NOISE CANCELLATION USING LEAST MEAN SQUARES ADAPTIVE FILTER

Jonathan Cedarleaf, Steve Philbert, Arvind Ramanathan

University of Rochester, Department of Electrical and Computer Engineering

ABSTRACT

For this project, the field of adaptive filtering was explored, as it relates to audio signal processing. There are many areas where adaptive filtering is useful, including echocardiograms, acoustic echo cancellation, feedback suppression, noise cancellation, signal prediction, and many others. The process explored herein uses the Least Mean Squares method, or LMS, to remove unwanted noise from an input signal. This was implemented and demonstrated successfully using an Open Multimedia Applications Platform (OMAP).

***Index Terms*—** Adaptive Filtering, LMS, Audio Signal Processing, MATLAB, DSK, OMAP

1. INTRODUCTION

In a basic sense, adaptive filtering is the creation and use of a filter, the parameters of which are changed based on an optimization algorithm. An example of adaptive filter usage could be removal of noise from an audio signal. If there is some irregular undesired noise in the output signal then adaptive filtering can be used to actively track the noise in the signal and remove it. This unwanted noise is often a wide-spectrum noise, compared with the source signal (such as white noise with a sinusoidal wave), so traditional static filters would not be adequate for noise removal.

2. LMS ALGORITHM

Least Mean Squares (LMS) is one of the most common algorithms used to implement adaptive filtering. [2] LMS changes its filter coefficients based on the desired signal by finding the least mean square of the error signal, *e[n]*, which is estimated by taking the difference between the desired signal, *d[n]*, and the filtered signal, *y[n]*. The desired signal reference can be the input signal without noise, or it can be a delayed version of the noisy input signal. The delay is required to remove the correlation between the two sources. In the latter case, the algorithm finds the common noise in both the signal and the filter output, and estimates the error between the two. Figure 2-1 shows two stages of the LMS filter. In practice a higher filter order can be used for better results.

Difference Equation For LMS [1]:

$$\boldsymbol{w}[n+1] = \boldsymbol{w}[n] - \mu \nabla_w J(\boldsymbol{w}[n]) \qquad (1.1)$$

$$\nabla_{\mathrm{w}} J(w[n]) = -2\boldsymbol{p}_{dx} + 2\boldsymbol{R}_x \boldsymbol{w}[n] \qquad (1.2)$$

$\boldsymbol{R}_x$ and $\boldsymbol{p}_{dx}$ are defined by

$$\boldsymbol{R}_x \simeq \boldsymbol{x}[n]\,\boldsymbol{x}^T[n] \; and \; \boldsymbol{p}_{dx} \simeq d[n]\boldsymbol{x}[n] \quad (1.3)$$

Substituting the above in (1.2) and combining (1.1) and (1.2), we obtain

$$\begin{aligned} \boldsymbol{w}[n+1] &= \boldsymbol{w}[n] + 2\mu\boldsymbol{x}[n](d[n](-\boldsymbol{x}^T[n]\boldsymbol{w}[n]) \\ &= \boldsymbol{w}[n] + 2\mu\boldsymbol{x}[n](d[n](-\boldsymbol{w}^T[n]\boldsymbol{x}[n]) \\ &= \boldsymbol{w}[n] + 2\mu e[n]\boldsymbol{x}[n] \end{aligned}$$

where:

$y[n] = \boldsymbol{w}^T[n]\boldsymbol{x}[n]$	*Filter output*
$e[n] = d[n] - y[n]$	*Error*
$\boldsymbol{w}[n] = (w_0 w_1 \dots w_{M-1})^T$	*Coefficients*
$\boldsymbol{x}[n] = (\boldsymbol{x}[n]\boldsymbol{x}[n-1] \dots \boldsymbol{x}[n-M+1])^T$	*Input data*

The *w[n]* component is the filter coefficients for the adaptive filter. Equation (1.1) gives the future coefficient for the filter by taking into account μ, the factor which determines the filter convergence rate. The coefficients are changed based on the error signal, which is a difference between the input signal and the desired signal.

The desired signal is a signal which does not have noise in it. When implementing the system with headphones, this desired signal will be the audio input (music) with no noise. The estimation of the error signal is the input signal with the ambient noise, compared with the desired signal. The difference between them (the error signal) is fed back to the estimation of the filter coefficient. By this process the filter weights are modified in real time based on the error signal.

