Adaptive Noise Cancellation using Normalized LMS Algorithm

Suraj Kota, *Student, University of Florida*

***Abstract*— In the field of Digital Signal Processing, signals are associated with noise and distortion. This is due to the fact that system undergoes the time varying physical process which are unknown sometimes. If statistical property of the signal is known then we are use a fixed filter but when the property of the signal is unknown we use adaptive filter. In many application of noise cancellation, the changes in signal characteristics could be quite fast. This requires the utilization of adaptive algorithms which converge rapidly. Least Mean Squares (LMS) and Normalized Least Mean Squares (NLMS) adaptive filters have been used in a wide range of signal processing application because of its simplicity in computation and implementation. In this paper we design and evaluate the performance of adaptive FIR filter using Normalized LMS algorithm for adaptive noise cancellation in order to clean he noisy speech signal. The effects on stability, convergence, speed and computation on choosing different parameters for NLMS is studied here and in the end we will decide on a system which has the best tradeoffs.**

***Index Terms*—Noise Cancellation, NLMS, MSE, Adaptive Filter**

I. Introduction

NOISE is an unwanted waveform that interfere with communication. Noise cancellation involves extraction of clean signal extraction of clean signal from a noise interfered signal. Traditional approaches to acoustic noise cancellation using passive techniques like silencers, barriers, etc. to remove noise are ineffective and costly at low frequencies.

Adaptive filtering is a methodology which involves changing of filter parameters or coefficients over time to adapt to dynamic changes in signal characteristics. This is in contrast to the conventional filter design techniques where the coefficients are constants and a priori information about the signal is known. The interesting thing about adaptive filters is their ability to self-learn.

Some of the applications of adaptive filtering are identification of unknown signals and noise cancellation. Adaptive filtering can efficiently attenuate low frequency noise as well. In order to remove noise from a garbled signal, the adaptive filter is fed a noise signal n'(k) along with the actual signal to be processed. The adaptive filter correlates the noise signal with the noise in the actual signal to extract out the actual clean signal

.

This work is submitted on 18th October 2016 to fulfill the coursework requirement for Elements of Machine Intelligence course.

Suraj Kota is with the University of Florida, Gainesville, FL 32611 USA (e-mail: suraj.k@ufl.edu).

II. Adaptive Algorithm

A. Least Mean Square Algorithm

In 1960, Widrow and Hoff presented the LMS algorithm. LMS algorithm was studied based on Minimum Mean Square Error (MMSE) and the steepest descent algorithm. The least-mean-square (LMS) algorithm belongs to the family of the linear stochastic gradient algorithms. It serves at least two purposes. First, it avoids the need to know the exact signal statistics (e.g., covariance and cross-covariance), which are nevertheless rarely available in practice. Second, these methods possess a tracking mechanism that enables them to track variations in the signal statistics and automatically adjust the tap weights using the estimated error which is computed by comparing the output of the filter with a desired response.

Let $x(n)$ and $d(n)$ represent the reference input and the desired output signal, respectively, to the adaptive filter. Let m denote the total number of filter coefficients.

Define the L x 1 coefficient vector $\omega(n)$ and the input vector $X(n)$ as

$$\omega(n) = [\omega_o(n), \omega_1(n),..., \omega_{m-1}(n)]^T \quad (1)$$

$$X(n)= [x(n),x(n-1),\ldots,x(n- L+1)]^T \quad (2)$$

$\omega(n+1)$ is estimated according to the following recursion:

$$\omega(n + 1) = \omega(n) + 2\mu X(n)e(n) \quad (3)$$

where e(n) is,

$$e(n) = d(n)- X^T(n)X(n) \quad (4)$$

and μ is a small positive constant, called the step size, which controls system stability and convergence rate. Because μ is a constant, convergence rate is slower, and misadjustment coefficient M is still bigger when the system reaches steady state. M could be expressed as:

$$\mathrm{M} = \mu \mathrm{L} X_{in} \quad (5)$$

L is filter order, X_{in} is input signal power. As can be seen from Equation (5), when the step size μ, input signal power X_{in}, or filter order L is large, the misadjustment coefficient M is large.

