White Noise Extraction Using Matlab and Xilinx ISE

Muhammad Izzul 'Afif Bin Ab Hadi
Electrical Engineering Department
Universiti Teknologi MARA, UiTM
Shah Alam, Selangor
afifabhadi@yahoo.com

***Abstract*—In this paper, a technique to eliminate noise and background noise from *.wav* signal was discussed. White noise is one of the noises that exist in ambient which are consisting of all audible frequency with equal intensity. In this work, waveform audio file format - WAVE, or more commonly known as *.wav* is use as a sample and the white noise from the signal was extracted using Butterworth filter. *Matlab* were employed to generate the pure white noise. The results show the performance of the filter and the comparison between the input and output signal. A comparison between *Spartan 3E* starter board and *Xilinx* implementation is will made using the same filter and the same input sound signal.**

Keywords-eliminate noise ; white noise; ; extract white noise ; butterworth filter; Matlab

I. INTRODUCTION

Noise is a nuisance or disturbance during communication, conversation and human hearing and it is unwanted. However, in data processing or computing it can be considered as unwanted data without meaning. Noise occurs because of many factors - interference, delay and overlapping. In sound signal, noise is very problematic because it will make the understanding of the information difficult to understand. Adaptive noise control (ANC) is a method for reducing unwanted sound. ANC is achieved by using the computer, which analyzes the waveform of the noise signal, then generates a signal reversed waveform to cancel it out by interference [1].

In this project, the white noise will be extracted from ***.wav*** signal. This ***.wav*** sound signal was recorded at the live concert. This recorded sound cannot be heard clearly because of the scream and shout that coming from the audience presence. In other words, the ***.wav*** file consists of the ambient or background noise which able to disturb the hearing of human ears. This is the description or a way to think about the white noise. Let's say two people are talking at the same time. Human brain can normally pick one of the two voices and actually listen to it and understand it. If three people are talking simultaneously, your brain can probably still pick out one voice. However, if 1,000 people are talking simultaneously, there is no way that your brain can pick out one voice. It turns out that 1,000 people talking together sounds a lot like white noise. Thus, based on to this recorded sound signal mention previously, the screaming and shouting of thousands audience there will probably produce the white noise [2]. The *Matlab* tools are used to investigate the noise in the sound signal using Butterworth filter. The Butterworth filter is then filtered the input signal by reducing the magnitude of the selecting region of cutoff frequency and produce the output signal based on the white noise specification – this was made by comparing the input and output signal. In order to show the performance of this filter, hardware implementation using *Spartan 3E* starter board is used to verify the extraction of the noise signal. *Xilinx* software is used as the Verilog interface to design the filter.

II. RESAERCH METHODOLOGY

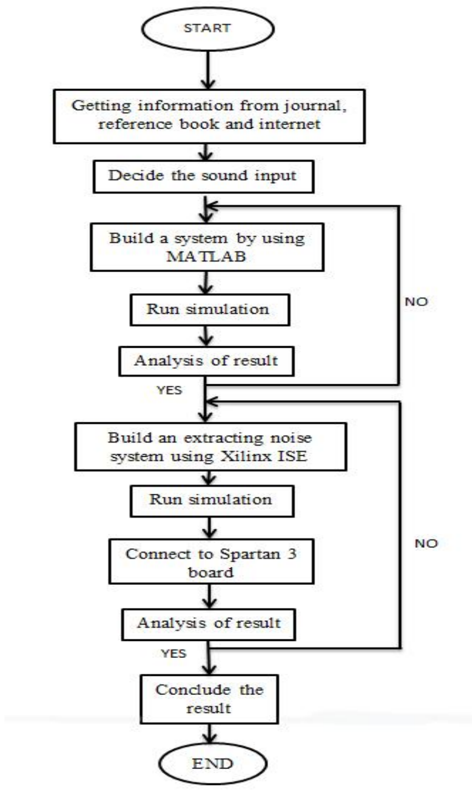

Fig.1: Methodology of the entire project

This project is starting with gathering all the information from any journal, reference book and internet. After doing some research based on what the project is going to be, the input sound is then decided. The input sound used in this project is the recorded sound which was recorded in the live concert and wide open to noise. After the input signal is decided, a filter design is build using *Matlab*. Here, many type of filter already test and the best filter is choose based on the simulation of the input signal. The graph result is then analyzed to see whether it is met the white noise specification or not.

