HIGH SCHOOL EDITION
Journal of Student Research

Volume 13 Issue 4 (2024)

FFT Based Adaptive Noise Cancellation

Justin Liu

Northwood High School, USA

ABSTRACT

This paper presents how to cancel the audio noise adaptively with using FFT (Fast Fourier Transform). It covers theoretical analysis, MATLAB simulation, performance evaluation (with different step size) and trade-off. It proves FFT can be used to cancel the audio noise adaptively.

Introduction

Speech is often corrupted by background noise. Thus noise reduction is required to attenuate the unwanted components while preserving the speech signal. If the characteristics of the signal and the noise are a priori knowledge, a stationary filter can be designed to optimize certain performance criterion. Unfortunately, the properties of the noise often vary with time. It is therefore necessary to use adaptive filters. Adaptive filters presented in his paper adjust their parameters to minimize the noise energy. Such noise reduction techniques require no prior knowledge of the noise. Early work in this area began with the development of the Wiener-Hopf integral equation whose solution gave an optimal estimate which minimizes the Root Mean Square (RMS) error. The Wiener-Hopf equation is analog in nature and tended to be difficult to implement, which limited its practical application. An important breakthrough was made in 1960 by Widrow and Hoff who developed the Least Mean Square (LMS) algorithm for use with finite-length impulse response (FIR) digital adaptive filters. The adaptive filters adapt the filter coefficients so as to minimize the output noise power. Due to the simplicity of the LMS algorithm, the techniques were popularly used in applications such as pattern recognitions; signal detections, adaptive beam-forming and so on. In this paper it will explore a specific application of adaptive filtering, i.e., adaptive noise cancellation.

Architecture Description

The configuration involved is shown below

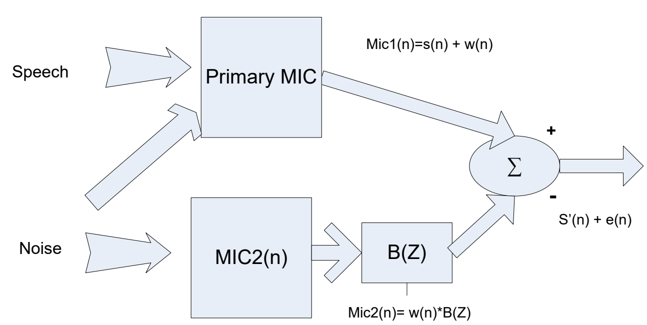

Figure 1. Block diagram of Adaptive Noise Canceller

Where B(Z) is the adaptive filter that models the acoustic path from the noise source to the primary microphone. The primary microphone is picking up speech and background noise from a noise source. Background noise is monitored by a secondary microphone close to the noise source and sufficiently separated from the primary microphone. The idea is to acquire the signal from the noise source and process it by an FIR filter and produce w'(n) and then subtract it from the primary microphone signal (s(n)+w(n)). If w'(n) is equal to w(n), then the noise is perfectly cancelled and the output is clean speech. Because of random acoustic (reflection, delay, etc) w(n) cannot be cancelled perfectly, i.e. w'(n) approximates w(n). The FIR filter, B[z] models the reflections and delays in the acoustic path between the noise source and the primary microphone. Since the acoustic path is time varying because of movement in the room, the filter needs to be continuously evaluated, or, an adaptive filter is required. The coefficients of B(z) can be estimated using a frequency-domain adaptive algorithm, which is presented in more details in the following section

Theoretical Analysis

The FFT-based adaptive noise canceller has two main advantages:

1) FFT replaces time domain convolution with frequency domain multiplication. Although complicated to implement, this technique is computationally efficient, especially for filter order N>16.

2) Block implementation of the FIR filter allows parallel processing of data, resulting in a gain of computational speed. In addition, since a block of data is used to estimate the gradient (instead of a single sample), the filter coefficient update is more precise.