LMS algorithm cannot achieve both rapid convergence and small steady-state error simultaneously.

B. *Normalized Least Mean Square Algorithm*

Normalized LMS algorithm is an improved algorithm of LMS. The drawback of LMS algorithm is the selection of step size μ. In order to solve this difficulty, we use NLMS where the step size is normalized.

$$\mu(n) = \frac{\alpha}{c + ||\, x(n)||^2} \tag{6}$$

where $||\, x(n)||^2$ is calculated by the power of the input signal $X^T(n)X(n)$. Here α (0< α<2)is the adaptation constant, which optimizes the convergence rate of algorithm. c in the denominator is a small positive constant used to avoid possible large step size that may result that the NLMS algorithm will be diverged when $||\, x(n)||^2$ is very small.

The step size μ(n) varies adaptively by following the changes in the input signal level. This prevents the update weights from diverging and makes the algorithm more stable and faster converging than when a fixed step size is used.

In NLMS $\omega(n+1)$ is estimated according to the following recursion:

$$\omega(n+1) = \omega(n) + \frac{2\alpha x(n)e(n)}{c+||\, x(n)||^2} \tag{7}$$

III. System Design For Adaptive Noise Cancellation

ANC deals with the enhancement of noise corrupted signals. Its biggest strength when compared to other enhancement techniques is that no a priori information about either the signal or noise is required. This advantage comes at the cost that we need in addition to the reference signal *d(n)* which is Signal corrupted by noise signal *x(n)* which contains noise $x_1(n)$ correlated with the noise component $x_0(n)$ in the primary signal. The principle here is to adaptively filter the reference signal to get a replica of the noise in the primary and subtract it from the primary. Here the system output is the error signal.

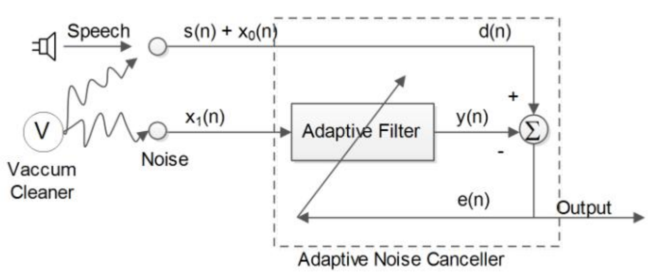

Figure 1: Block Diagram of ANC System

Now we can see that:

$$e(n)=s(n)+x_0(n)-y(n) \tag{8}$$

Squaring and taking the expectation operation on both sides gives us:

$$E\{e^2(n)\}=E\{s^2(n)\}+E\{(x_0(n)-y(n))^2\}+2E\{s(n)(x_0(n)-y(n))\} \tag{9}$$

Assuming that s(n) is uncorrelated with $x_0(n)$-and $x_1(n)$- we get that *s(n)* is also uncorrelated with *y(n)* since *y(n)=x(n)*w(n)*. Thus the last term above becomes zero and we get:

$$E\{e^2(n)\}=E\{s^2(n)\}+E\{(x_0(n)-y(n))^2\}+2E\{s(n)(x_0(n)-y(n))\} \tag{10}$$

Now to minimize the error we cannot reduce *s(n)* and hence we get the MSE minimized when $y(n)= x_0(n)$.This implies that *e(n)= s(n)*, the desired result.

Also if we assume, $x_0(n)= x_1(n) * w(n)$ where the transmission path between $x_0(n)$ and $x_1(n)$ has been modeled by *w(n)*, we can conclude that the adaptive filter models the unknown transmission path *w(n)*. We assume the noise components are correlated. If they are uncorrelated then the speech will further be degraded as there is another additive noise component.

In actual implementation, the signal from Project1.mat provided as *reference* is taken as $s(n)+x_0(n)$ and signal provided as *primary* is taken as $x_1(n)$ and learning is carried using equation (7) with c = 0.001 and input vector X is created as in equation (2).

IV. Experimental Analysis

In this section, we observe the performance of the above system with different filter orders and step sizes and try to find the optimal filter order that would work for the given speech signal. We first analyze the behavior of system with a 2-tap filter and then see if some of the insights obtained can be generalized when using other filter orders and step sizes.

