorshee Raha, "De-Noising/Noise Cancellation Mechanism for Sampled Speech/Voice Signal", Ninth International Conference on Wireless and Optical Communications Networks IEEE Conference Publications, pp 1-4, 2012.

[7] Gongping Huang\ Jacob Benesti, and Jingdong ehen, "Study Of The Frequency-Domain Multichannel Noise

Reduction Problem With The Householder Transformation”, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE Conference Publications, pp 486 – 490, 2017.