After completing the summation in *Matlab*, the next step is to design the same system using *Xilinx ISE* software to prove that the filter system that already design in *Matlab* can perform the same result and the same output signal in *Xilinx ISE*. When the result is met the specification, the *Spartan 3E* board is use as the external hardware. The output signal which already filter will be load to the *Spartan 3E board* and the sound then play and listen whether the output sound is which noise or not.

The following section describes the elements been used to simulate the filter using *Matlab.*

A. Bandpass Filter and Its Application

Bandpass filter is one types of filter that reject the cut-off frequency of two points: high cut-off frequency and low cut-off frequency. The width of the bandpass region can be varied depending on the applications where a particular band or spread or frequency needs to be filtered from a wider range of mixed signals [3].

The bandpass filter is a combination of the properties of low pass and high pass filter into a single filter. The process of the bandpass filter is showed in Fig.2. The block diagram depicts the sequence of how the signal is filtered using bandpass filter. The low pass filter will block the frequencies that higher than the low cut-off frequency whereas the high pass filter will reject the lower frequency of the high cut-off frequency [4].

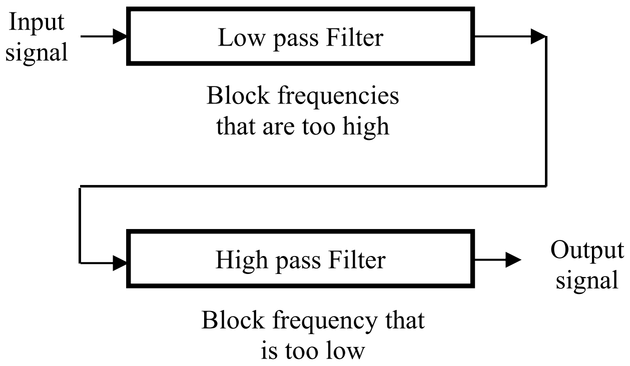

Fig.2: Block diagram of the combination of low pass filter and high pass filter as a bandpass filter.

B. Butterworth Filter Approximation

The Butterworth filter is a type of signal processing filter designed to have a flat frequency response in the passband. It is known as a classical method of analog filter and maximally flat filters. The Butterworth filter is defined as follows:

$$|H(j.\omega)|^2 = \frac{1}{1+\left(\frac{\omega}{\omega_{scale}}\right)^{2N}}$$

where $\omega = 2\mu f$ radian frequency, $\omega_{scale} = 2\mu f_{scale}$ is a constant scaling frequency, and N is the number of order of the filter. In this simulation, the number of order used is N=2.The magnitude of $|H(j.\omega)|^2$ is maximum at $\omega = 0$ [5].

The *Matlab* code used to invoke the function of Butterworth filter is as follows:

```
Fs = 44100;  % Sampling Frequency
N   = 2;    % Order
Fc1 = 2000;   % First Cutoff Frequency
Fc2 = 10000;  % Second Cutoff Frequency

h  =fdesign.bandpass('N,F3dB1,F3dB2', N, Fc1, Fc2, Fs);
Hd = design(h, 'butter');
output = filter(Hd,Y);
```

F_s are the sampling of frequency where the recorded ***.wav*** sound signal has purely 44.1 kHz of frequency. This sample of frequency was taken when the input signal was load to the *Matlab*. N is the number of order of the filter. The number of order is decided as N = 2. F_{c1} is the low cut-off frequency of the filter; where the beginning point of the filter will start to filter the signal. As mention, the white noise is a random noise that has equal energy at each frequency. The F_{c1} = 2 kHz is set for the low cut-off frequency in this work. F_{c2} is the high cut-off frequency and it is decided to be at the frequency of 10 kHz for this simulation.

C. WAV Signal

Waveform Audio File Format (WAVE or more commonly known as ***.wav*** due to its filename extension) is a sample sound that will be used as the input sound, which already recorded during a concert [6]. This input sound is recorded at the concert because it is aim to record any noise that exists from ambient and environment. This recorded sound is hardly to be heard clearly because the sound consists of the screaming and shouting from the audience presence.

In this simulation, the ***white.wav*** sound will be filtered to eliminate the sound and produce the pure white noise.

III. RESULT AND DISCUSSION

Fig.3 shows the input or the original sound obtained from the recorded sound in live concert. Whereas, the output signals in Fig. 4 is the filtered audio signal from the input signal. Both signals are the comparison between input signal and output signal that had been obtained using *Matlab*. As shown in Fig.4, the circle with red color depicts the reductions in the amplitude of the output signal. These signals show that the frequency is filtered using Butterworth filter. These ripples are much lower compared to input signal. However, the different of these two graphs cannot be seen clearly because the sample range is 44.1 kHz – this ***.wav*** signal consists of too many data, and every data that already filtered cannot be seen clearly. Thus, to illustrate the filter operations the signal is reduced to 40 Hz as shown in Figs. 3 and 4. The power spectral density is scaled for 20 Hz as shown in Figs.5 and 6.