The convergence coefficient, μ, is used to control the rate at which the filter converges, a larger value will cause the filter coefficients to change at a faster rate, and vice-versa. This coefficient is multiplied with the error signal, updating the next filter weight(s).

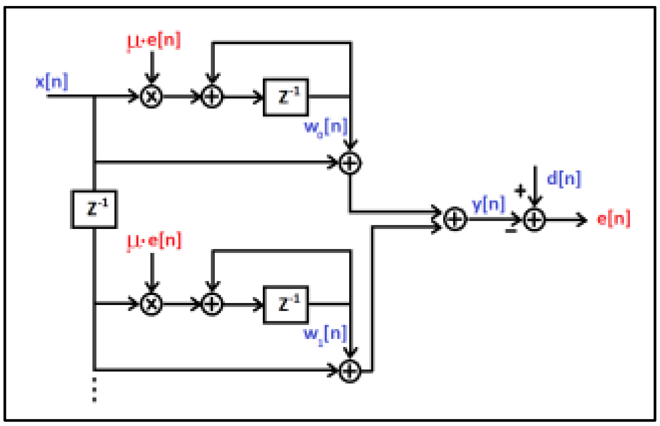

Figure 2-1 Second order LMS filter

3. IMPLEMENTATION

The desire is the exploration in the area of adaptive filtering with relation to audio signal processing, specifically the topic of active noise cancellation. As described previously, the method of choice was the LMS adaptive filter approach. The first approach was a simulation in MATLAB and the second approach was implementation on hardware in real time.

3.1 MATLAB Implementation

A simulation of the LMS algorithm was performed in MATLAB to become familiar with the algorithm parameters and its operation.

The source signal was a 1,000 Hz sine wave and the noise signal was randomly generated signal with amplitude half of the sine wave. The input to the filter was the summation of the sine wave and noise signal. The desired signal was a delayed version of the input signal. Figure 3.1-1 shows the input and output signals of the filter. The filter output shows the signal converging to the sine wave shape and the error output shows the converging to the noise shape.

The parameters used in the MATLAB implementation were a delay of 30 samples, convergence coefficient, μ, of 0.0001 and filter order of 26. With a sample rate of 11025 Hz, this has a latency of 2.4 milliseconds.

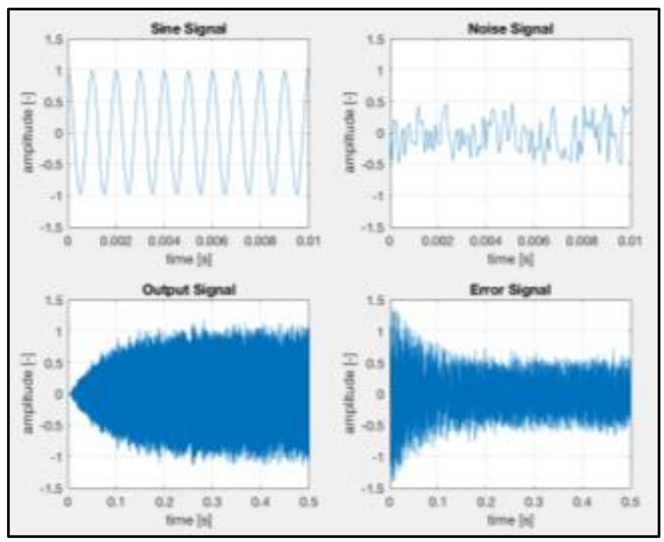

Figure 3.1-1 MATLAB simulation data

In Figure 3.1-2, the first plot shows the noisy input signal. The second plot shows the filtered output; notice the desired sine signal isolated from the noise. There is still noise present, but the noise energy is greatly reduced. The final plot shows the error signal, which describes how the filter removes much of the noise over time, without removing the desired sine wave. The sine wave is not in the error signal after 0.2 seconds.

