Final Report On
Active Noise Cancelling

Zhiqiang XIE
ID: 77892769
ShanghaiTech University

Haocong LUO
ID: 33974917
ShanghaiTech University

He WANG
ID: 48853147
ShanghaiTech University

June 10, 2017

Abstract

This is the final report of the project of EE150 Signals and Systems. Through this project we tried to implement an active noise cancelling(ANC) headphone using the FxLMS algorithm, including both system simulation in Simulink and hardware implementation(though not functional due to latency). The report will discuss the principles of the LMS method of ANC, the system implementation with Simulink and the hardware design and implementation. In the end, the reason for the failure of the hardware will be discussed, and some critical problems involved in this project will also be noted.

I. Introduction

The most direct method of cancelling a noise is to block or absord it. For example using a muffler or specially designed chamber which reflects the noise back. However, for acoustic environments, these passive noise cancelling methods performs not very well on low frequency noises. Therefore, an active method of adding a 180-degree-out-of-phase signal of the noise based on the superposition principle.

Since it is an active method, we need to know the noise in advance, or in other words, to estimate the noise.

II. Minimizing the MSE

The direct way of evaluating the result of the estimation is to measure the error between the estimation and the desired signal,

$$e(n) = d(n) - y(n) = \mathbf{w}^T \mathbf{x}(n) \tag{1}$$

, where $\mathbf{w}$ is the vector of coefficients of a discrete FIR filter which is used to produce the estimation from the input reference signal $\mathbf{x}$ by linearly combining the present and past information of the reference.

For adaptive purposes, we need the coefficients $\mathbf{w}$ to be variable with time, so we make it $\mathbf{w}(n)$.

At optimum, the mean square error(MSE) should be zero, denote $J(n)$ to be the MSE,

$$J(n) = E[e^2(n)] \tag{2}$$

, expand the square we have

$$J(n) = E[d^2(n)] - 2\mathbf{p}^T \mathbf{w}(n) + \mathbf{w}^T(n)\mathbf{R}\mathbf{w}(n) \tag{3}$$

, where $\mathbf{p} = E[d(n)\mathbf{x}(n)]$ is the cross-corellation between the desired signal and the reference, $\mathbf{R} = E[\mathbf{x}(n)\mathbf{x}^T(n)]$ is the auto-corellation of the reference.

It can be seen that $J(n)$ is actually a quadratic form of $\mathbf{w}(n)$. Therefore there is a global minium at

$$\frac{dJ(n)}{d\mathbf{w}(n)} = -2\mathbf{p}^T + 2\mathbf{w}^T(n)\mathbf{R} = 0 \tag{4}$$

$$\mathbf{w}(n) = \mathbf{R}^{-1}\mathbf{p} \tag{5}$$

Now at the global minium we inspect the cross-corellation between the error and the reference,

$$\begin{aligned} E[e(n)\mathbf{x}(n)] &= \mathbf{R}^{-1}\mathbf{p} \\ &= E[(d(n) - \mathbf{w}^T(n)\mathbf{x}(n))\mathbf{x}(n)] \\ &= \mathbf{p} - \mathbf{R}\mathbf{w}(n) \\ &= \mathbf{p} - \mathbf{R}\mathbf{R}^{-1}\mathbf{p} \\ &= 0 \end{aligned} \tag{6}$$

, which shows that the residual of the input signal has no corellation with the reference, which is the noise, impling perfect elimination of the noise.

III. The LMS Algorithm

The direct approach to solve for $\mathbf{w}(n)$ involves calculating the inverse of the auto-corellation matrix which has high computation cost. In addition, the characteristics of the noise and the signal is usually unknown. There should be an iterative method to gradually approximate the optimum.

The LMS approach use the instantaneous error to estimate the MSE,

$$\widehat{J}(n) = e^2(n) \tag{7}$$

, take the derivative

$$\begin{aligned} \frac{d\widehat{J}(n)}{d\mathbf{w}(n)} &= e^2(n) \\ &= 2e(n)\frac{de(n)}{d\mathbf{w}(n)} \\ &= -2e(n)\mathbf{x}(n) \end{aligned} \tag{8}$$

, then we plug it into the ordinary steepest descent method

$$\begin{aligned} \mathbf{w}(n+1) &= \mathbf{w}(n) - \frac{\mu}{2}\frac{\widehat{J}(n)}{d\mathbf{w}(n)} \\ &= \mathbf{w}(n) + \mu e(n)\mathbf{x}(n) \end{aligned} \tag{9}$$

.

