Application of Classical Adaptive Filters in Speech Enhancement

Mohammad Shams Esfand Abadi[1,2], Hossein Sirousi[3],
[1] Department of Electrical Engineering, Tarbiat Modares University, Tehran, Iran
[2,3]Department of Electrical Engineering, Shahid Rajaee Teachers Training University, Tehran, Iran
E-mail: mshams@modares.ac.ir

Abstract: *In many applications of noise cancellation the changes in signal characteristics could be quite fast. This requires the utilization of adaptive algorithms, which converge rapidly. Least mean square (LMS) and Normalized LMS (NLMS) adaptive filters have been used in a wide range of signal processing applications because of its simplicity in computation and implementation. The Recursive Least Squares (RLS) algorithm has established itself as the "ultimate" adaptive filtering algorithm in the sense that it is the adaptive filter exhibiting the best convergence behavior. Unfortunately, practical implementations of this algorithm are often associated with high computational complexity and/or poor numerical properties. In this paper we have performed and compared these classical adaptive filters for attenuating noise in speech signals. In each algorithm, the optimum order of filter of adaptive algorithms have also been found through experiments.*

Keywords: Adaptive Filter, Least Mean Squares, Recursive Least Squares, Noise Cancellation.

1 Introduction

Adaptive filtering has been, and still is, an area of active research, playing important roles in an ever-increasing number of applications. In an adaptive filter, the coefficients are periodically updated according to an adaptive filtering algorithm in order to minimize a certain cost function. Least Mean Square (LMS), Normalized LMS and Recursive Least Square (RLS) are well known adaptive filter in signal processing application. The LMS algorithm is very popular because of its low computational complexity [1]. However, the convergence rate of the LMS depends on the length of the filter and on the input statistics. The Normalized Least Mean Square (NLMS) algorithm [1], [2] is also a widely used adaptation algorithm due to its computational simplicity and ease of implementation. Furthermore, this algorithm is known to be robust against finite word length effects. One of the major drawbacks of the NLMS algorithm is its slow convergence for colored input signals. The RLS algorithm is known to be capable of performing much better than the LMS algorithm but practical implementations of this algorithm are often associated with high computational complexity and/or poor numerical properties [1], [2] ,[3]. It is well known that two of most frequently applied algorithms for noise cancellation [1], [4], [5] are normalized least mean squares (NLMS) and recursive least squares (RLS) [6] algorithms. Considering the two algorithms, it is obvious that NLMS algorithm has the advantage of low computational complexity. On the contrary, the high computational complexity is the weakest point of RLS algorithm but it provides a fast adaptation rate. Thus, it is clear that the choice of the adaptive algorithm to be applied is always a trade off between computational complexity and fast convergence. In this paper we have performed and compared these classical adaptive filters for attenuating noise in speech signals. In each algorithm, the optimum order of filter of adaptive algorithms have also been found through experiments.

We have organized our paper as follows: The classical adaptive filters are reviewed in section 2. In Section 3 the adaptive noise cancellation setup will be explained. In the next section the simulation results has been presented and finally Section 5 is the conclusion.

2 The Classical Adaptive Filters

In this section we will review the theory of classical adaptive filter algorithms.

2.1 Motivation

We start our discussions by reviewing the linear estimation problem and the corresponding steepest descent methods.

Thus let d be zero-mean scalar-valued random variable with variance $\sigma_d^{\ 2} = E|d|^2$, and let u^*, that "*" is conjugate operator, be a zero-mean $M \times 1$ random variable with a positive-definite covariance matrix, $R_u = Eu^* u$. The $M \times 1$ cross covariance vector of d and u is denoted by $R_{du} = Edu^*$. The weight vector w that solves $\min E|d - uw|^2$ is given by:

$$w^0 = R_u^{-1} R_{du} \qquad (1)$$

that the resulting minimum mean square error is

$$m.m.s.e = \sigma_d^{\ 2} - R_{ud} R_u^{-1} R_{du} \qquad (2)$$

There are several steepest descent methods that approximate w^0 iteratively, until eventually converging to it. For example, the following recursion with a constant step size,

$$w_i = w_{i-1} + \mu[R_{du} - R_u w_{i-1}] \;\; w_{-1} = initial\, guess \quad (3)$$

where the update direction, $R_{du} - R_u w_{i-1}$, was seen to be equal to the negative conjugate transpose of the gradient vector of the cost function at w_{i-1},

$$R_{du} - R_u w_{i-1} = -[\nabla_w J(w_{i-1})]^* \qquad (4)$$

where $J(w) = E|d - uw|^2$. Also we can use an iteration step-size, $\mu(i)$, and studied the recursion:

$$w_i = w_{i-1} + \mu(i)[R_{du} - R_u w_{i-1}], w_{-1} = initialguess \quad (5)$$