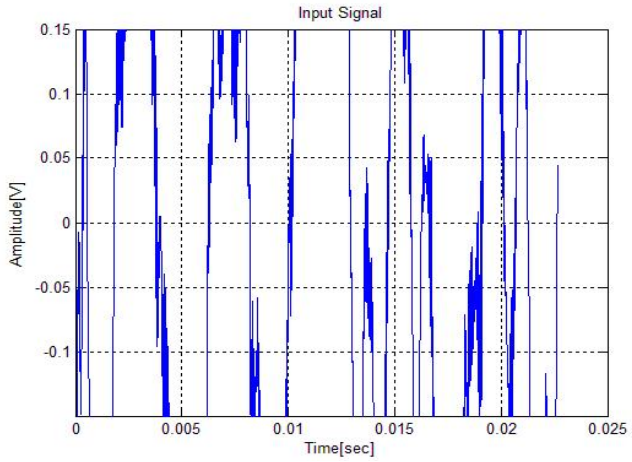

Fig.3: Input signal of original sound

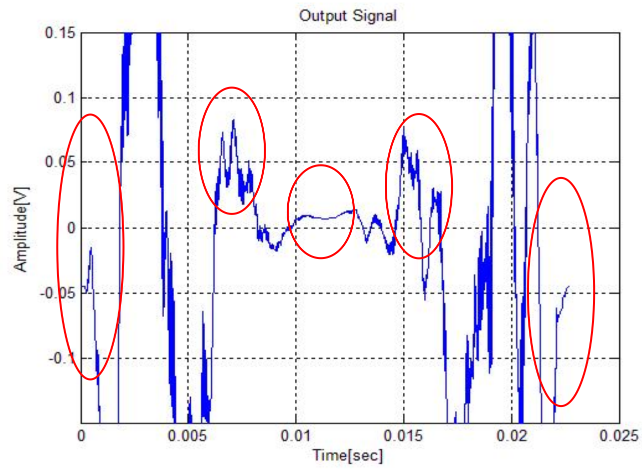

Fig.4: Output signal of original sound

As mentioned previously, the low-cutoff frequency was set at 2 kHz. This result can be observed by the first red circle shown in Fig.4. – there are decreasing in output signal amplitude compared with the input signal due to the low-pass filter. The subsequent red circles also show the decreasing of the ripple due to high cutoff frequency set at 10 kHz. This is the starting point for the high cut-off frequency using high pass filter.

The task of high pass filter is just opposite of a low pass filter which offers an easy passage of high frequency. However, it is difficult to block a low frequency signal. The combination of low pass and high pass filter in turn, will result the signal changed based of the cutoff frequencies and this combination can be examined with bandpass filter.

Fig 5 and 4 shows the comparison between the input and output signal of the power spectral density (PSD) of the sound signal. PSD is a function to show the variation as a function of frequency. In other words, it shows at which frequencies variations are strong and which variation frequencies are weak. Computation of PSD is done directly by the method called Fast Fourier Transform (FFT) or computing autocorrelation function and then transforms it.

The Fourier transform defines a relationship between a signal in the time domain and its representation in the frequency domain. The Fourier transform of a signal is a continuous complex valued signal capable of representing real valued or complex valued continuous time signals. Mathematically, switching between the two which are the time domain to frequency domain can be expressed as:

$$|X| = \sqrt{X_r^2 + X_i^2} \text{ and } \quad \angle X = \tan^{-1}\left(\frac{X_i}{X_r}\right)$$

or equivalently,

$$X_r = |X| \cos(\angle X) \text{ and } X_i = |X| \sin(\angle X)$$

$|X|$ and $\angle$X are the magnitude and phase of the complex number respectively, and X_r and X_i are the real and imaginary components of the complex number respectively. The magnitude is plotted on a dB/Hz scale, as shown in Figs.5 and 6. This showed that the amplitude is changed to power in dB, and the time is converted to frequency.

The Fourier transform is defined by the equation:

$$X(f) = \int_{-\infty}^{\infty} x(t)\, e^{-j2\mu f t}\, dt$$

where *X(f)* is the Fourier transform of *x(t)* frequency is measured in Hertz, with f as the variable frequency [7].