Therefore, we can use equation(8) to update the coefficients of the FIR filter.

However, descending with a fixed step size is usually considered too naive because a large step size may not be able to find the global minimum while a very small step size takes too long to converge. In application the simple LMS algorithm is modified to have variable step size, called normalized-LMS algorithm, revealing

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu(n)e(n)\mathbf{x}(n) \tag{10}$$

, where $\mu(n)$ is the variable step size. The idea is that we want the step size to vary with the reference signal, so we can normalize the step size with the estimated power of the reference

$$\mu(n) = \frac{\alpha}{\mathbf{x}^T(n)\mathbf{x}(n)} \tag{11}$$

, where α is called the normalized step size.

The equation for updating the FIR coefficients can be modified into

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \alpha\frac{\mathbf{x}(n)e(n)}{\mathbf{x}^T(n)\mathbf{x}(n) + \sigma} \tag{12}$$

, the σ in the denominator is a small regularization term to prevent the pole.

IV. System Design

The overall system simulation in Simulink is shown in Figure 1 below

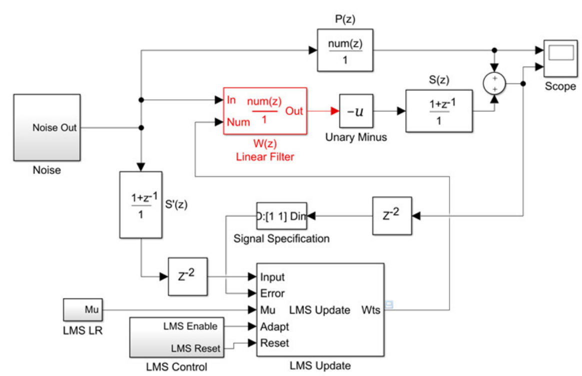

Figure 1: *The Feedforward ANC System*

The system uses the feedforward ANC method to cancel the noise, where 2 microphones are used respectively to collect the reference signal and the error signal, and the effect of the secondary path has been taken into consideration.

i. The Secondary Path

In real life, the acoustic signal will travel through a long path before it is catched by the digital system. The secondary path consists of the loud speaker system which involves D/A conversion and amplifing, the acoustic paths among the speaker, the summing junction of the anti-noise and the signal and its path to the error microphone and the digitizing process of thre error microphone.

In the following figure 2 we use $Sp(z)$ to denote the propagation procress from the loud speaker to the summinng junction, $Pp(z)$ to denote the propagation procress of the original signal travelling to the summing junction and $R(z)$ to denote the digitizing procressing.

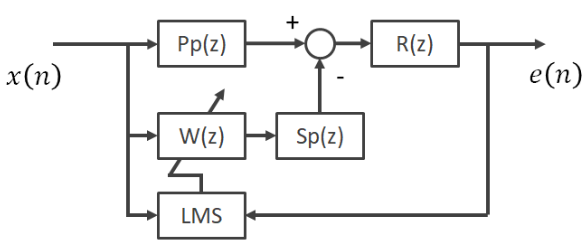

Figure 2: *The Secondary Path*

ii. The FxLMS Algorithm

Due to the exsistence of the secondary path, the output of the adaptive system is not aligned with the reference, which increases the instability or risk of no-convergence of the system.

Use $S(z)$ to denote the secondary path, the error can be written as

$$\begin{aligned} e(n) &= d(n) - y(n) \\ &= d(n) - \mathbf{w}^T \mathbf{x'}(n) \\ &= d(n) - \mathbf{w}^T (\mathbf{S}^T \mathbf{x}(n)) \end{aligned} \tag{13}$$

It can be seen that the original normalized-LMS algorithm should be modified to take account for the effect of the secondary path. As can be derived from Equation(9) the solution is to filter the reference input with an system identical with the secondary path.

The modified system is shown below in figure 3.

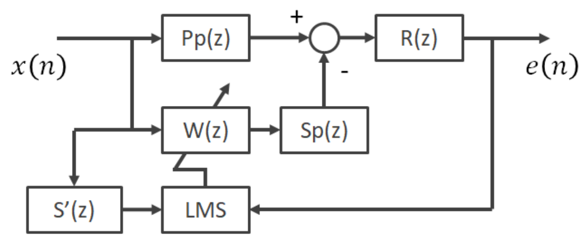

Figure 3: *The FxLMS Algorithm*

V. Simulation

We first simulate our system in Simulink to verify its functionality. Here we use a dummy secondary path with only one constant coefficient 1.