And In Newton's recursion, we have,

$$w_i = w_{i-1} + \mu R_u^{-1}[R_{du} - R_u w_{i-1}], w_{-1} = initial\, guess \quad (6)$$

where R_u^{-1} resulted from using the inverse of the Heissian matrix of $J(w)$, namely,

$$R_u = \nabla_w^{\ 2} J(w_{i-1}) = \nabla_{w^*}[\nabla_w J(w_{i-1})] \qquad (7)$$

More generally, when regularization is employed and when the step size is also allowed to be iteration-dependent, the recursion for Newton's method is replaced by:

$$w_i = w_{i-1} + \mu(i)[\varepsilon(i) I + R_u]^{-1}[R_{du} - R_u w_{i-1}] \quad (8)$$

for some positive scalars $\{\varepsilon(i)\}$, they could be set to a constant value for all i, say $\varepsilon(i) = \varepsilon$.

2.2 The LMS Algorithm

Assume that we have access to several observations of the random variables d and u:

$$\{d(0), d(1), d(2), d(3),....\} \;\; and \;\; \{u_0, u_1, u_2, u_3,......\} \quad (9)$$

One of the simplest approximations for $\{R_{du}, R_u\}$ is to use the instantaneous values:

$$\hat{R}_{du} = d(i) u_i^*, \qquad \hat{R}_u = u_i u_i^* \qquad (10)$$

This construction simply amounts to dropping the expectation operator from the actual and replacing the random variables $\{d, u\}$ by observations $\{d(i), u_i\}$. In this way, the gradient vector in Eq.4 is approximated by the instantaneous value:

$$-[\nabla_w \hat{J}(w_{i-1})]^* = d(i) u_i^* - u_i^* u_i w_{i-1} = u_i^*[d(i) - u_i w_{i-1}] \quad (11)$$

and the corresponding steepest-descent recursion in Eq.3 becomes:

$$w_i = w_{i-1} + \mu u_i^*[d(i) - u_i w_{i-1}] \;\; w_{-1} = initial \;\; guess \quad (12)$$

we continue to write w_i to denote the estimate that is obtained via this approximation procedure although, of course, w_i in (12) is different from the w_i that is obtained from the steepest descent algorithm (3): the former is based on using instantaneous approximations whereas the latter is based on using $\{R_{du}, R_u\}$.

The stochastic-gradient approximation (12) is one of the most widely used adaptive algorithms in current practice due to its striking simplicity. It is widely known as the least mean-squares (LMS) algorithms, or sometimes as the Widrow-Hoff algorithm in honor of its originators [1].

2.3 The NLMS Algorithm

We start with the so-called normalized LMS algorithm, which can be motivated in much the same as LMS was except that now we start from the regularized Newton's recursion (8) and assume that the regularization sequence $\{\varepsilon(i)\}$ and the step size sequence $\mu(i)$ are constants, say, $\varepsilon(i) = \varepsilon$ and $\mu(i) = \mu$. Thus using:

$$w_i = w_{i-1} + \mu[\varepsilon I + R_u]^{-1}[R_{du} - R_u w_{i-1}] \qquad (13)$$

and replacing the quantities $(\varepsilon I + R_u)$ and $(R_{du} - R_u w_{i-1})$ by the instantaneous approximations $(\varepsilon I + u_i^* u_i)$ and $u_i^*[d(i) - u_i w_{i-1}]$, respectively, we arrive at the stochastic-gradient recursion:

$$w_i = w_{i-1} + \mu[\varepsilon I + u_i^* u_i]^{-1} u_i^*[d(i) - u_i w_{i-1}] \quad (14)$$

