Simulation of LMS Algorithm for Active Noise Cancellation

Ahmad Khairul Hafiz B Abdullah
Department of Electrical Engineering, University Technology MARA, Shah Alam, Selangor
Email: khairulhafiz1988@gmail.com

***Abstract*- This paper presents the simulation of (least mean square) LMS method using *Matlab* to adjust the coefficient of the adaptive filter in order to match the desired response with mp3 music data as external source. *Xilinx ISE* is employed to design the adaptive filter by using *Verilog* programming.**

Keywords : Active noise control, Adaptive algorithm, LMS, Verilog.

I. INTRODUCTION

An active noise control (ANC) is used to reduce noise [1]. It can digitally design to minimize noise effect with an adaptive algorithm**.** Adaptive filters are dynamic filters which iteratively altered their characteristics in order to achieve an optimal desired output [2]. An adaptive filter algorithmically alters its parameters in order to minimize a function of the difference between the desired output *d(k)* and its actual output *y(k)* [2]. To illustrate, Fig. 1 shows a block diagram of adaptive noise cancellation system [2]. The reference signal *x(k)* is a signal that same signal with the noise signal *n(k)*. The block "*filter"* represents the digital filter used to processed to produce an digital filter output *y(k)*, from *x(k)* [3]. This filter will equalize output filter *y(k)* to desired input *d(k)* [1]. The desired input *d(k)* is also denoted as the input sound signal *s(k)* with the unwanted noise environment *n(k)* [3]. The process of equalization will take into account when at each iteration of the error signal *e(k)* is feedback into the LMS algorithm filter, where the filter characteristics are altered by adjusting the FIR coefficient *w(k)* accordingly [2].

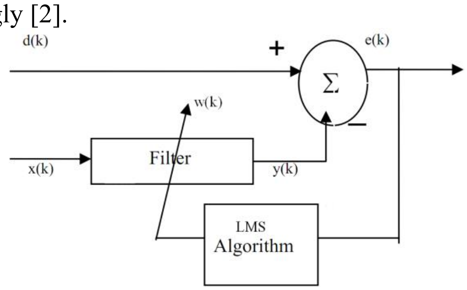

Fig 1: Adaptive filter as active noise control

In this work, the concept of adaptive filter is implemented using *Matlab* simulation. The simulation of the LMS algorithm is used to calculate the difference between the desired output *d(k)* and the error signal *e(k).*The FIR coefficient is changed algorithmically in order to minimize the *e(k).* Means that, LMS uses a stepsize parameter μ, input signal *x(k)* and the difference of desired signal *e(k)* and filter output signal *y(k)* to frequently calculate the update of the filter coefficient. Fig.2 shows the *Matlab* model based ANC, where the system shown in Fig.3 was design using *Verilog* code.

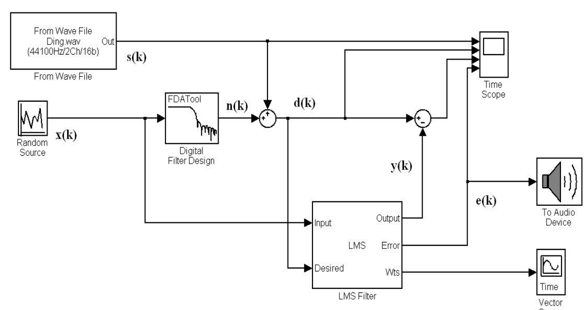

Fig 2: MATLAB model LMS based ANC

II. Theory Of Least Mean Square (LMS) Algorithm

LMS algorithm is a type of adaptive filter known as stochastic gradient-based algorithms as it utilizes the gradient vector of the filter tap weights to converge on the optimal wiener solution [2]. Each iterations of the LMS algorithm require three distinct steps order:

1. The output of the FIR filter, *y(k)* is calculated using equation (1)

$$\boldsymbol{y(k)} = \sum_{i=0}^{K-1} \boldsymbol{w(k)x(k-i)} = \boldsymbol{w^T(k)x(k)} \quad \textbf{(1)}$$

2. The value of the error estimation is calculated using equation (2)

$$e(k) = d(k) - y(k) \quad (2)$$

3. The tap weights of the FIR vector are updated in preparation for the next iteration, by equation (3)

$$w(k+1) = w(k) + 2\mu e(k)x(k) \quad (3)$$

The LMS algorithms are favored in adaptive filtering due to its computational simplicity and easier to implement than all other commonly used adaptive algorithms [2].

