Adaptive Noise Cancellation using Normalized Mean Square Algorithm

Analysis & Evaluation of adaptive FIR filter for noise cancellation using NLMS Algorithm for various performance measures.

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. The paper show various performance measure that can be used to measure the performance of adaptive noise cancellation system using NLMS algorithm. The paper builds to paramount part where it shows cross validation using Monte Carlo/random sub sampling validation using ERLE as a performance measure and cross validates the model.**

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

I. 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).

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.

A. 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].

$$y(n)=\sum_0^{n-1} h(k)x(n-k) \quad (1.A.1)$$

$$H(z)=\sum_0^{n} h(k)z^{-k} \quad (1.A.2)$$

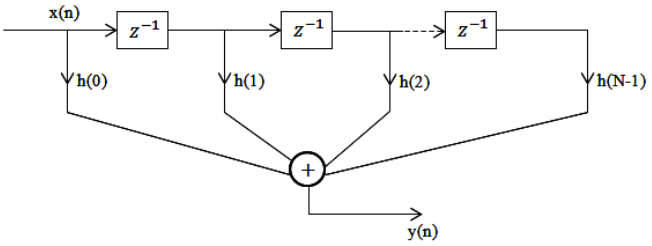
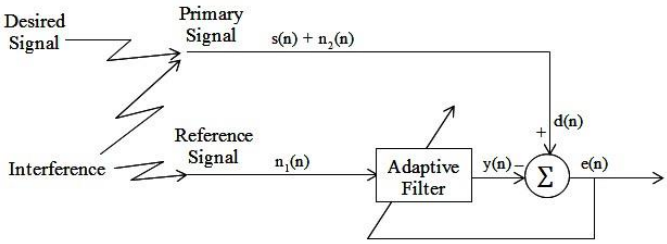


Fig 1.A.1 FIR Traversal Filter

II. 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 n_1 uncorrelated with the signal but correlated in some way with the noise $n2$. The reference signal n_1 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 $\hat{s}$, or e(n).

In noise canceling systems a practical objective is to produce a system output $\hat{s} = s + n - \hat{n}$ 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, n_0, n_1 and y are statistically stationary and have zero means. The signal s is uncorrelated with n_0 and n_1, and n_1 is correlated with n_0.

$$\hat{s} = s + n - \hat{n}$$

$$\Rightarrow \quad \hat{s}^2 = s^2 + (n - \hat{n})^2 + 2s(n - \hat{n})$$

s is uncorrelated with n_0 and $\hat{n}$ (Initial Assumption)

$$E[\hat{s}^2] = E[s^2] + E[(n - \hat{n})^2] + 2E[s(n - \hat{n})]$$

$$= E[s^2] + E[(n - \hat{n})^2]$$

The signal power $E[s^2]$ will be unaffected as the filter is adjusted to minimize $E[\hat{s}^2]$.

$$\Rightarrow \quad \min E[\hat{s}^2] = E[s^2] + \min E[(n - \hat{n})^2]$$

Thus, when the filter is adjusted to minimize the output noise power $E[\hat{s}^2]$, the output noise power $E[(n - \hat{n})^2]$ 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 $(\hat{s} - s) = (n - \hat{n})$

III. ADAPTIVE FILTERING ALGORITHM

A. 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).

$$\mu = \frac{\hat{u}}{\delta + ||x(n)||}$$

Where, û is the step size parameter of NLMS $0 < \hat{u} < 2$ and || * || is Euclidean norm.

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

$$w(n+1)=w(n)+2\mu\, e(n)x(n)=w(n)+2\frac{\hat{u}}{\delta+||x(n)||}e(n)x(n)$$

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 $0<\hat{u}<2$
Initialization	Having prior knowledge to compute w(0) or set it to some random value(maybe 0)
Step 1: Filtering $y(n)=w^t(n)x(n)$ Step 2: Error initialization e(n)=d(n)-y(n) Step 3: Tap weight vector adaptation: $w(n+1)=w(n)+2\mu\, e(n)x(n)=w(n)+2\frac{\hat{u}}{\delta+\|x(n)\|}e(n)x(n)$	

Computation Complexity of NLMS algorithm:

Step	Equations	*	+ or -	/
	Initialization: w(0)=0	-	-	-
	For n=1…	-	-	-
1	$y(n)=w^t(n)x(n)$	L	L-1	-
2	e(n)=d(n)-y(n)		1	-
3	$w(n+1) =w(n)+2\mu\, e(n)x(n)=w(n)$ + $+2\frac{\hat{u}}{\delta+\|x(n)\|}e(n)\, x(n$	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]

IV. 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 $s' \neq 0$.

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

$$e(n) = d(n) - y(n).$$
$$= s + n - \hat{s} - \hat{n}$$

The desired signal will be modified (usually decreased). [5]

V. Performance Measure

In this section we look at various performance measure that we will be using to analyze the output signal after applying the NLMS Algorithm.