This recursion, in its current form, requires the inversion of the matrix $(\varepsilon I + u_i^* u_i)$ at each iteration. This step can be avoided by working the recursion into an equivalent simpler form. Thus note that $(\varepsilon I + u_i^* u_i)$ is a rank-one modification of a multiple of the identity matrix, and the inverse of every such matrix has a similar structure. To see this, we simply apply the matrix inversion formula to get:

$$\left[\varepsilon I + u_i^{*} u_i\right]^{-1} = \varepsilon^{-1} I - \frac{\varepsilon^{-2}}{1+\varepsilon^{-1}\|u_i\|^2} u_i^{*} u_i \qquad (15)$$

Where the expression on the right hand side is a rank-one modification $\varepsilon^{-1} I$. If we now multiply both sides of (15) by u_i^* from the right we obtain:

$$\left[\varepsilon I + u_i^{*} u_i\right]^{-1} = \varepsilon^{-1} u_i^{*} - \frac{\varepsilon^{-2}}{1+\varepsilon^{-1}\|u_i\|^2} u_i^{*} \|u_i\|^2 =$$

$$\varepsilon^{-1} u_i^{*}\left[1 - \frac{\|u_i\|^2}{\varepsilon + \|u_i\|^2}\right] = \frac{u_i^{*}}{\varepsilon + \|u_i\|^2} \qquad (16)$$

Which is a scalar multiple of u_i^*. Substituting into (14) we arrive at the ε-NLMS recursion [1]:

$$w_i = w_{i-1} + \frac{\mu}{\varepsilon + \|u_i\|^2} u_i^{*}\left[d(i) - u_i w_{i-1}\right], i \geq 0 \quad (17)$$

2.3 The RLS Algorithm

A second example of an algorithm that employs a more sophisticated approximation for R_u is the recursive-Least-squares(RLS)algorithm. Just like ε-NLMS and ε-APA, we again start from the regularized Newton's recursion (8), namely,

$$w_i = w_{i-1} + \mu(i)\left[\varepsilon(i) I + R_u\right]^{-1}\left[R_{du} - R_u w_{i-1}\right] \quad (18)$$

while we still replace $(R_{du} - R_u w_{i-1})$ by the instantaneous approximation:

$$u_i^{*}\left[d(i) - u_i w_{i-1}\right] \qquad (19)$$

we now replace R_u by a better estimate for it, which we choose as the exponentially weighted sample average:

$$\hat{R}_u = \frac{1}{i+1}\sum_{j=0}^{i} \lambda^{i-j} u_j^{*} u_j \qquad (20)$$

for some scalar $0 \leq \lambda \leq 1$. Assume first that $\lambda = 1$.

Then the above expression for $\hat{R}_u$ amounts to averaging all past regressors up to time i, namely,

$$\hat{R}_u = \frac{1}{i+1}\sum_{j=0}^{i} u_j^{*} u_j \qquad when \quad \lambda = 1 \qquad (21)$$

choosing a value for λ that is less than one introduces memory into the estimation of R_u.

This is because such a λ would assign larger weights to recent regressors and smaller weights to regressors in the remote past. In this way , the filter will be endowed with a tracking mechanism that enables it to forget data in the remote past and to give more relevance to recent data so that changes in R_u can be better tracked.

We further assume that the step-size in (18) is chosen as:

$$\mu(i) = 1/(i+1) \qquad (22)$$

whereas the regularization factor is chosen as:

$$\varepsilon(i) = \lambda^{i+1}\, \varepsilon/(i+1) \qquad i \geq 0 \qquad (23)$$

for small positive scalar ε. This choice for $\varepsilon(i)$ is such that regularization disappears at time progresses. With the above approximations and choices, the regularized Newton's recursion (18) becomes:

$$w_i = w_{i-1} + \left[\lambda^{i+1}\varepsilon I + \sum_{j=0}^{i} \lambda^{i-j} u_j^{*} u_j\right]^{-1} \times$$

$$u_i^{*}\left[d(i) - u_i w_{i-1}\right] \qquad (24)$$

this recursion is inconvenient in its present fro since it requires, ae each time instant I, that all previous and present data be combined to form the matrix:

$$\Phi_i = \left(\lambda^{i+1}\varepsilon I + \sum_{j=0}^{i} \lambda^{i-j} u_j^{*} u_j\right) \qquad (25)$$

which then needs to be inverted. These two complications (of data storing and matrix inversion) can be alleviated as follows. Observe from the definition of Φ_i that it satisfies the recursion:

$$\Phi_i = \lambda \Phi_{i-1} + u_i^{*} u_i \qquad \Phi_{-1} = \varepsilon I \qquad (26)$$