III. LMS Algorithm Implementation

The LMS algorithm is designed as shown in Fig. 3. The design is based on equation (3). The coefficients were calculated according to equation (3). For this design, the delay is necessary to separate the current coefficients *w(k)* from the next set of coefficients *w(k+1)* [5]. The LMS algorithm design component consists of a multiplier and an adder with delay and feedback to update the filter coefficient [5] . The delays are simply D Flip-Flops [6] . According to equation (3) the filter output *y(k)* is subtracted from the desired input *d(k)* to produce an error signal *e(k)*. The error signal *e(k)* is then multiplied with stepsize μ and the input signal *x(k)* and then added with the current coefficient *w(k)*. For this design, *Verilog* code has been used to implement on FPGA hardware for future work.

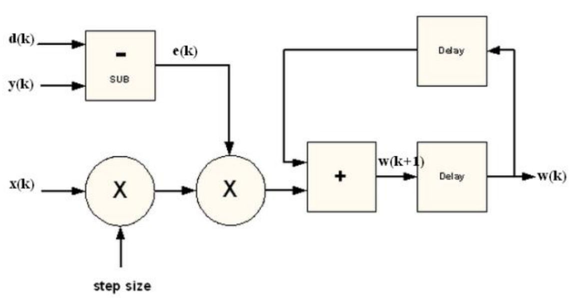

Fig3: LMS Algorithm structural

IV. Research Methodology

Fig. 4 shows the flow chart of the works that has been outlined to accomplish this task. Firstly, the type of adaptive algorithms was selected. There are several types of LMS - Least Mean Square (LMS) normalized LMS (NLMS) and Recursive Least Square (RLS). In this project, Least Mean Square (LMS) algorithm was preferred. Then, the ANC system is designed by using *MATLAB*. The adaptive filter is then simulated to provide attenuate noise signal in waveform. Also, this adaptive filter will be design using Verilog code to implement in the *Spartan-3e* for future work. Lastly, the output is obtained by testing the noise signal to verify the signal is without noise or less noise.

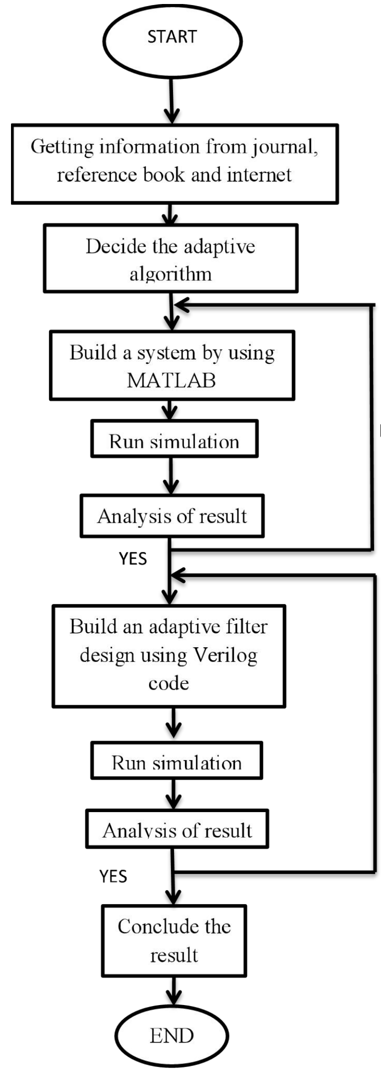

Fig 4: Overall flowchart of the project

V. Results and Discussions

The results of LMS algorithm technique using *Matlab* are shown in Figs. 5, 6, 7, 8, 9 and 10. The parameter of the design of the algorithm were frequency sampling of 8kHz, the step size μ = 0.02 and the adaptive filter of 32 order FIR filter. The sample time was taken 100 samples so that the pattern and any changes for waveform is easier to be observed. In this simulation, the input audio signal *s(k)* is a '*ding.wav*' file format with single data was used as the clean signal. The waveform is extracted using

[s,Fs,nbits]=wavread('ding.wav')

This waveform consists of input sound signal which is the collected the data from the computer system through audio sound *.wav* format file. The waveform for the input audio signal *s(k)* is shown at Fig. 5.