The block diagram of the frequency domain LMS adaptive filter is depicted below

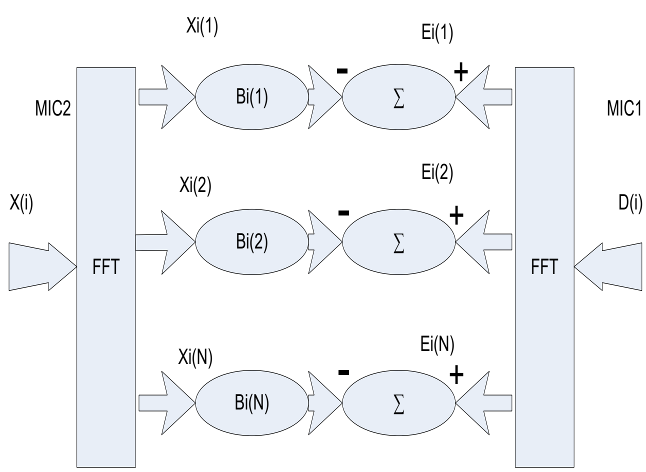

Figure 1. Block diagram of LMS adaptive filter

Where

$$D_i = [D_i(1) D_i(2) \cdots D_i(N)]^T$$
$$X_i = [X_i(1) X_i(2) \cdots X_i(N)]^T$$
$$B_i = [B_i(1) B_i(2) \cdots B_i(N)]^T$$

$$E_i = \left[E_i(1)E_i(2)\cdots E_i(N)\right]^T$$

Note that N is the number of data in each frame, and i is the frame index.

Derivation of The Filter Update Equation

The error between filter output and the desired signal is

$$E_i = D_i - X_i * B_i \qquad \text{Equation 1}$$

We wish to minimize the mean square error (MSE)

$$MSE = \varepsilon_i = E[E_i E_i^*] \qquad \text{Equation 2}$$

Where E[] denotes the expectation operation. Using the method of steepest descent, the filter coefficients are updated by

$$B_{i+1} = B_i - \mu\nabla(\varepsilon_i) \qquad \text{Equation 3}$$

Where $\nabla(\varepsilon_i)$ is the gradient of ε_i and is evaluated as

$$\nabla(\varepsilon_i) = \partial\varepsilon_i / \partial B_i = [\partial\varepsilon_{i1} / \partial B_i(1)\partial\varepsilon_i / \partial B_i(2)\ldots\partial\varepsilon_i / \partial B_i(N)]^T \qquad \text{Equation 4}$$

Due to the expectation operation, ε_i is not instantaneously available. Therefore it will be approximated each iteration by the instantaneous-square-error (ISE) function

$$ISE_i = E_i E_i * \qquad \text{Equation 5}$$

The estimated gradient is:

$$\nabla(\varepsilon_i) \cong \partial(E_i E_i^*) / \partial B_i \qquad \text{Equation 6}$$

E_i is a complex vector and can be written as

$$\begin{aligned} E_i &= E_{Ri} + jE_{Ii} \\ \Rightarrow &= D_{Ri} + jD_{Ii} - (X_{Ri} + jX_{Ii})(B_{Ri} + jB_{Ii}) \\ \Rightarrow &= D_{Ri} - X_{Ri}B_{Ri} + X_{Ii}B_{Ii} + j(D_{Ii} - X_{Ri}B_{Ii} - X_{Ii}B_{Ri}) \end{aligned}$$

Equation 7

Therefore

$$E_{Ri} = D_{Ri} - X_{Ri}B_{Ri} + X_{Ii}B_{Ii}$$

Equation 8

$$E_{Ii} = D_{Ii} - X_{Ri}B_{Ii} - X_{Ii}B_{Ri}$$

Equation 9

Equation can be expressed with respect to the real and imaginary parts

$$\partial(E_iE_i^*)/\partial B_i = \partial(E_iE_i^*)/\partial B_{Ri} + j\partial(E_iE_i^*)/\partial B_{Ii}$$