Let $P_i = \Phi_i^{-1}$. Then applying the matrix inversion formula to gives:

$$P_i = \lambda^{-1}\left[P_{i-1} - \frac{\lambda^{-1} P_{i-1} u_i^{*} u_i P_{i-1}}{1 + \lambda^{-1} u_i P_{i-1} u_i^{*}}\right], P_{-1} = \varepsilon^{-1} I \quad (27)$$

This recursion shows that the update from P_{i-1} to P_i requires only knowledge of the most recent regressor u_i. In this way, at each time instant i, the algorithm only needs to have access to the data $\{w_{i-1}, d(i), u_i, P_{i-1}\}$ in order to determine $\{w_i, P_i\}$. The matrix P_{i-1} essentially summarizes the information from all previous regressors [3].

3 Adaptive Noise Cancellation

The basic idea of an adaptive noise cancellation algorithm is to pass the corrupted signal through a filter that tends to suppress the noise while leaving the signal unchanged. This process is an adaptive process, which means it can not require a priori knowledge of signal or noise characteristics. Adaptive noise cancellation algorithms utilize two or more microphones(sensor). One microphone is used to measure the speech + noise signal while the other is used to measure the noise signal alone. The technique adaptively adjusts a set of filter coefficients so as to remove the noise from the noisy signal. This technique, however, requires that the noise component in the corrupted signal and the noise in the reference channel have high coherence. Unfortunately this is a limiting factor, as the microphones need to be separated in order to prevent the speech being included in the noise reference and thus being removed. With large separations the coherence of the noise is limited and this limits the effectiveness of this technique. In summary, to realize the adaptive noise cancellation, we use two inputs and an adaptive filter. One input is the signal corrupted by noise (Primary Input, which can be expressed as $\mathrm{s(n)} + \mathrm{N}_0(n)$). The other input contains noise related in some way to that in the main input but does not contain anything related to the signal (Noise Reference Input, expressed as $\mathrm{N}_1(n)$). The noise reference input pass through the adaptive filter and output $\mathrm{y(n)}$ is produced as close a replica as possible of $\mathrm{N}_0(\mathrm{n})$. The filter readjusts itself continuously to minimize the error between $\mathrm{N}_0(n)$ and y(n) during this process. Then the output y(n) is subtracted from the primary input to produce the system output $e = s + N_0 - y$, which is the denoised signal. Assume that S, N_0, N_1 and y are statistically stationary and have zero means. Suppose that S is uncorrelated with N_0 and N_1 , but N_1 is correlated with N_0. We can get the following equation of expectations:

$$E[e^2] = E[s^2] + E[(N_0 - y)^2] \qquad (28)$$

When the filter is adjusted so that $E[e^2]$ is minimized, $E[(N_0 - y)^2]$ is also minimized. So the system output can serve as the error signal for the adaptive filter. The adaptive noise cancellation configuration is shown in figure 1. In this setup, we model the signal path from the noise source to primary sensor as an unknown FIR channel W_e. Applying the adaptive filter to reference noise at reference sensor, we then employ an adaptive algorithm to train the adaptive filter to math or estimate the characteristics of unknown channel W_e.

If the estimated characteristics of unknown channel have negligible differences compared to the actual characteristics, we should be able to successfully cancel out the noise component in corrupted signal to obtain the desired signal. Notice that both of the noise signals for this configuration need to be uncorrelated to the signal s(n). In addition, the noise sources must be correlated to each other in some way, preferably equal, to get the best results.

Do to the nature of the error signal, the error signal will never become zero. The error signal should converge to the signal S(n), but not converge to the exact signal. In other words, the difference between the signal s(n) and the error signal e(n) will always be greater than zero. The only option is to minimize the difference between those two signals.

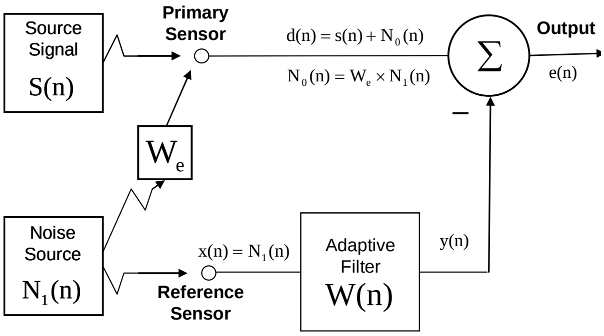

Figure1. Adaptive noise cancellation setup

4 Experimental Results

In this section we assess the performance of each algorithm in noise cancellation setup as shown in figure 1. The LMS, NLMS and RLS are implemented and compared with each other.