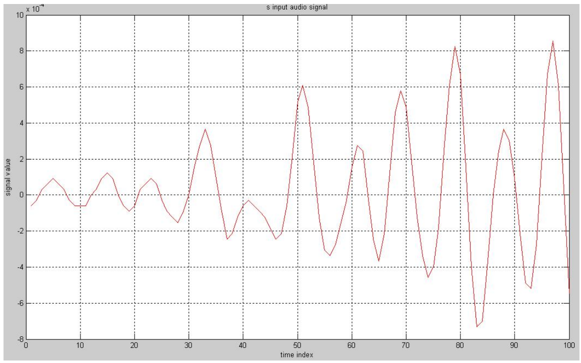

Fig. 5: input audio signal *s(k)*

Fig. 6 shows the random source that will act as the input signal *x(k)*. The input signal *x(k)* is the signal of the contaminating signal – the signal is with noise. To extract the random signal command write:

x = randn(1,100,'single')

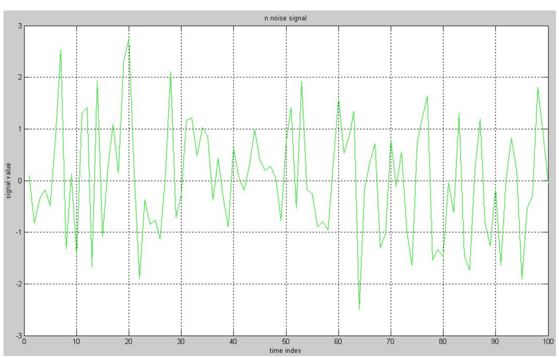

Fig. 6: Input signal *x(k)*

Fig. 7 shows the desired signal *d(k)*. The waveform of *d(k)* is expected to produce high signal than the input audio signal *s(k)* because the signal *d(k)* is an input audio signal *s(k)* corrupted with the noise *n(k)*. The command used to extract the desired signal is given by

d = filter(b,1,x)+ s

where *b* is FIR filter system, *x* is input signal and *s* is input audio signal.

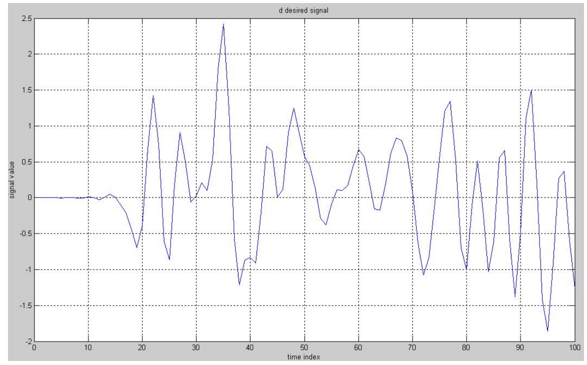

Fig. 7: Desired signal *d(k)*

The output *y(k)* of the adaptive filter is shown in Fig. 8, which will be used to produce a digital filter output *y(k)*, from input signal *x(k)* through the FIR filter. This output waveform represents the resultant from the input signal going through the FIR filter by applying the given command:

[y,e] = filter(ha,x,d)

Where *ha* is FIR LMS adaptive filter, *x* is input signal and *d* is the desired signal.

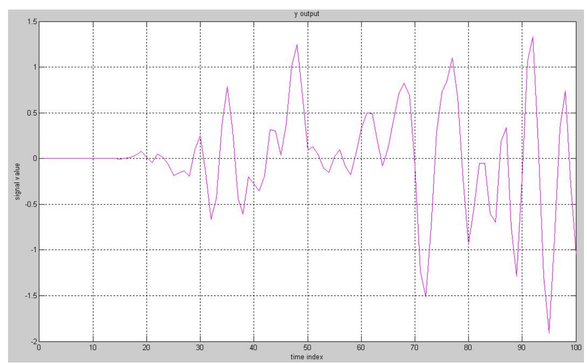

Fig. 8: Adaptive filter output *y(k)*

The extraction of error signal *e(k)* is made by using

[y,e] = filter(ha,x,d)

The result is shown in Fig. 9 describe the waveform from adaptive filter which is derived from the different between desired signal *d(k)* and output signal *y(k)* and cause the error signal to converge (approaching) to a near zero value after many clock cycle.

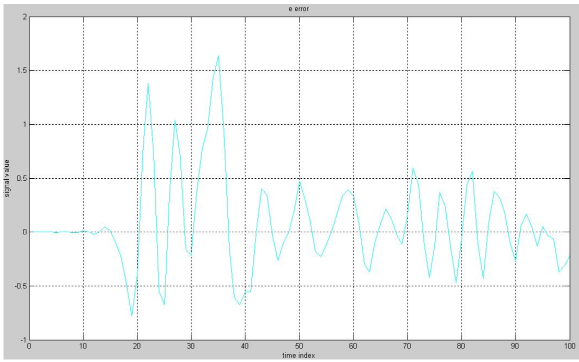

Fig. 9: Error signal *e(k)*

Fig. 10 shows the result of the ANC coefficient. The blue line with point indicates the actual value while the green line with point indicates the estimate value. The 32 point at the graph represent as the number of filter order. Based on the graph, the position of actual point and estimate point is closer each other *i.e.* the ANC design coefficient is correct. The coefficient is used to adjusting the filter coefficient to minimize the error in order to achieve the best desired output. The command used to extract the desired signal is given by

figure,stem([b.',ha.coefficients.']);