We recorded the noise generated by the huge fan in an SIST room as the noise. The frequency spectrum of the noise is shown below in figure 4.

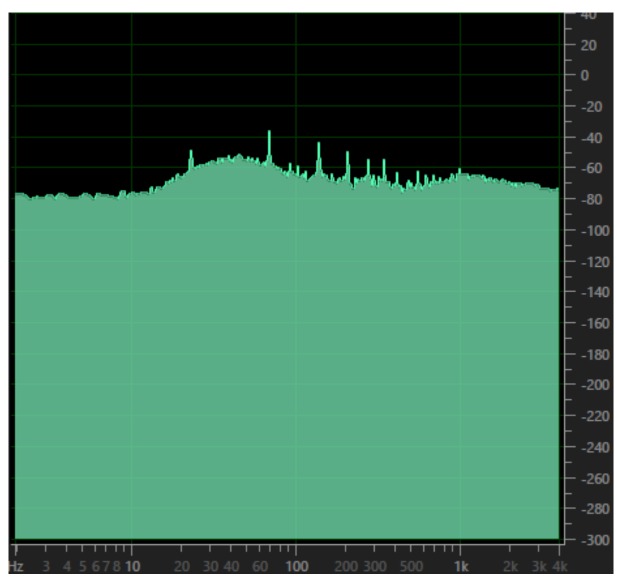

Figure 4: *The Spectrum of the Fan Noise*

It can be seen that the most significant part of the noise are all located the low frequencies(below 2000 Hz)

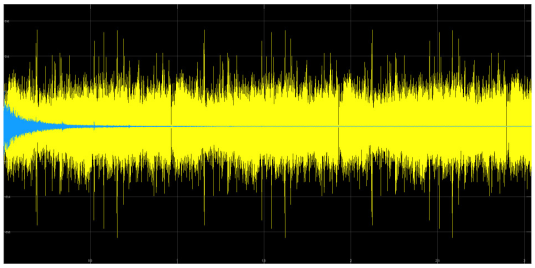

Figure 5: *The Simulation Result*

In the above figure, the blue part is the error, and the yellow part is the original noise.

The simulation shows very good noise elimination effect(almost 100%) and converges quickly(in about 2s). Impling that the system design is correct and works well.

VI. Hardware Implementation

At first we decided to use a DSP or FPGA to build an ANC headphone, but we quickliy found that the DSP evaluation board within our budget(Ti C55x series) only support a maximum speed of 400kbps through the I^2C port, which is too slow for our audio system. Thus, we need to use the serial port dedicated to audio applications like the I^2S port.

However, there is no commercial audio codec board that does the A/D conversion fast and then send the data to the I^2S available, too. Due to the limit of available time at school it is not feasible to design our own PCB to get the audio codecs working for our needs, we finally decided to abandon the DSP/FPGA plan and use an ordinary PC instead.

We use the following materials to build our headphone

- Sennheiser Headphone HD201
- Panasonic Electrode Microphone WM-60A
- Mic Pre-amplifier MAX9812L
- Simple USB Sound Cards

The HD201 headphone is cheap and has a large chamber for us to put things in, and the Panasonic Microphone WM-60A are used by the industry to measure the frequency response of acoustic equipments thus it has good precision and board and uniform frequency response.

The feedforward ANC needs 2 microphones, one as the reference and the other for collectiong the error signal. The configuration of the 2 microphones are shown in figure 4 below.

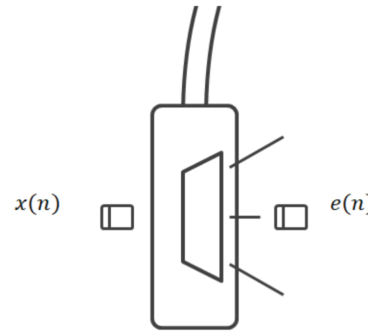

Figure 6: *The Microphone Configuration*

VII. Secondary Path Estimate

We need to obtain a system identical with the secondary path as decribed in the previous section, so we must perform some estimation on the secondary path. There are two ways of modeling the secondary path, the on-line method and the off-line method. Since in a headphone, the secondary path is relatively stable and not changing, an off-line modeling method is selected in this project.

We consider the secondary path as a discrete FIR filter and uses an LMS filter to obtain its coefficients. The system for modeling the secondary path is shown below in figure 4.