The original speech is corrupted with office noise. The SNR of primary signal is -10.2180 dB. This signal is then processed as in Figure 1.

The filter's order is chosen as $M = 10$. The LMS (with $\mu = 0.003$), NLMS (with $\mu = 0.005$), RLS (with $\lambda = 1$) algorithms are applied for adaptation of the filter coefficients. The SNR of the filtered-signal is calculated for each experiment. The SNR Improvement (SNRI) is defined as the final SNR minus the original SNR. The results of the simulation have been shown in Table 1 and figures 2-7. From Table 1, we see that the SNRI of RLS algorithm is higher than LMS and NLMS.

Figure 2 shows the Original signal, Primary Signal, Output of Filter, Mean Square Error for adaptive noise cancellation using LMS algorithm. We have also plotted the time evolution of filter-taps for LMS algorithm in figure 3. Figures 4-5 and figures 6-7 show the output of the adaptive noise cancellation and filter taps using NLMS and RLS algorithms. As we can see the RLS algorithm has good capabilities in convergence of filter taps and canceling the noise of speech in primary input.

Table 1: SNR Improvement in dB

	LMS	***NLMS***	***RLS***
SNRI(dB)	15.2859	16.1742	41.7355

In order to obtain the optimum order of the filter for each algorithm, we changed the order of filter from 1 up to 300 and then calculated SNRI for each order of filter.
Then, we plotted SNRI against the order of filter, that have been shown in figures 5-7.
According to figures5-7, we notice that this figures are similar in all the algorithms. We also understand, the maximum value of SNRI occurs when the value of order is equal to 8 that we name it "Optimum Order".

4 Conclusions

In this paper we have performed and compared classical adaptive filters, such as LMS,NLMS and RLS algorithms, for attenuating noise in speech signals. In each algorithm the time evolution of filter taps, mean square error, and the output of filter are illustrated. Also, the Optimum Order of filter is calculated through experiments. The results show that the RLS algorithm has good convergence capabilities and is numerically robust in compare with LMS and NLMS algorithm.

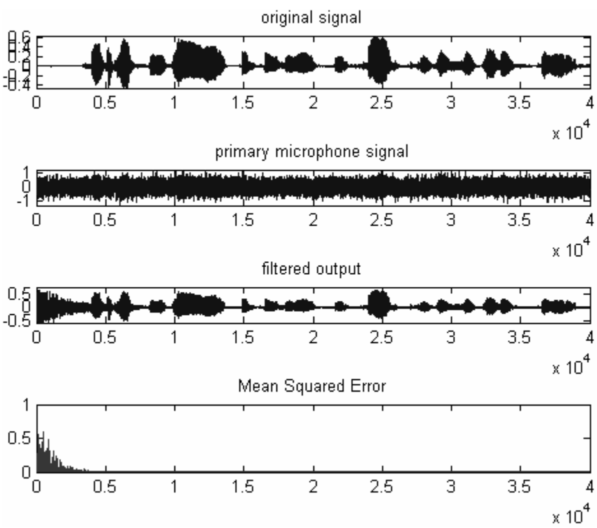

Figure 2. Original , Primary and Filtered Output signal and Mean Square Error in ANC through LMS algorithm.

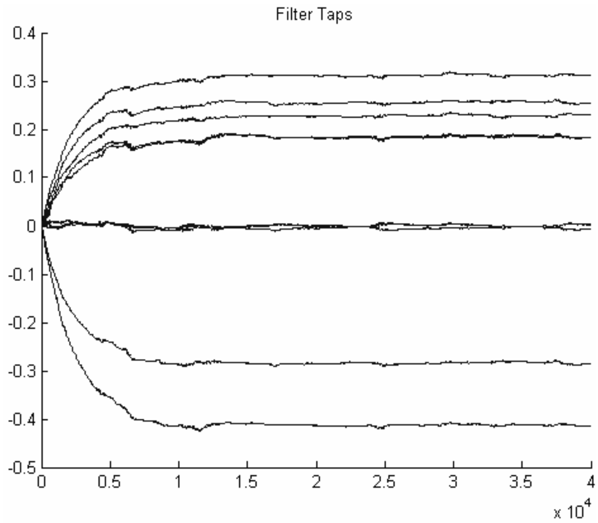

Figure 3. Time evolution of filter taps in ANC through LMS algorithm.

