Adaptive Noise Cancellation using Normalized Mean Square Algorithm

Analysis & Evaluation of adaptive FIR filter for noise cancellation.

Devyash Sanghai *(Author)*

Computer and Information Science

University of Florida

Gainesville, United States of America

devyashsanghai@gmail.com

*Abstract*— The goal of this paper is to analyze and evaluate adaptive Finite Response Filter using the NLMS algorithm. An adaptive filter is a filter that adjusts its transfer function according to optimizing adaptive algorithm. The efficiency of the adaptive filter is being tested for Normalized Mean Square Algorithm.

*Index Terms*—Finite Impulse Response Filter, Normalized Mean Square Algorithm

# Introduction

Filtering refers to removing unwanted signal from the input signal and giving the desired signal. Such as below where, sˆ is the desired output from s + n (input signal).

s + n sˆ

Filter

Fig 1.1 Direct Filter

Filters that are used for direct filtering can be either fixed or Adaptive.

1. Fixed filters – Fixed filters are used when we have prior knowledge of the unwanted signal and the desired signal. example. -They are of different frequencies. Then we can simply use a band pass filter to filter out the undesired signal.

2. Adaptive filters - An adaptive filter is a system with a linear filter that has a transfer function controlled by variable parameters and a means to adjust those parameters according to an optimization algorithm [1]. They require no prior knowledge of the desired and undesired signal characteristics Moreover; adaptive filters have the capability of adaptively tracking the time variations of input statistics.

## FIR Filter

A finite impulse response (FIR) filter is a filter whose impulse response (or response to any finite length input) is of finite duration, because it settles to zero in finite time [2].

y(n)=) (1.A.1)

H(z)= (1.A.2)

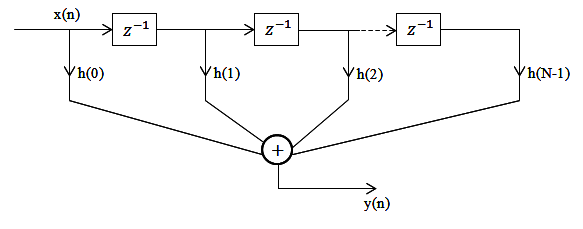
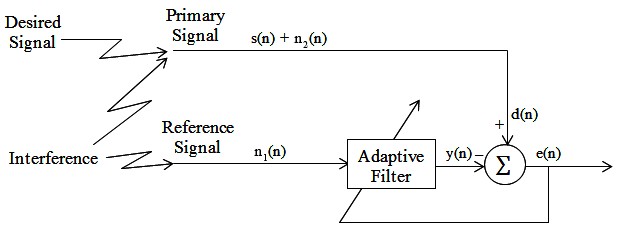


Fig 1.A.1 FIR Traversal Filter

# Adaptive Noise Cancelation



In ideal case, Adaptive Noise Canceller (ANC) has two inputs – primary signal and reference signal. The primary signal consists of desired signal and interference; the reference signal consists of only interference signal. We assume the interference input receives a noise *n1* uncorrelated with the signal but correlated in some way with the noise *n2*. The reference signal *n1* passes through a filter to produce an output that is a close estimate of primary input noise which is how adaptive filters function. This noise estimate is subtracted from the corrupted signal to produce an estimate of the signal at *s*ˆ, or e(n).

In noise canceling systems a practical objective is to produce a system output *s*ˆ = *s + n – n*ˆ that is a best fit in the least squares sense to the signal s. This objective is accomplished by feeding the system output back to the adaptive filter and adjusting the filter through an LMS adaptive algorithm to minimize total system output power. In other words, the system output serves as the error signal for the adaptive process.

Assume that s, n0, n1 and y are statistically stationary and have zero means. The signal s is uncorrelated with n0 and n1, and n1 is correlated with n0.