Fig.5 shows the PSD of the input signal obtained from Fig.3 using FFT. The result shows the PSD of the signal is almost - 40dB and decrease abruptly until it reached - 90dB at 13 Hz. Then the signal start to increase uniformly at approximately 13 Hz until it reached amplitude of - 65dB.

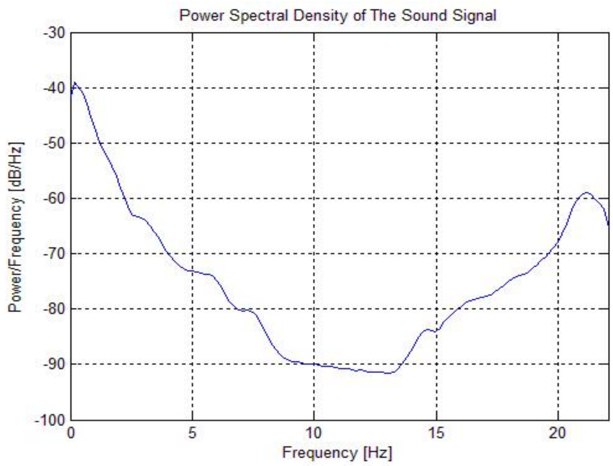

Fig.5: Power spectral density of the input signal

Fig.6 shows the output of PSD obtained from the output signal in Fig. 4. The output shows the different in amplitude compared to the input signal (Fig. 5). The magnitude of the output value is getting smaller. It can be seen when the frequency is at 2 kHz the amplitude is decreased – 46 dB to -98 dB. In other words, if the frequency is higher than f_{c1}= 2 kHz, the frequency is blocked. In contrast, at f_{c2}= 10 kHz signal start to increase because of the high pass filter – this implied that frequency lower than 10 kHz is rejected.

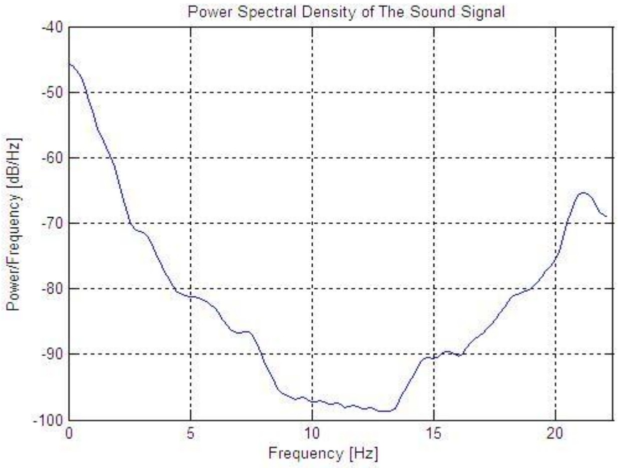

Fig 6: Power spectral density of the output signal

Fig.7 shows the frequency spectrum***.wav*** sound signal filtered using Butterworth filter. The Fourier transform function is used because it is necessary to analyze the noise in the sound magnitude. In frequency spectrum, the magnitude cannot be measured easily because of ripples. These ripples come from random noise of sound signal that is not filtered by the Butterworth filter. In Fig.8 is the PSD of output signal that is already filtered in Butterworth filter. The frequency spectrum in Fig. 7 is transformed to power spectral density as shown in Fig.8 - the pattern of graph is smooth and easy to collect the data of the output signal. The purpose to collect the data is to see how large and how many differences between input and output signal.