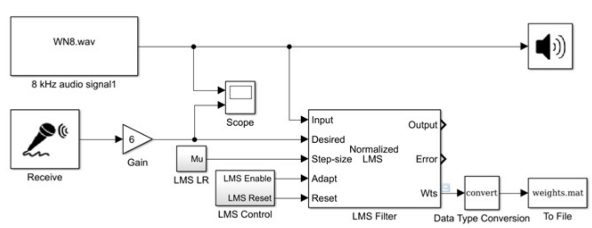

Figure 7: *Secondary Path Estimate*

We feed the system with different signals and inspect the convergence of the Wts output.

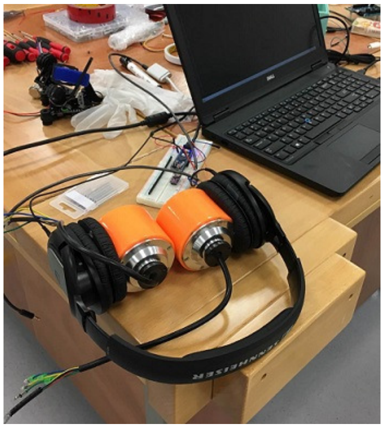

Figure 8: *Secondary Path Estimating Procress*

i. Sine Wave

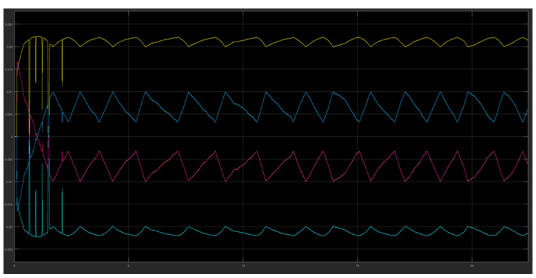

Figure 9: *S(z) Coefficients Estimate Using Sine Wave*

ii. Chirp

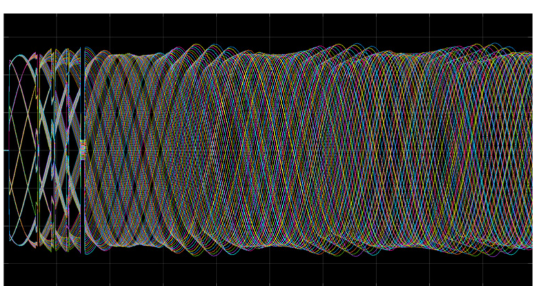

Figure 10: *S(z) Coefficients Estimate Using Chirp*

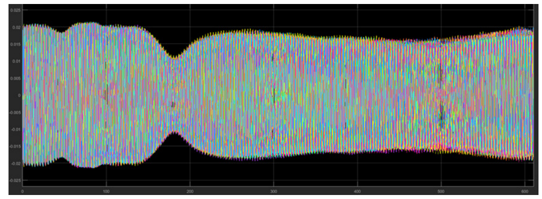

Figure 11: *S(z) Coefficients Estimate Using Another Chirp*

iii. Band-limited White Noise

Notice that although the LMS ANC cannot eliminate white noise in application, but we can still use the white noise to estiamte the secondary path because the two systems are different. Using a white noise in secondary path estimation will get the theoretically best result because it covers the brandest frequencies.

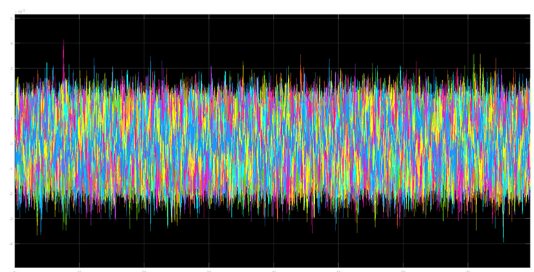

Figure 12: *S(z) Coefficients Estimate Using Band-limited White Noise*

Here we can see the weired fact that the estimated secondary paths are quite different from each other. This contradicts with the fact that the secondary path inside the headphone should be stable.

A possible explanation for this phenomena is that the latency in the I/O procress disturbs the stability of the system. An evidence for this is that when we test our headphone with the dummy secondary path, its reponse to the outer noise has a large ear-detectable delay(about 1s to 2s).

VIII. Dealing with Latency

We conclude the reason for the failure of hardware implementation as the effect of I/O latency.

i. Latency Analysis

In the system, their all together three audio I/O sources. One audio output is the loud speaker playing the anti-noise, the other two are the microphones obtaining the referencec and error signals.

We obtain the following figure explaining the audio system diagram in Matlab and Simulink from the documents, the output stream is alike.