A. Mean Square Error

The mean square error(MSE) is the mean square value of the difference between the desired signal and the filter output. It is devined as

$$\text{M.S. } E = E\left[[d(n) - \hat{y}(n)]^2 \right]$$

E{.} denotes mathematical expectation. .

$\hat{y}(n)$ denotes output of the adaptive filter.

M.S.E may not be the best performance measure for 2 reasons

- When M.S. E converges to level of noise, or to small value. It does mean $\hat{y}(n)$ converges to y.
- M.S.E does not give an explicit precise measure of echo attenuation since it depends on variance of the additive noise[11]

B. Echo Return Loss Enhancement

Main objective to assess the echo cancellation by the adaptive filter is the echo return loss enhancement.

$$\text{ERLE} = \frac{E[y^2(n)]}{E[\, y(n)\, \hat{y}n\,]}$$

$$ERLE_{dB} = 10log_{10}\frac{d(n)}{e(n)}$$

Where d(n) is the desired singal

And e(n) is the actual achieved

The performance of the echo canceller can be improved with step-size control.

VI. Simulation Result and Analysis

A. Performance Surface Contour

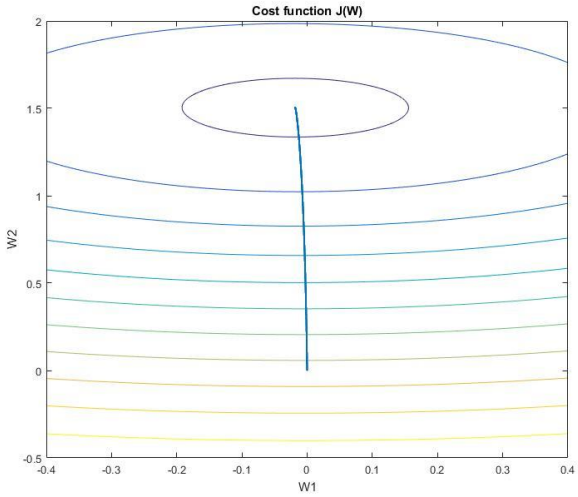


Fig: VI.A.1

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.

B. Learning Curve

A learning curve is the plot of mean square 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].

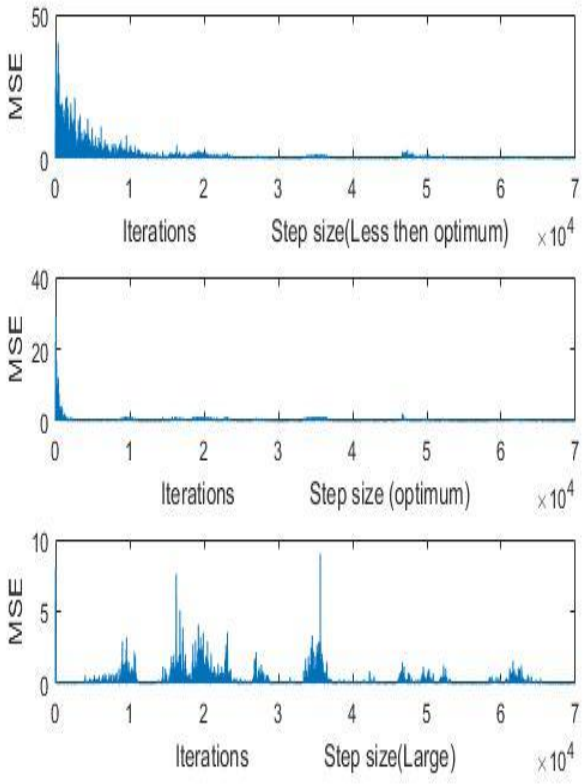


Fig. V.B.1

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.

Step size value should lie between:
$0 < \mu < 1/\lambda max$ [6]

Overall we can summarize,

Step Size	Iteration required to converge(Approximately)
0.001	29,510
0.01	2,829
1	Does not converge

C. Signal to Noise Ratio Improvement as a performance measure.

Signal-to-noise ratio (SNR) is a measure compares the level of a desired signal to the level of background noise. It is defined as the ratio of signal power to the noise power, expressed in decibels. A ratio higher than 1:1 (greater than 0 dB) indicates more signal than noise. [7]

In Adaptive noise cancellation, since we do not know the value to the correct signal exactly. We calculate SNR improvement using the ratio of the power of the output signal to the power of error in the input signal.

$$ERLE = \frac{P_{output\ voice\ signal}(z)}{P_{input\ voice+noise\ signal}(z)}$$

$$ERLE_{dB} = 10log_{10}\frac{e(n)}{X(n)}$$

Keeping the filter order constant (M=2), We variate the step size from 0.0001 to 0.01.