A. *Extraction of speech using NLMS with a 2-tap filter*

Here, we tried to optimize the filter weights of a 2-tap filter for noise cancellation using NLMS algorithm for various step sizes i.e. 0.001, 0.01, 0.05, 0.1 and 0.5. The system was allowed to change the filter weights throughout all the iterations.

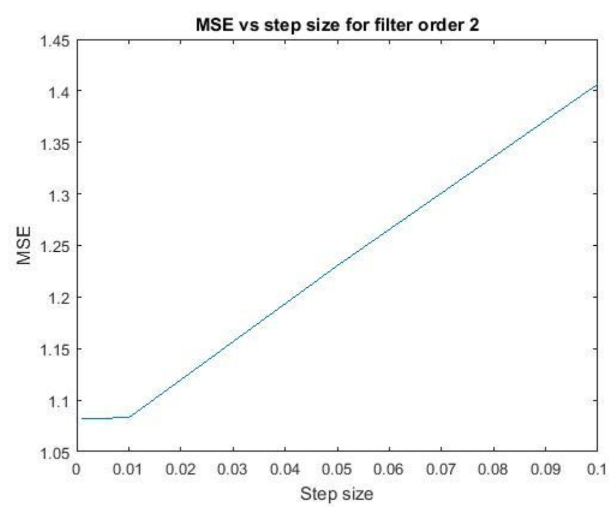

Figure 2: MSE vs Step size for filter order 2

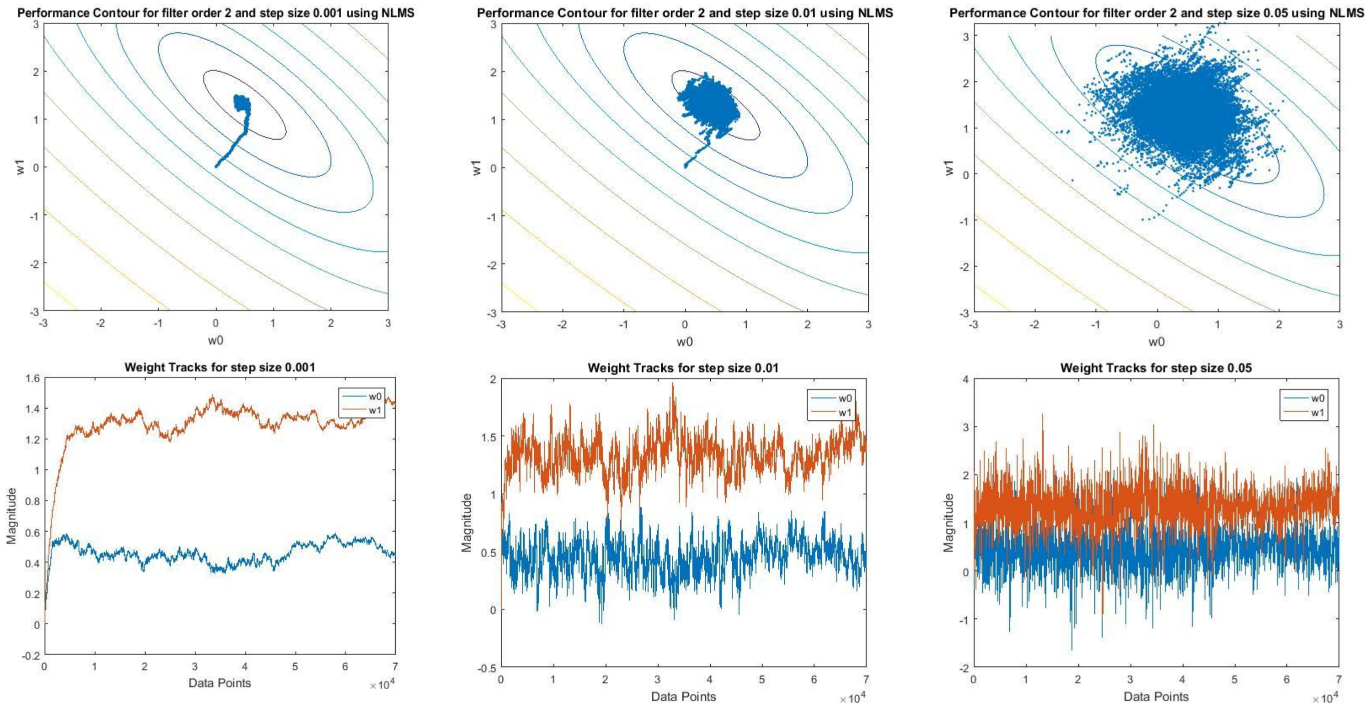

Figure 3: Performance Contours(Above) & corrosponding weight track graphs(below) for different step size and filter order 2

From figure 2, we can say that the quality of clean speech reduces drastically on increasing the step size more than 0.01. Let us see why this happens from figure 3. Figure 3 shows the performance contours and corresponding weight tracks of the filter weights for step sizes 0.001, 0.01 and 0.05. From the weight tracks we can infer that in case of step size 0.01 and 0.05, the weights have not converged and the weights keep on changing over the iterations. This observation is validated from the performance contours which shows how the filter weight values change as the algorithm moves towards the optimality. From the performance contours in figure 3, we can see that, once the weights reach near the minimum MSE cost the optimal weights keep on moving as the system is not stable for this step size and filter order. One reason for this is that we have over trained the filter even after it has reached a point after which it cannot minimize the cost any further. As we can see from the learning curve in figure 4 that the filters do not converge. Filter order 0.5 and 0.1 have the worst performance. 