Research Journal of Applied Sciences, Engineering and Technology 4(17): 3102-3107, 2012
ISSN: 2040-7467

Submitted: March 10, 2012 Accepted: March 30, 2012 Published: September 01, 2012

Noise Minimization from Speech Signals using RLS Algorithm with Variable Forgetting Factor

[1]V.K. Gupta, [2]Mahesh Chandra and [3]S.N. Sharan
[1]Department of ECE, IMS Engineering College, Ghaziabad, UP, India-201009
[2]Department of ECE, Birla Institute of Technology, Mesra, Ranchi, Jharkhand, India-835215
[3]Director, Greater Noida Institute of Technology, Greater Noida, U.P, India

Abstract: In this study RLS algorithm with Double Log-Sigmoid function (DSRLS) is proposed to minimize the effect of noise from speech signals. The performance of DSRLS is compared with the performance of RLS and RLS algorithm with Log-Sigmoid function (SRLS). Experiments were performed on noisy data which was prepared by adding machine gun, F16 and speech noise to clean speech samples at -5dB, 0dB, 5dB and 10dB SNR levels. The simulation results show that both SRLS and DSRLS perform better than RLS in terms of SNR improvement. However, DSRLS performs best in terms of SNR improvement with MSE decrement.

Key words: Adaptive filter, DSRLS, Mean Square Error (MSE), Signal to Noise Ratio (SNR), SRLS, VFFRLS

INTRODUCTION

The problem of controlling the noise level in audio signal has been the focus of a tremendous amount of research over the years. Noise minimization is essential for speech signal transmission, processing and reception due to ever growing applications of telephone and cellular communication. In the process of transmission of information with a microphone, through a channel, noise automatically gets embedded to the signal. Noise or interference reduces the perceived quality or intelligibility of the audio signal. Active Noise Cancellation (ANC) (Kuo and Morgan, 1999; Kuo *et al*., 2003) is one such approach that has been proposed for reduction of steady state noise. ANC refers to an electromechanical or electro-acoustic technique of canceling acoustic disturbance to yield a noise-free environment. Wideband active noise control systems often use adaptive filters for noise minimization. Adaptive algorithms play an important role in many diverse applications such as communications, acoustics, speech, radar, sonar, seismology, and biomedical engineering (Kuo *et al*., 2003). There are basically two types of adaptive filter algorithms, Minimum Mean Square Error (MMSE) and Recursive Least Squares (RLS) algorithms (Farhang-Boroujeny, 1999; Haykin, 2002). Most of the other algorithms are evolution of these two basic algorithms. The advantages of Least Mean Square (LMS) algorithm are low computational complexity and simple structure. But its process of convergence is slow and there is a contradiction between convergence rate and the adaptive step size. In addition to, there are a class of algorithms collectively known as the Recursive Least Squares (RLS) algorithm (Eleftheriou and Falconer, 1986; Slock and Kailath, 1991; Ewada, 1994). RLS adaptive algorithm with fast convergence is unlike the LMS algorithm convergence which is sensitive to array of parameters related to the input signal. So RLS adaptive filter algorithms with Variable Forgetting Factors (VFFRLS) are implemented in this study. The use of variable forgetting factor (Ting and Childers, 1990; Leung and So, 2005; Leung and So, 2005; Wang, 2009; Bhotto and Antoniou, 2011) has shown improvements in the performance for non-stationary process.

METHODOLOGY

Adaptive filters: Adaptive filters continuously change their impulse response in order to satisfy the given conditions, and by doing so, change the very characteristic of their response. The aim of an adaptive filter is to calculate the difference between the desired signal and the adaptive filter output. This error signal is fed back into the adaptive filter and its coefficients are changed algorithmically in order to minimize the cost function. When the adaptive filter output approaches to the desired signal then the error signal goes to zero. In this situation the noise signal would be completely cancelled and the far user would not feel any disturbance.