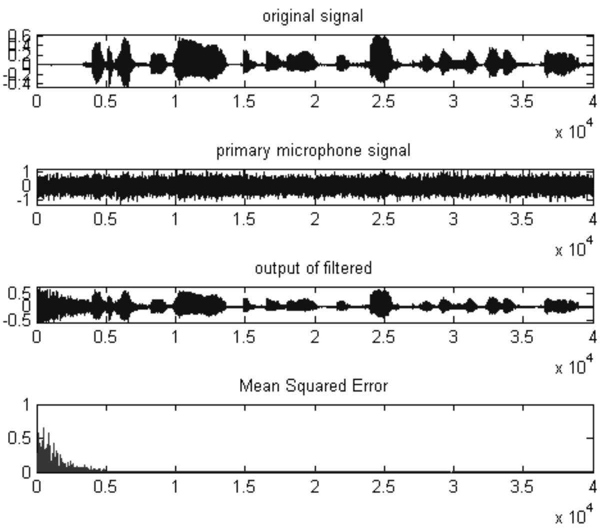

Figure 4. Original , Primary and Filtered Output signal and Mean Square Error in ANC through NLMS algorithm.

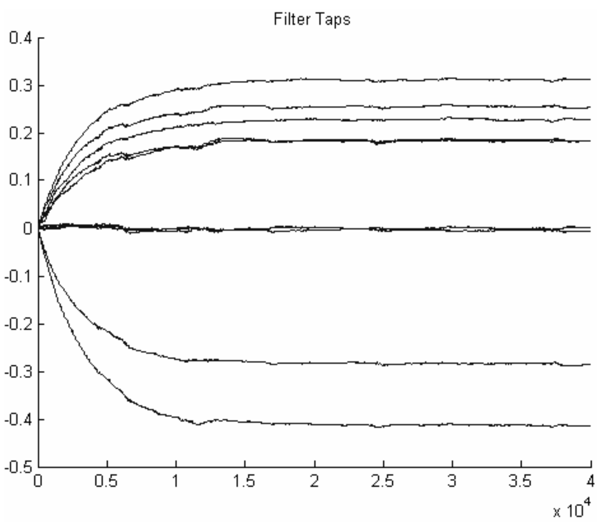

Figure 5. Time evolution of filter taps in ANC through NLMS algorithm.

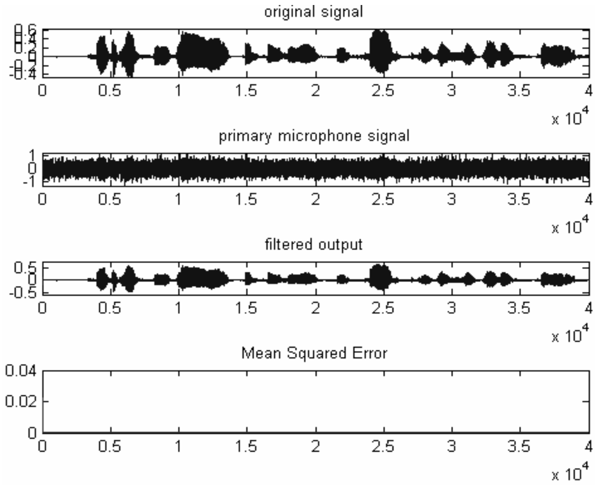

Figure 6. Original , Primary and Filtered Output signal and Mean Square Error in ANC through RLS algorithm.

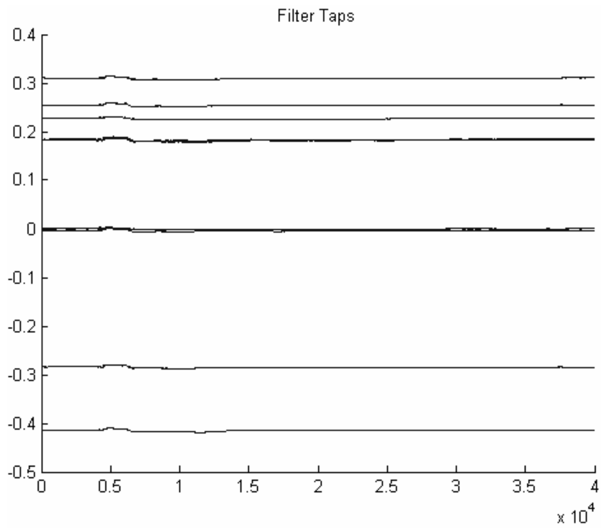

Figure 7. Time evolution of filter taps in ANC through RLS algorithm.