*s*ˆ = *s* + *n* – *n*ˆ

 *s*ˆ 2 = *s*2 + (*n*- *n*ˆ )2 + 2 s (*n* - *n*ˆ )

s is uncorrelated with n0 and *n*ˆ (Initial Assumption)

*E*[ *s*ˆ 2] = *E*[*s*2] + *E*[(*n* - *n*ˆ )2] + 2*E*[*s*(*n* - *n*ˆ )]

= *E*[*s*2] + *E*[(*n* - *n*ˆ )2]

The signal power *E*[s2] will be unaffected as the filter is adjusted to minimize *E*[ *s*ˆ 2].

 min *E*[ *s*ˆ 2] = *E*[*s*2] + min *E*[(*n* - *n*ˆ )2]

Thus, when the filter is adjusted to minimize the output noise power *E*[ *s*ˆ 2], the output noise power *E*[(*n* - *n*ˆ )2] is also minimized. Since the signal in the output remains constant, therefore *minimizing the total output power maximizes the output signal-to- noise ratio*. [3]

Since ( *s*ˆ - *s*) = (*n* – *n*ˆ )

# adaptive FILTERING algorithm

## Normalized Least Mean Square Algorithm

Normalized Least Mean Square Algorithm removes the dependence of the correction coefficient in the weight calculation. In LMS Algorithm, Input has a correction coefficient of 2 µ e(n) x(n) which is proportion to the value of x (n).

Where, û is the step size parameter of NLMS 0 < û < 2 and

|| \* || is Euclidean norm.

The tap weight w(n) is now presented as:

w(n+1)=w(n)+2μe(n)x(n)=w(n)+2e(n)x(n)

NLMS Algorithm can be summarized as:

|  |  |
| --- | --- |
| Inputs: | Tap weight vector w(n), Input vector x(n), and desired output d(n) |
| Outputs: | Filter output y(n), Tap weight vector update w(n+1) |
| Parameters | M=number of taps  ð=small constant  û=step size parameter of the NLMS algorithm 0<û<2 |
| Initialization | Having prior knowledge to compute w(0) or set it to some random value(maybe 0) |
| Step 1: Filtering  y(n)=(n)x(n)  Step 2: Error initialization  e(n)=d(n)-y(n)  Step 3: Tap weight vector adaptation:  w(n+1)=w(n)+2μe(n)x(n)=w(n)+2e(n)x(n) | |

Computation Complexity of NLMS algorithm:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Equations | \* | + or - | / |
|  | Initialization: w(0)=0 | - | - | - |
|  | For n=1… | - | - | - |
| 1 | y(n)=(n)x(n) | L | L-1 | - |
| 2 | e(n)=d(n)-y(n) |  | 1 | - |
| 3 | w(n+1) =w(n)+2μe(n)x(n)=w(n) +  +2e(n) x(n | 2L+2 | 2L | 1 |
|  | Total | 3L+2 | 3L | 1 |

Thus we can see that the Complexity of the NLMS Algorithm is O(3L+2) = O(n) Linear Time Complexity. [4]

# Analysis of the given INPUT

We shall now consider further analysis of NLMLS algorithm using a real world sample data. In the ‘project.mat’ we are being given two signals:

1. Reference signal

2. Primary signal

and sample rate fs.

Now as per ideal case (Assumed in the section II) the given data is not true. Since the data set given to us is a real world data the reference signal contains signal correlated with the desired voice.

The input signals are defined as follows:

Primary Signal = *s* + *n*

Reference Signal = s’ + n’

where:

s = the desired signal,

s' = a signal that is correlated with the desired signal s,

u = an undesired signal that is added to s, but not correlated with s or s'

n' = a signal that is correlated with the undesired signal n, but not correlated with s or s’,

Since in the given data reference signal or the input signal includes components of the desired signal. This means s' ≠ 0.

**Perfect cancelation of the undesired interference is not possible in the case**, but improvement of the signal to interference ratio is possible.

The output will be

e (n) =d (n) – y(n).

= s + n -  *s*ˆ - *n*ˆ

The desired signal will be modified (usually decreased).

The output signal to interference ratio has a simple formula referred to as power inversion.

Where

 = output signal to interference ratio.