0.5 step size is the case of rattling where the iterative process continues to wander around the neighborhood of optimal *w* without ever stabilizing. Among the other step sizes, 0.01 has a better convergence rate among others but the step size is so small that the filter takes a very long time to converge. We can see that even till the end of iterations the error is decreasing and we do not have a linear shape where we can say that the filter has converged. For step sizes 0.05 and above the filter does not even converge, it is still in learning stage trying to decrease the error.

Figure 5 shows a bar graph of Signal to Noise ratio (SNR) for different step sizes. SNR is calculated as

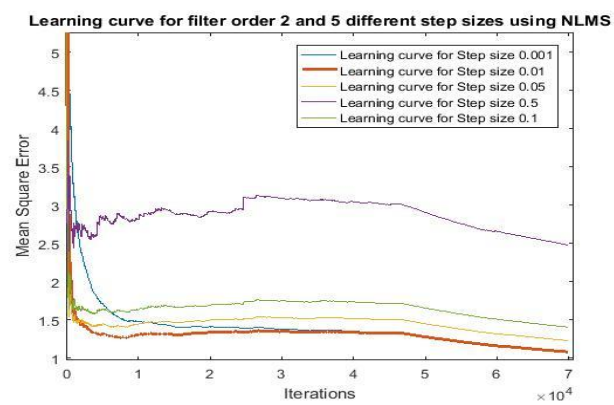

Figure 4: Learning curve for different step sizes for filter order 2

$$, SNR = 10 \log \frac{\|d(n)\|^2}{\|e(n)\|^2} \tag{11}$$

where, d(n) is the desired and *e(n)* is the error signal. In our calculation we consider the signal with speech plus noise as *d* and the clean speech signal obtained as *e*. So, the higher the SNR better the performance of the system. We can see that for step sizes 0.001 and 0.01 the SNR value is the highest which tells us that less of noise was accumulated with this step size than when the system used step sizes 0.1 or 0.5.

For filter order 2, with any step size the voice of the speaker was not audible. This means that this filter order is not a good solution for this signal.

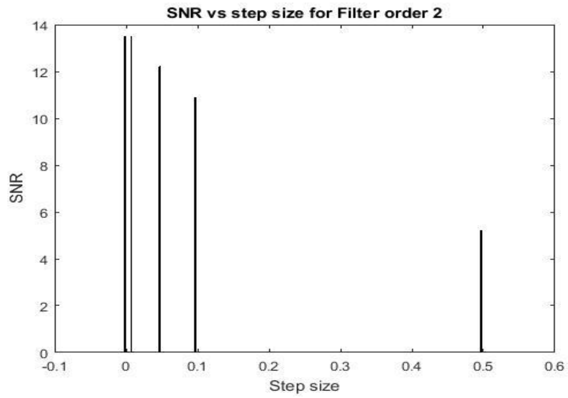

Figure 5: SNR for different step sizes and filter order 2

B. Extraction of speech using NLMS with higher filter orders

Here, we tried to optimize the filter weights of a different filter orders for noise cancellation using NLMS algorithm for various step sizes i.e. 0.001, 0.01, 0.05, and 0.1. The system was allowed to change the filter weights throughout all the iterations. Let us examine the optimal step sizes for each selected filter order.