Equation 10

Using the property of partial derivative

$$\partial(E_iE_i^*)/\partial B_i = \partial(E_i^*)/\partial B_i + j\partial(E_i)/\partial B_i$$

Equation 11

Equation 10 turns into

$$\partial(E_iE_i^*)/\partial B_i = E_i\partial(E_i^*)/\partial B_{Ri} + E^*\partial(E_i)/\partial B_{Ri} + jE_i\partial(E_i^*)/\partial B_{Ii} + jE_i^*\partial(E_i)/\partial B_{Ii}$$

Equation 12

Solving each of the above partial derivatives gives

$$\partial(E_i^*)/\partial B_{Ri} = -X_i^* \quad \textit{Equation 13}$$

$$\partial(E_i)/\partial B_{Ri} = -X_i \quad \textit{Equation 14}$$

$$\partial(E_i^*)/\partial B_{Ii} = jX_i^* \quad \textit{Equation 15}$$

$$\partial(E_i)/\partial B_{Ii} = -jX_i \quad \textit{Equation 16}$$

Substituting Equation 13 to Equation 16 into Equation 12

$$\begin{aligned}\partial(E_iE_i^*)/\partial B_i &= E_i(-X_i^*) + E_i^*(-X_i) + jE_i(jX_i^*) + jE_i^*(-jX_i)\\ &= -E_iX_i^* - (E_iX_i^*) - E_iX_i^* + (E_iX_i^*)\\ &= -2E_iX_i^*\end{aligned} \quad \textit{Equation 17}$$

Substituting the above result into Equation 3 gives the filter update equation:

$$B_{i+1} = B_i + 2\mu E_iX_i^* \quad \textit{Equation 18}$$

Step Size u Selection

For the block LMS algorithm to converge, the step size u should satisfy the following condition:

$$0 < \mu < 2/\lambda_{\max}$$

Where λ_{max} is the largest eigenvalue of the correlation matrix of the input signal vector X_i. Proper selection of μ is important. Large u means fast convergence and small μ implies slow convergence. On the other hand, large μ will also give rise to large mis-adjustment. Therefore in selecting μ , there is a trade-off between adaptation speed and mis-adjustment.

Simulation Results

SNR Performance

The adaptive noise canceller is evaluated using both objective and subjective performance measures. The objective measure is the signal to noise ratio (SNR). The subjective measure is a rating of 1 to 5 with 5 as best
The SNR is defined as

$$SNR_{db} = 10\log_{10}\left[\frac{\sum_{n=1}^{Nsample} S^2(n)}{\sum_{n=1}^{Nsample} e(n)^2}\right]$$

Where Nsample is the total number of samples in the sound file, s(n) is the clean signal, e(n) is error signal. The simulation results are tabulated as follows.

Table 1. Simulation results for music audio file

N	**SNR delta (after-before)**	**Subjective Rating**	**Best μ**	**File**
4	21.16	5	*0.0005*	*music*
16	19.03	5	*0.001*	*music*
64	16.3	4	*0.0001*	*music*
128	15.1	4	*0.000058*	*music*
256	14.9	4	*000058*	*music*
512	13.19	4	*0.000078*	*music*

The second columns of the above tables show the SNR improvement. A positive SNR improvement value indicates that the noise in the resultant signal is reduced. Several observations can be made from the table.
On a whole, the signal with music noise is better processed when smaller step size is picked. Both the SNR improvement and subjective rating are higher for smaller FFT size.
The best u that produces the best SNR improvement decreases as N increases.