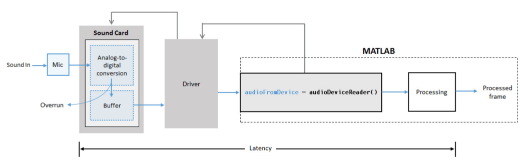

Figure 13: *The Input Audio Stream*

From the figure it can be known that there are 3 stages influencing the overall system latency, the sound card, the driver and the Matlab itself.

We have already optimized the Matlab part by decreasing the sampling rate to 8000Hz, which is considered twice the Nyquist sampling rate in the system(We have found that the most siginificant parts of our noise are below 2000Hz, and then consider sparing some space for signal reconstruction we choose $8000Hz = 2000Hz * 2 * 2$ as our sampling rate).

The sound card, which is a highly-intergrated device, has little to optimize. The only thing left that can be optimized is the driver. Actually, ordinary Windows PCs use the Microsoft DirectSound driver, which is known to be very slow because it relies on the OS and the CPU to procress the audio signals. It may not be an issue for every day audio applications but for a real-time application like this ANC project, it will not work well.

ii. Optimize by ASIO

In order to optimize the latency in the driver layer, we tried to use the ASIO driver instead of the default DirectSound driver. ASIO is an audio driver made by professional audio company Steinberg, which can siginificantly reduce the latency when recording and playing audio. We tested the latency performance using ASIO and DirectSound using the built-in Matlab example file, the results are shown below.

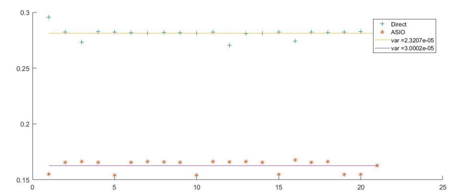

Figure 14: *Latency Test Between ASIO and DirectSound*

It can be seen that the overall lantency is greatly reduced by almost half, but the absolute value is still higher than 100ms and the variance is not very good.

Another issue within our headphone is that not every audio device supports the ASIO driver, like our additional sound card for the 2 microphones. The ASIO driver only improves the performance of the audio playback, the rest two sources of high latency is still present.

In our system with a sample rate of $8000Hz$, a lantency of 200ms means the loss of 1600 samples(we are always using sample-based method instead of frame-based method for the least possible latency). Due to the limit of the computational power and Simulink itself, we set the discrete FIR filter length to 75, which is much less than the loss of 1600. It means that the data we get and the signal we want to estimate have almost no corellation, resulting in the failure of hardware implementation. The latency measured in the figure is only concerning about the electronic propagation delay, not including other physical delay. For a stable and small latency, we may still consider it as an LTI system which our system can tolerate. But for a huge delay of more than 1000 samples, due to the loss of corellation the adaptive filtering will not work well.

IX. Analysis and Conclusion

In the end of the project, we may say that the hardware implementation is not functional although the software simulation shows positive results. The essential reason for the failure of the hardware is that we are trying to do a real-time task on a time-sharing operating system.

In a time-sharing operating system, the scheduled time for each task is not guaranteed. Therefore, the actual latency is varying with the current condition of the operating system. In the previous test for latency, we made the test in a clean and rebooted OS and still has a considerable variance. In the application there are three audio I/O device working at the same time and it is difficult to know and control how the OS scheduled these three tasks, revealing much more unpredictable and variable latency.

However, assume we have a stable but large latency, there is still possibility that we can make the system stable by adding extra FIR filters taking the latency into account.

In real life, the most cost-effective method is to use real-time hardware like the DSP or the FPGA, with clock-locked microphones and speakers, just like the commercial ANC headphones do.

For improvements on the algorithm, besides the variations of LMS algorithm, the adaptive genetic algorithm(AGA) can also be applied to replace the LMS algorithm to directly search for the global minimum, avoiding the possible local minimum brought by the secaondary path.

X. Division of Labour

Besides the group discussion and other collaborated work, everyone's major work are

- Zhiqiang XIE: Software Simulation and Analysis
- Haocong LUO: Hardware Implementation and Report Writing
- He WANG: Hardware Implementation and Presentation Work

References

[1] B. Widrow, et al. Adaptive Noise Cancelling: Principles and Applications *Proc. IEEE*, vol. 63, pp.1692-1716, Dec. 1975.

[2] Luigi Piroddi. Active Noise Control(PhD Course). *http://home.deib.polimi.it/piroddi/anc.html*