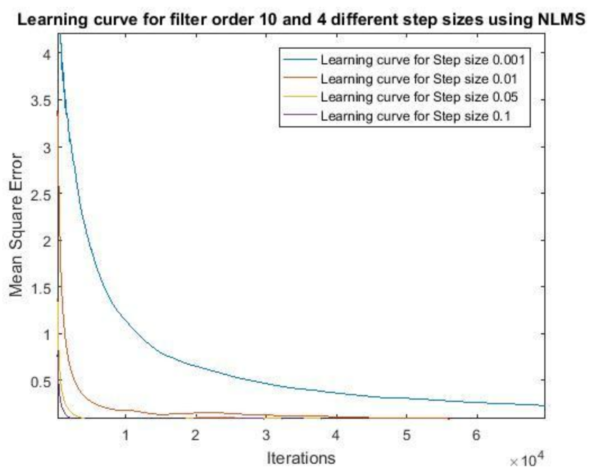

Figure 6: Zoomed view of learning curve for filter order 10 and different step sizes

From figure 6,7 and 8 we can see that 0.1 is the fastest converging step size for all the filter weights and also achieves minimum MSE. However, figure 9 shows that the MSE accumulated by these filter orders is different and filter order 10 shows a better performance. We can also see from figure 10 that unlike the case of filter order 2 the weight tracks show a better convergence shape. Also note that the weight tracks change all of a sudden at some point near between 45000 and 50000. This is because, the statistics of signal would have changed at that point. The learning curve is still smooth and so we can say that the system adapts to the changes nicely.

Figure 11 and figure 9 help us understand that SNR(computed by equation (11)) can be a measure of performance. We can see that the filters with higher SNR(figure 11) have accumulated a lower MSE(figure 9). Hence, we can do cross validation based on ERLE. Highest SNR was found for filter order 15 with value 43.33dB.

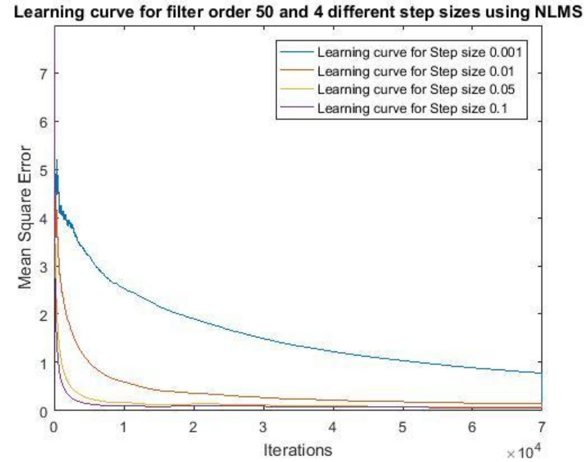

Figure 7: Zoomed view of learning curve for filter order 50 and different step sizes

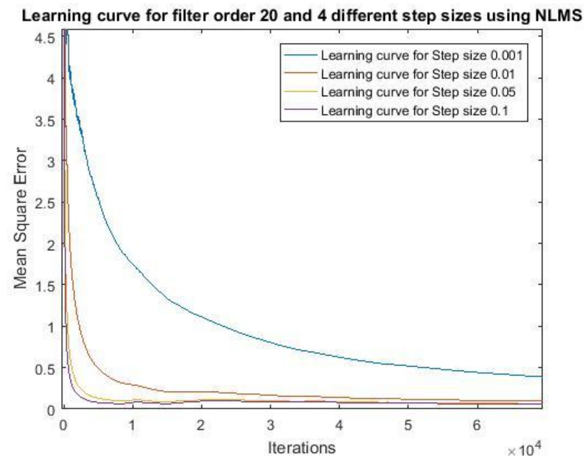

Figure 8: Zoomed view of learning curve for filter order 20 and different step sizes