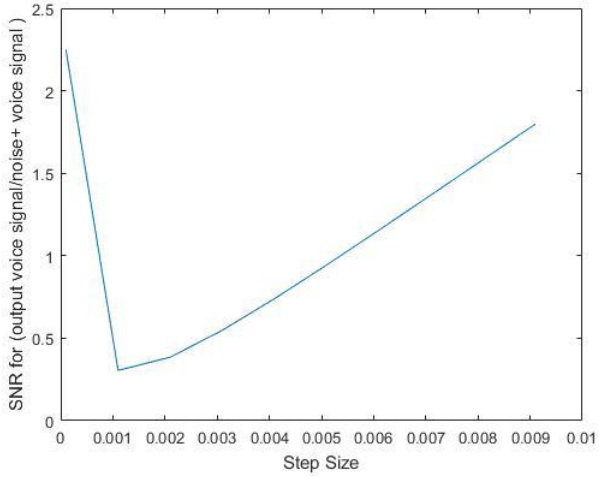


Fig V.C.1

From the above plot we can interpret that at step size 0.0011 we get the highest ERLE (negative). **The voice becomes**

decipherable, Albeit not without considerable background noise.

D. Cross-Validation based on ERLE

Cross-validation is a model validation technique for assessing how the results of a statistical analysis will generalize to an independent data set.[9] It is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice[8].

Common types of cross-validation

- Exhaustive: It does all possible ways to divide the original sample into training and test.
- Non Exhaustive: It does not compute all ways of splitting the original sample.

Although, we could do exhaustive cross validation,
we shall only do non exhaustive cross validation due to computation limitations.
We use repeated **random sub sampling validation**, also called as **Monte Carlo cross-validation** which randomly splits the dataset into training and validation data. For each such split, the model is fit to the training data, and predictive accuracy is assessed using the validation data. The results are then averaged over the splits.
Advantages over other techniques:

- over k-fold cross validation: The proportion of the training/validation split is not dependent on the number of iterations (folds).

Disadvantage over other techniques

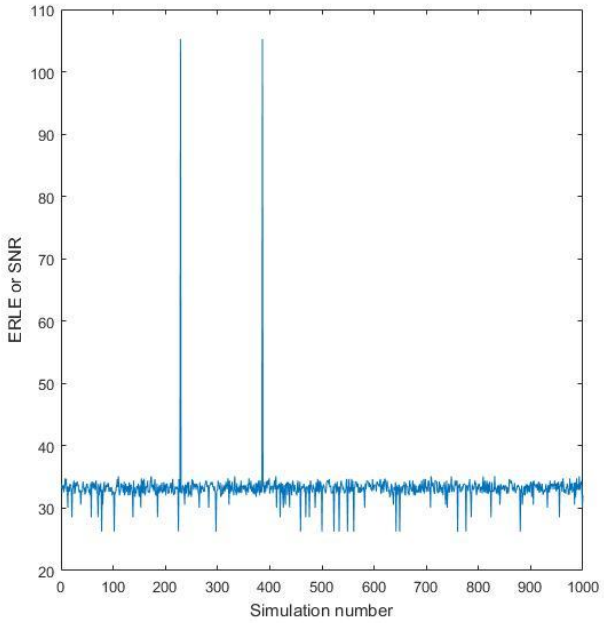
- validation subsets may overlap
- Some observations may never be selected in the validation subsample.

This method also exhibits Monte Carlo variation, meaning that the results will vary if the analysis is repeated with different random splits.

As the number of random splits approaches infinity, the result of repeated random sub-sampling validation tends towards that of leave-p-out cross-validation. [10]

Below we can find the simulation done for 1000 iterations, Filter order=14 and step size=0.1. In each iteration, we have plot the value of ERLE by randomly dividing (Gaussian Distribution) the input signals into training and validation set.

.