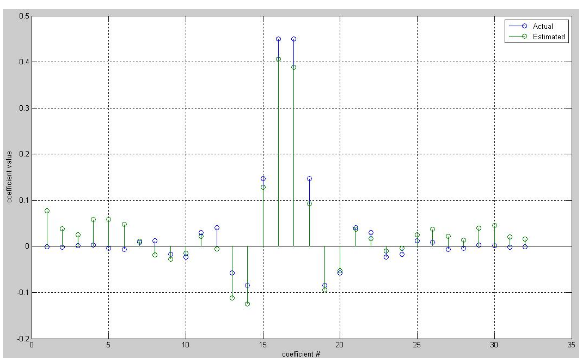

Fig. 10: ANC coefficient

The *Verilog* code developed to obtain the LMS module as shown in Fig. 11. This LMS module was derived from the equation (3) using *Xilink ISE*. The module consists of two multipliers, a substractor and an adder with delays. These delays were to separate the current coefficient with the next coefficient.

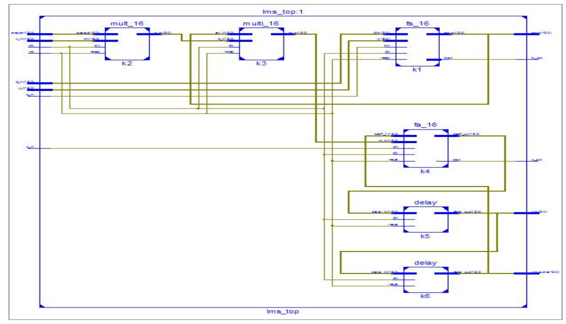

Fig. 11: LMS module using *Xilinx ISE*

The module was then tested using testbench as shown in Fig. 11. From the testbench, the blue arrow indicate the output of update coefficient *w(n+1)* and red arrow indicate the current coefficient *w(n)* that will use in the next clock cycle. Note that the result of the testbench was tested using an integer number. In this work, the coefficient was not tested and in progress.

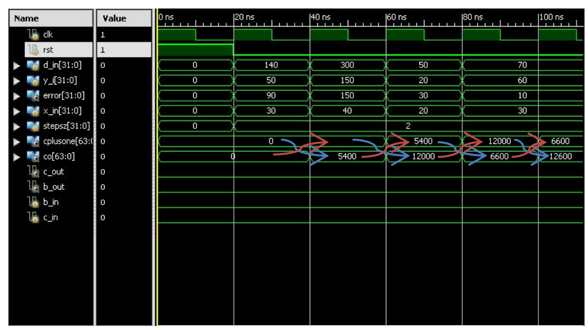

Fig. 11: Testbench result of LMS algorithm for ANC

VI. Conclusion

The simulation of LMS algorithm for active noise cancellation using *Matlab* has been achieved. The simulation result based on design shown in Fig. 5, 6, 7, 8, 9, 10. The system worked as expected and the noise was cancelled out. In addition, the LMS module with *Xilinx ISE* was achieved and tested. However, some works are still in progress.

VII. References

[1] Active noise control, [Online]. Available: http://en.wikipedia.org/wiki/Active_noise_control

[2] Radhika Chinaboina *et. al*, “Adaptive Algorithms for Acoustic Echo Cancellation in Speech Processing”, *International Journal of RRAS,* vol. 7 Issue 1, April 2011

[3] V. R.Vijaykumar *et. al*, “Modified Adaptive Filtering Algorithm for Noise Cancellation in Speech Signals*”, Electronics and Electrical Engineering,* vol. 74, no. 2, 2007, p. 17 – 20.

[4] Allam Mousa, Marwa Qados, Sherin Bader, “Adaptive Noise Cancellation Algorithms Sensitivity to Parameters”*, International Conference on Multimedia Computing and Systems (ICMCS)*, 7 -9 April 2011,p. 1-5.

[5] Lattice Semiconductor Corporation, *LMS Adaptive Filter* [Online].Available: http://www.latticesemi.com/products/intellectualproerty/referencedesigns/lmsadaptivefilter.cfm

[6] Rafid Ahmed Khalil and AwsHazimsaber, “*FPGA* Implementation of Adaptive Noise Canceller”*, Al-Rafidain Engineering,* vol. 17, no.4, August 2009