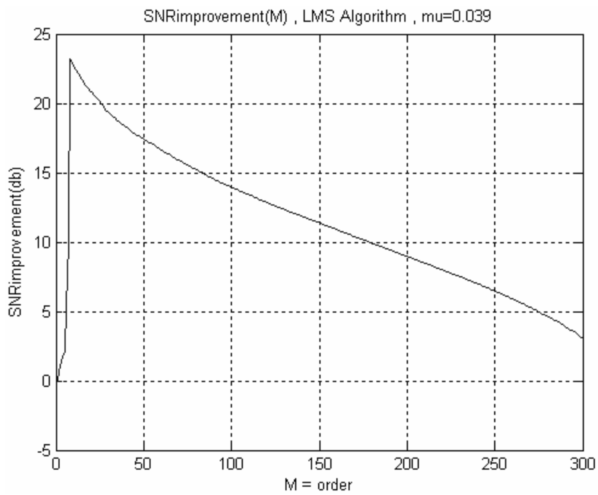

Figure 8. SNRI against order of filter for LMS algorithm.

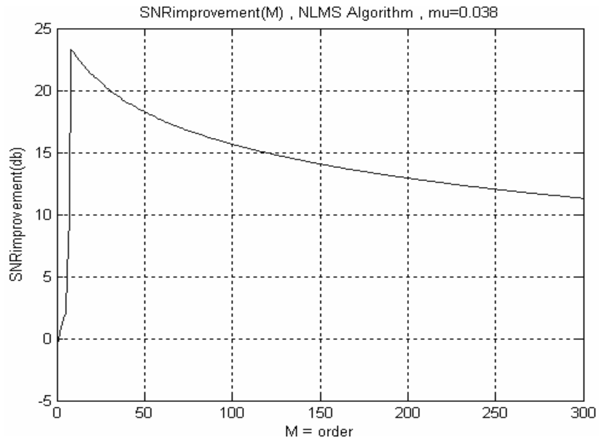

Figure 9. SNRI against order of for NLMS algorithm.

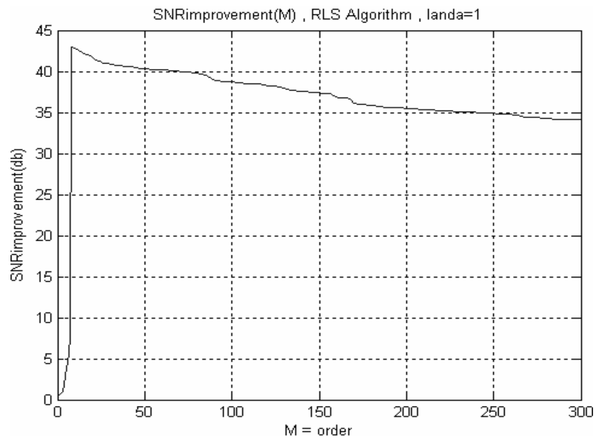

Figure 10. SNRI against order of filter for RLS algorithm.

References

[1] B. widrow, S. Steams, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1985.

[2] S. Haykin, *Adaptive Filter Theory*, 4th ed., Prentice Hall, 2002.

[3] A. H. Sayed, *Fundamentals of Adaptive Filtering*. Wiley, 2003.

[4] S. I. A. Sugiyama, *An adaptive noise canceller with low signal distortion for speech codes*, IEEE Trans. Signal Processing, vol. 47, pp.665-674, Mar. 1999.

[5] W. Harrison, J. Lim, E. Singer, *A new application of adaptive noise cancellation*, IEEE Trans. Acoust., Speech, signal Processing, vol. 34, pp. 21-27, Jan 1986.

[6] J. H. Husøy and M. S. E. Abadi, *A comparative study of some simplified RLS type algorithms*, in Proc. Intl. Symp on Control, Communications and Signal Processing, Hammamet, Tunisia, March 2004, pp. 705-708.