) = reference signal to interference ratio.

z = frequency in the z-domain.[1]

This formula means that the output signal to interference ratio at a particular frequency is the reciprocal of the reference signal to interference ratio [5]

# Simulation Result and Analysis

## Performance Surface Contour

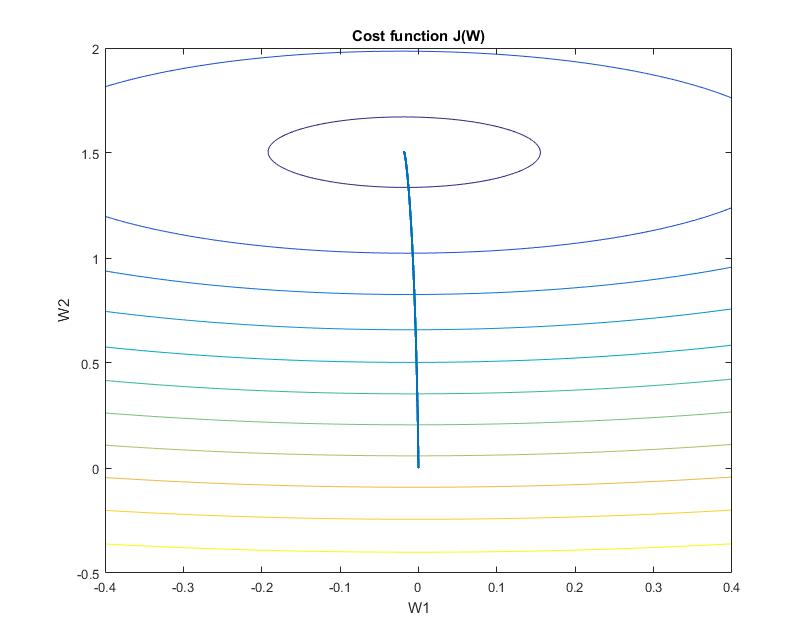


Fig: V.A.1

A performace surface contour above is plotted by first plotting the lower layer by calculating Mean square error for every w1 and w2 between w2:-0.4 to 0.4 w1: -0.5 to 2 and overlaying the points of w1 and w2 that we got during out iterations.

The information that can be interpreted from the above figure is that NLMS algorithm reaches minimum cost over iteration. The weights are adjusted over iteration such that cost function converges to zero.

## Learning Curve

A learning curve is the plot of error vs the iterations of the filter. A learning curve tell us about the time required for the adaptive filter to learn the input signal and predict the desired signal successfully.

Here we plot the error vs iterations which tells us when the filter converges by variating step size. μ controls how fast and how well the algorithm converges to the optimum filter coefficients. If μ is too large, the algorithm will not converge. If μ is too small the algorithm converges slowly and may not be able to track changing conditions. If μ is large but not too large to prevent convergence, the algorithm reaches steady state rapidly but continuously overshoots the optimum weight vector [1].

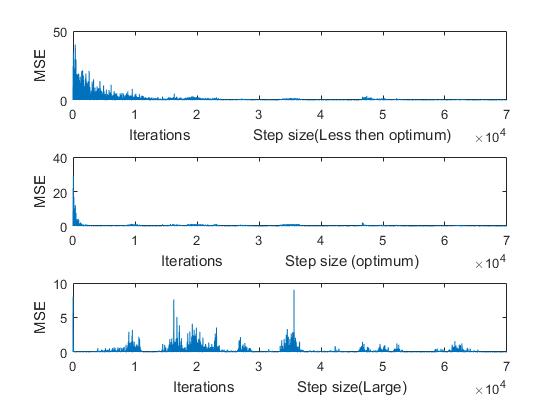


Fig. V.B.1

The above figure was simulated in matlab by keeping the step size (µ) as 0.001,0.01,1 with Filter order (M) =14.

As we can see convergence for very small step size takes lot of time. Approximately 29510 iterations.

For Optimum(Approx.) step size, adaptive filter converges in approximately 2829 iterations.

For Step size very large, adaptive filter goes to zero at the 56th iteration but overshoots every time.

## SNR Improvement

Here we are calculating the e

## Cross-Validation based on ERLE

Cross –Validation of

## Best Step Size

## Best Filter Order

# Improved Performance By exchanging Reference signal with primary signal

# Acknowledgment

.

# References

1. https://en.wikipedia.org/wiki/Adaptive\_filter
2. https://en.wikipedia.org/wiki/Finite\_impulse\_response
3. http://www.cs.cmu.edu/~aarti/pubs/ANC.pdf
4. ADAPTIVE FILTERING ALGORITHMS FOR NOISE  
   CANCELLATION Authorss: Rafael Merredin Alves Falcão
5. Widrow Page 313