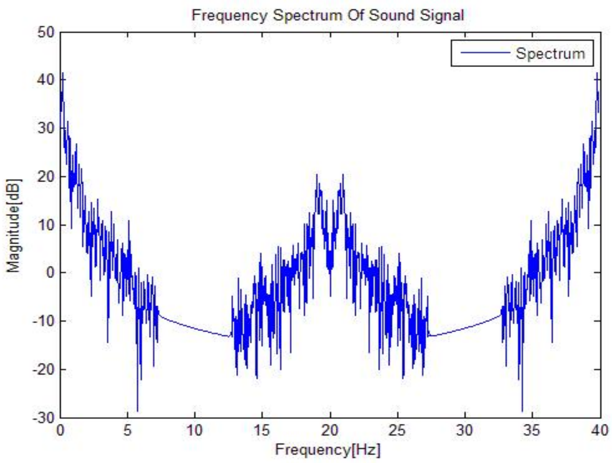

Fig.7: Frequency spectrum of output signal

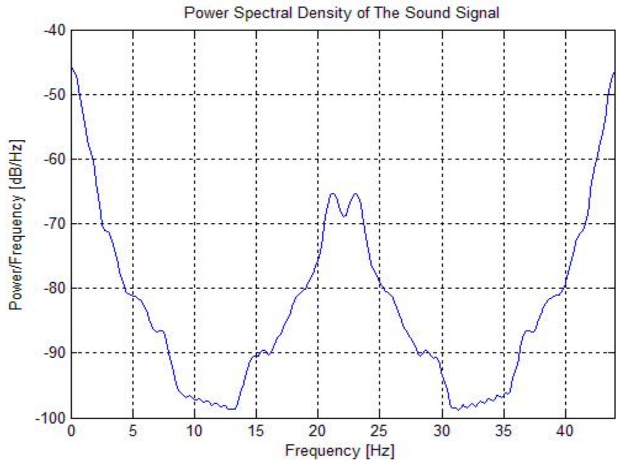

Fig.8: Power spectral density of the output signal

The output result was not as expected because the output signal seems not much different with the input signal. This is because white noise range is between 0 to 0.05 in magnitude. This also can be seen in Fig. 4 circled in red. Furthermore, the cut off frequency of the Butterworth filter does not match with the necessary frequency. This ***.wav*** signal contains thousands of data. However, the design of the filter system collected from the input and the output are 1000 data. The input and the output 0 to 0.025 seconds contains thousands of data signal and when this signal is entering the filter there are a little difference between the input and the output signal. The possible explanation is the Butterworth filter can only filtered the signal that only in the range of its bandpass, hence the filtered input signal is not much different from its output signal. In addition, the cutoff frequency also was adjusted so that it can eliminate the unnecessary sound, but the result do not meet with the expected result - this is maybe because of too many data signal passing through it.

The implementation of Butterworth filter in *Xilinx Spartan-3E* starter board was in progressed. The board contains a *Xilinx XC3S500E Spartan-3E FPGA* with up to 232 users - *I/O pins*, *320-pins FPGA* packages and over 10 000 logic cell. The Spartan-3E starter kit also provided the *MicroBlaze* 32-bit embedded RISC processor and the *Xilinx Embedded Development Kit (EDK)*. With those features, the *Xilinx Spartan-3E* starter kit is well suited for hardware implementation of Butterworth filter. The purpose of designing the Butterworth filter in *Xilinx ISE* is to do the comparison between the software simulation and hardware implementation [8].

IV. CONCLUSION

In this work, the Butterworth filter was used to extract the white noise and eliminate the unwanted sound. The result of the simulation using *Matlab* software is already made and the original audio signal already been filtered by using Butterworth filter. The output audio signal also been performed and there are different between the input audio signal. The output sound also been played after the simulation are done.

REFERENCES

[1]. Allam Mousa, Marwa Qados, Sherin Bader "Adaptive Noise Cancellation Algorithm Sensitivity to Parameters", International Conference on Multimedia Computing and System (ICMCS), 2011, 7-9 April 2011, p.1-5.

[2]. How Stuff Works, Inc.1998-2012. *White Noise* [Online]. Available at: http://science.howstaffworks.com/question47.htm.

[3]. Teri Hamill, Ph.D. Nova Southeastern University. 2009, September, 27. *Filter* [Online]. Available at: http://audsim.com/csd7000/filterhtml/filters2.html.

[4]. Michael Stutz. 1999-2000. *Band-pass Filter* [Online]. Available at: http://www.allaboutcircuits.com/vol_2/chpt_8/4.html.

[5]. Matheonics Technology Inc. 2009. *Butterworth Approximation* [Online]. Available at: http://www.matheonics.com/Tutorials/Butterworth.html.

[6]. Wikipedia. 2012, May, 22. *Wav* [Online]. Available at: http://en.wikipedia.org/wiki/WAV.

[7]. Bracewell, *The Fourier Transform and Its Applications* (3rd ed), Boston: McGraw-Hill, 2000.

[8]. Tian Lan, Jinlin Zhang. "FPGA Implementation of an Adaptive Noise Canceller", 2006 International Symposium on Information Processing, ISIP 2008, 23-25 May 2008, p.553-558.

[9]. Woon S. Gan, Sen M. Kuo. "An Integrated and Active Noise Control Headsets", IEEE Trans. On Consumer Electronics, vol.48, issue: 2, May 2002, p.242-247.

[10]. Jalal Taghia, Jalil Taghia, Nasser Mohammadiha, Jinqiu Sang, Rainer Martin. "An Evaluation of Noise Power Spectral Density Estimation Algorithms in Adverse Acoustic Environment", IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2011, 22-27 May 2011, p.4640-4643.