Convergence Curves for N= 4, 64, 128 and 512 with Best u

HIGH SCHOOL EDITION
Journal of Student Research

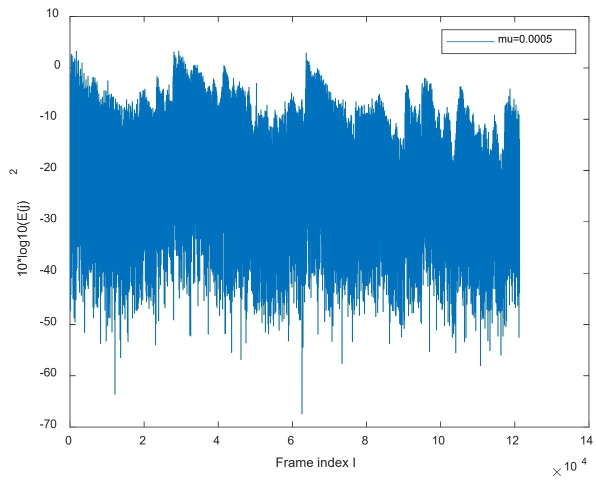

Figure 2. Convergence curve for N=4

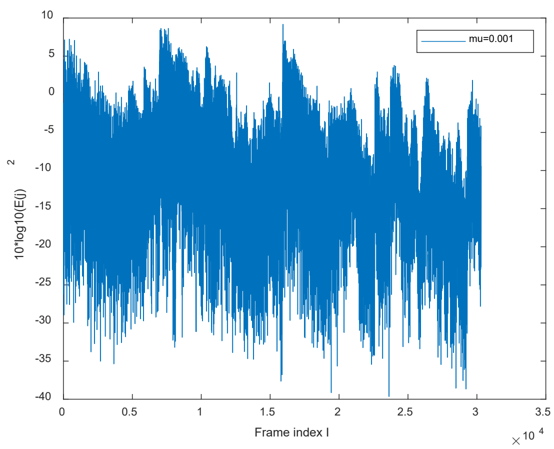

Figure 3. Convergence curve for N=16

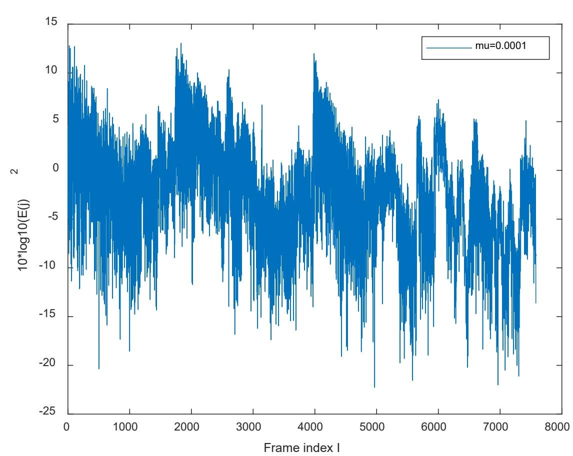

Figure 5. Convergence curve for N=64

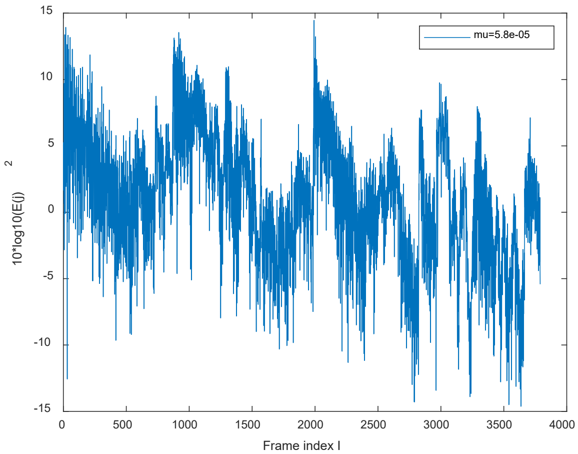

Figure 4. Convergence curve for N=128

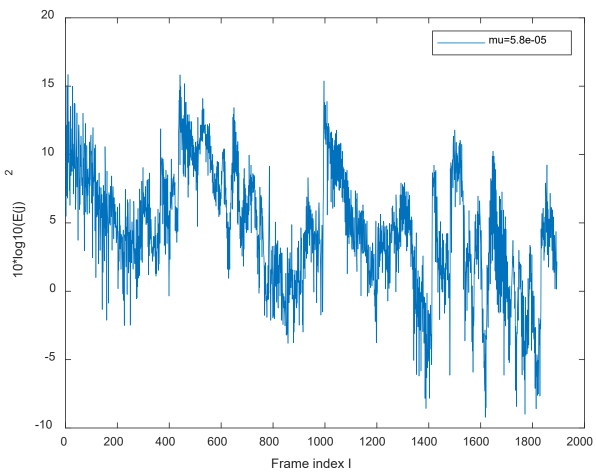

Figure 5. Convergence curve for N=256

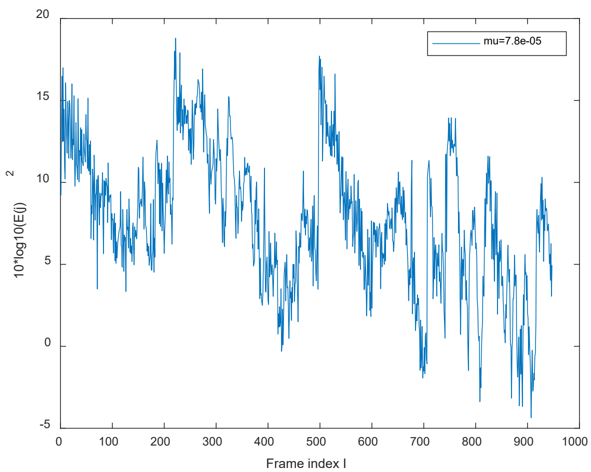

Figure 6. Convergence curve for N=512

Figure 3 to 8 exhibit how the algorithm converges under different N's. It can be observed that the piecewise decrease with time for all four cases at about 1/3 of the signal length, all six curves experience a big jump, which signifies a large noise suddenly introduced at that point. The filter has to restart the adaptation process the convergence curve of the larger N is above that of the smaller N. This is because the system with larger N has larger initial error. Therefore it will take longer time for large N system to converge if the step size is small.

Conclusions

In this paper, an FFT-based adaptive noise canceller is implemented and tested with white noise. Implementation is simulated with MATLAB.

Some conclusions can be drawn from the simulation results: resultant signal is improved by measures of SNR and subjective rating. As the order of the FIR filter, N, increase, the filtering effect gets better and better. But SNR might not always accurately