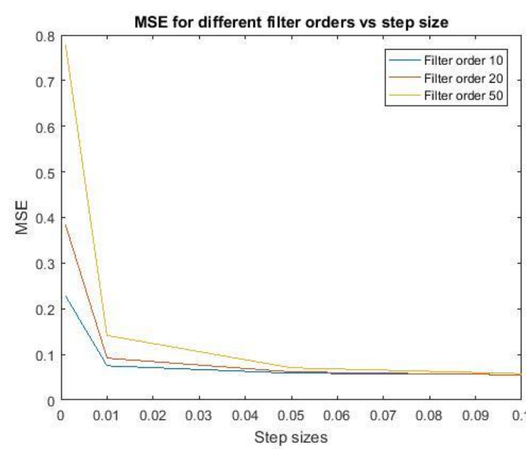

Figure 9: MSE vs Step sizes for different filter orders

Also, it was observed that higher SNR filter (15) has cleaner speech compared to others and so ERLE also is a measure for speech intelligibility.

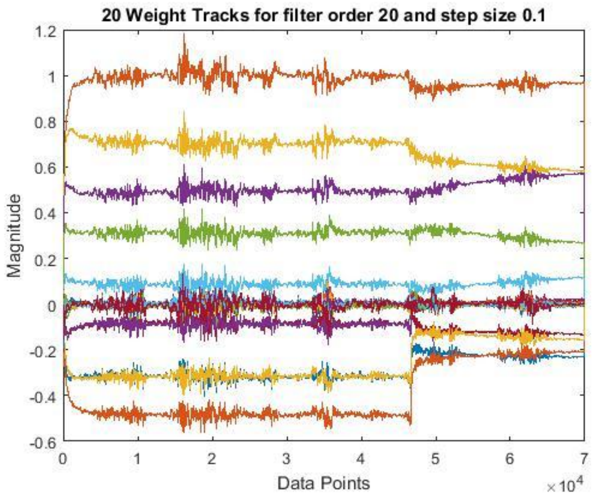

Figure 10: Weight tracks for filter order 20

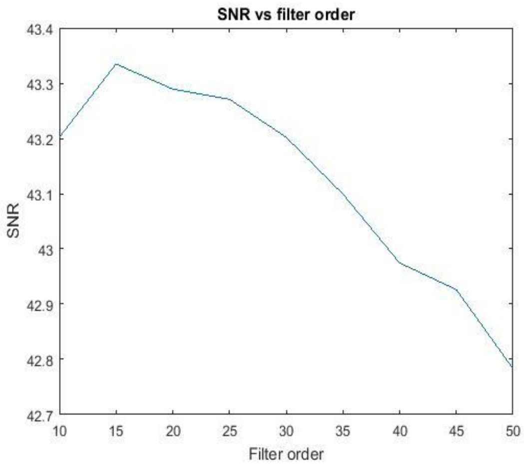

Figure 11: SNR vs filter order for step size 0.1

V. Conclusion

The optimal filter order for this problem with the given signal is found to be 15 with step size 0.1 and we can hear some one saying "**we will not condemn any repulsive actions to leads us to war**". NLMS is recommended for use in noisy environments. Increasing the filter order increased the performance of the system.

References

[1] https://www.mathworks.com/help/dsp/ug/overview-of-adaptive-filters-and-applications_f1-5512.html#f1-5567
[2] https://jmvalin.ca/papers/valin_inlms.pdf
[3] http://www.ijsr.net/archive/v4i7/SUB156322.pdf
[4] http://www.ijaiem.org/RATMIG-2013/EC%2019%20LEAST%20MEAN%20SQUARE%20ADAPTIVE%20FILTER%20FOR%20NOISE%20CANCELLATION.pdf
[5] http://www.slideshare.net/bratisundarnanda/project-34302050
[6] https://repositorio-aberto.up.pt/bitstream/10216/68177/1/000154774.pdf