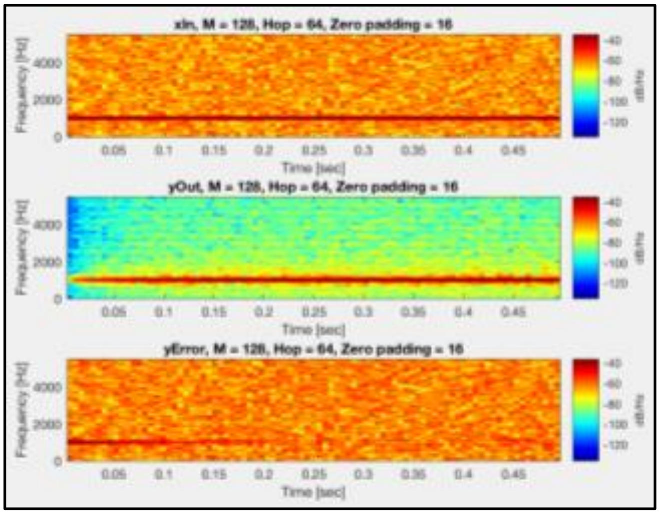

Figure 3.1-2 MATLAB Spectrograms

3.2 Hardware Implementation

With a clear understanding of the LMS algorithm in MATLAB, a real-time implementation was created using the OMAP-L138 board to apply the LMS filter, and the DSK 6713 DSP Starter Kit was used as a microphone preamplifier/processor. The filter was implemented on the OMAP-L138 using the same approach as that used in MATLAB, with the exception of the noise input and signal input. While the experiment in MATLAB used a delayed form of the input signal as *d[n]*, the desired signal on the OMAP-L138 was the pure music signal (with no noise). The input to the filter was music played from a cell phone (direct line-in) combined with the room noise picked up by a microphone.

In order to complete the processing required for the LMS adaptive filter, the sampling frequency on the OMAP-L138 had to be lowered from the standard 44,100 Hz to 16,000 Hz. This change may have a tendency to cause some undesired effects to music played through the system, but it enabled the system to process a filter of order 65, and as mentioned previously, a higher filter order results in better noise cancellation. A lower sampling rate means more processing time in the interrupt, which is where the filter implementation is computed. This filter order resulted in a latency time of about 4 milliseconds, which is adequately short for the purposes of this system. Also, in practice a convergence coefficient, μ, of 0.01 was used. This was experimentally determined to provide the most optimal results.

In an actual usage scenario, the environment in which the system is used would be audibly noisy, without the active filtering in place. The headphones used in this project had a significant amount of passive noise cancellation, so in order to accurately simulate a noisy environment, the noise signal from the microphone was added to the music played through the headphones. Then this music/noise combination signal was used as the filter input.

Figure 3.2-1 Implementation used

To test the robustness of the hardware implementation, the system was tested with various noisy input signals. Such tests were a sinusoidal input combined with a sinusoid at a different frequency, a music signal combined with white noise, and a music signal combined with a sinusoidal signal. In all cases, the system was able to accurately isolate the desired signal.

4. CONCLUSION

The real-time implementation of the LMS adaptive filter was able to successfully remove much of the noise added to the input signal. When implementing a microphone for measuring room noise, the system removed much of the constant energy, wideband noise elements interfering with the music played through the headphones.

The system implemented here is best suited for the removal of constant energy, wide bandwidth noise (that has little to no abrupt changes), such as noise from a plane or car engine, noise from air vents, or any constant “room noise.”

5. FUTURE WORK

The limitations of this system come when there is some narrow band noise, such as people talking, sharp sounds like clapping, and other narrow band non-periodic sounds. While this system was able to demonstrate noise cancellation for constant noise, a more practical design would include a wider range of capabilities.

The relationship between sampling frequency and filter order can also be further examined; by increasing the overall processing speed, the sampling rate could be increased which increases the Nyquist limit, and/or the number of filter coefficients could be implemented which would provide a better filter response,

The methods in this paper examined the LMS algorithm, other variations of adaptive filters can be implemented such as NLMS, RLS, LPC, etc.

6. REFERENCES

[1] A. Poularikas, Adaptive filtering, 1st ed.., CRC Press, January 9, 2015.

[2] P. Diniz, Adaptive filtering, 1st ed. New York (NY), Springer, 2013.

[3] B. Farhang-Boroujeny, Adaptive Filters: Theory and Applications, 2nd Edition, 1st ed. John Wiley & Sons, 2013.

[4] "Understanding the Echo *‘Phenomenon’* Causes and Solutions", Adaptivedigital.com, 2017. [Online]. Available: http://www.adaptivedigital.com/product/echo_cancel/echo_explain.htm. [Accessed: 04- May- 2017].

[5] "Overview of Adaptive Filters and Applications - MATLAB & Simulink - MathWorks United Kingdom", Mathworks.com, 2017. [Online]. Available: https://www.mathworks.com/help/dsp/ug/overview-of-adaptive-filters-and-applications_f1-5512.html. [Accessed: 04- May- 2017].