reflect how well the signal is processed. The adaptive step size u plays a significant role in the adaptive process. Large u gives fast adaptation. Small u gives slow adaptation. Large u might also lead to instability and large mis-adjustment. Therefore trade-off shall be made between the adaptation rate and mis-adjustment In order to improve the convergence performance of the algorithm, two methods can be used:

1. Instead of using a constant u for each fixed N, we can vary it during the adaptive process according how large the error is
2. Instead of initializing the frequency components of the FIR filter bi(k) to be zero for i=1, we can initialize them with some other values that will give better initial condition and thus faster convergence.

Since both of above techniques require more in-depth knowledge in the field of adaptive filters. They will be the future work to explore.

Acknowledgments

I would like to thank my advisor for the valuable insight provided to me on this topic.

References

[1] N. Wiener, *Extrapolation, Interpolation and Smoothing of Stationary Time Series, with Engineering Applications*, Wiley, 1949.
[2] B. Widrow and M. Hoff, *IRE WESCON CONV. REC., Adaptive Switching Circuits*, pp. 96-104,1960.
[3] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, 1996.
[4] A. V. Oppenheim, R. W. Schafer with J. R. Buck, *Discrete-Time Signal Processing*, 2nd ed., Prentice Hall, 1999.
[5] J. G. Proakis, M. Salehi, *Communication Systems Engineering*, Prentice Hall, 1994.
[6] Mathworks, *MATLAB online documentation*, 2024.