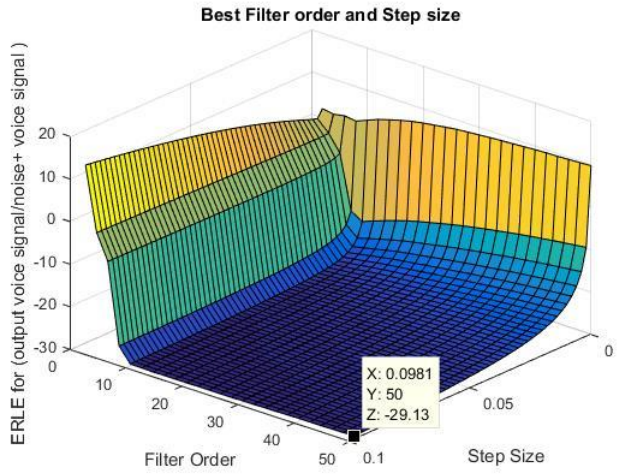


Average value of ERLE= 33.237dB

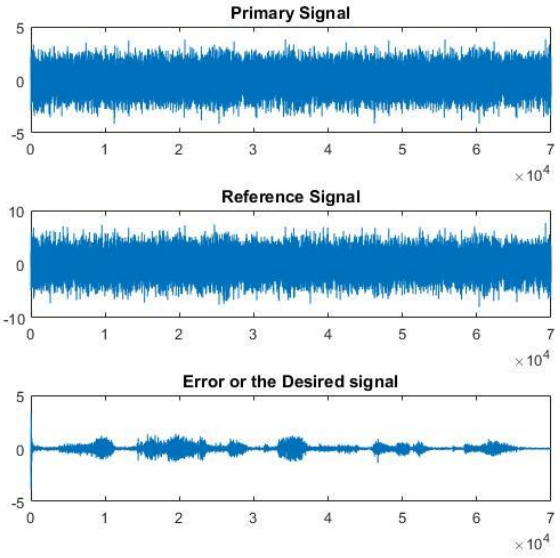
The sudden high spikes in the output indicate very accurate prediction of the actual signal. The sudden low spikes indicate when the signal.

E. Best Filter Order using ERLE.

In this section we shall show the best filter order ERLE. We later corroborate the performance measure by showing playing the sound, since the result of sound cannot be shown in the paper, we shall just show the plot of sound signal for optimum parameters and random parameters for comparison.



The above plot is plotted by variating the step size from 0.0001 to 0.1 and filter order from 3 to 50 and plotting the respective ERLE. The best filter order and step size come out to be at filter order 50 and step size 0.0981.



From the above plot we can infer that the output is cleaned of the initial background noise. The **SNR for the output is 29.13** Although this is not the cleanest noise. We got the cleanest noise at filter order 14 and step size 0.1. ERLE here is only measuring the ratio of output voice to original signal. As we don't have the original signal ERLE cannot completely predict the correct filter order and step size

.

VII. Conclusion

We have effectively analyzed performance of FIR filter using Normalized Mean Square Algorithm for adaptive noise cancellation. We showed that if the desired voice signal correlation is present in both the input the noise cannot be completely clean. Then we plot the performance measure of the NLMS algorithm we showed that the filter converges to a minimum. We performed Monte Carlo Cross Validation using the ERLE to proof our model. We found out the best filter order and step size using ERLE as a cost function

References

[1] Adaptive filter. (n.d.). Retrieved October 18, 2016, from https://en.wikipedia.org/wiki/Adaptive_filter

[2] Adaptive Noise Cancellation - CMU Computer Science. (n.d.). Retrieved October 18, 2016, from http://www.cs.cmu.edu/~aarti/pubs/ANC.pdf

[3] ADAPTIVE FILTERING ALGORITHMS FOR NOISE CANCELLATION Authors: Rafael Merredin Alves Falcão

[4] Revolvy, L. (n.d.). "Adaptive filter" on Revolvy.com. Retrieved October 18, 2016, from http://www.revolvy.com/main/index.php?s=Adaptive filter

[5] [6] S. Haykin, Adaptive Filter Theory. 4th ed. Englewood Cliffs, NJ: Prentice Hall, 2002.

[7] https://en.wikipedia.org/wiki/Signal-to-noise_ratio

[8] "Newbie question: Confused about train, validation and test data!". http://www.heatonresearch.com/node/1823.

[9] (n.d.). Retrieved October 18, 2016, from https://en.wikipedia.org/wiki/Cross-validation_(statistics)#cite_note-Newbie_question:_Confused_about_train.2C_validation_and_test_data.21-4

[10] Cross Validation. (n.d.). Retrieved October 18, 2016, from http://www.cs.cmu.edu/~schneide/tut5/node42.html

[11] Sparse Adaptive Filters for Echo Cancellation. (n.d.). Retrieved October 18, 2016, from https://goo.gl/DQdClh