Rls algorithm: RLS algorithms are known for excellent performance when working in time varying environments. This algorithm attempts to minimize the cost function shown in Eq. (1). The parameter λ is an exponential weighting factor that should be chosen in the range

Corresponding Author: V.K. Gupta, Department of ECE, IMS Engineering College, Ghaziabad, UP, India-201009

$0 < \lambda < 1$. This parameter is also called forgetting factor since the information of the distant past has an increasingly negligible effect on the coefficient updating. Hence this algorithm gives more emphasis on recent input samples and tends to forget the past input samples. The cost function is expressed as follows:

$$\xi(n) = \sum_{k=1}^{n} \lambda^{n-k} e_n^2(k) \tag{1}$$

where, $k = 1$ is the time at which the RLS algorithm starts. Unlike the LMS algorithm and its derivatives, the RLS algorithm directly considers the values of previous error estimations. The computational complexity of RLS algorithm is higher than that of LMS algorithm (Ewada, 1994).

The RLS algorithm is derived (Farhang-Boroujeny, 1999) in the following section. First we define $y_n(k)$ as the output of the FIR filter, at n, using the current tap weight vector, and the input vector of a previous time k. The estimation error value $e_n(k)$ is the difference between the desired output value at time k, and the corresponding value of $y_n(k)$. These and other appropriate definitions are expressed in the following equations for k = 1, 2, 3..........n.

$$y_n(k) = w^T(n)x(k) \tag{2}$$

$$e_n(k) = d(k) - y_n(k) \tag{3}$$

where, $d(n) = [d(1), d(2) \ldots\ldots d(n)]^T$

$$y(n) = [y_n(1), y_n(2) \ldots\ldots y_n(n)]^T$$

$$e(n) = [e_n(1), e_n(2) \ldots\ldots e_n(n)]^T$$

$$e(n) = d(n) - y(n) \tag{4}$$

If we define $X(n)$ as the matrix consisting of n previous input column vectors up to the present time then $y(n)$ can also be expressed as the following equation:

$$y(n) = X^T(n)w(n) \tag{5}$$

where, $X(n) = [x(1), x(2), \ldots\ldots x(n)]$

The cost function can also be expressed in matrix form using $\lambda(n)$:

$$\xi(n) = \sum_{k=1}^{n} \lambda^{n-k} e_n^2(k) = e^T(n)\lambda(n)e(n) \tag{6}$$

where, $\lambda(n) = \begin{pmatrix} \lambda^{n-1} & 0\ldots & 0 \\ 0 & \lambda^{n-2}\ddots & 0 \\ 0 & 0\ldots & 1 \end{pmatrix}$

Substituting values from Eq. (4) and (5) in (6), the cost function can be expanded and then reduced as following equation:

$$\begin{aligned}
\xi(n) &= e^T(n)\lambda(n)e(n) \\
&= \begin{Bmatrix} d^T(n)\lambda(n)d(n) - d^T(n)\lambda(n)y(n) \\ - y^T(n)\lambda(n)d(n) + y^T(n)\lambda(n)y(n) \end{Bmatrix} \\
&= \begin{Bmatrix} d^T(n)\lambda(n)d(n) - d^T(n)\lambda(n)(X^T(n)w(n)) \\ -(X^T(n))^T\lambda(n)d(n) \\ +(X^T(n)w(n)^T)\lambda(n)(X^T(n)w(n)) \end{Bmatrix} \\
&= d^T(n)\lambda(n)d(n)2\theta_\lambda^T(n)w(n) + w^T(n)\psi_\lambda(n)w(n)
\end{aligned} \tag{7}$$

where, $\psi_\lambda(n) = X(n)\lambda(n)X^T(n)$
$\theta_\lambda(n) = X(n)\lambda(n)d(n)$

From Here $\Psi_\lambda(n)$ is correlation matrix of tap inputs and $\theta_\lambda(n)$ is cross-correlation between tap inputs and desired response. Then the gradient of the above expression is derived for the cost function with respect to the filter tap weights. By forcing this to zero, the coefficients for the filter, $\overline{w}(n)$ are calculated, which minimises the cost function:

$$\overline{w}(n) = \Psi_\lambda^{-1}(n)\theta_\lambda(n) \tag{8}$$

The matrix $\Psi_\lambda(n)$ in the above equation can be expanded and rearranged in a recursive form; we can then use the special form of the matrix inversion lemma to find an inverse for this matrix, which is required to calculate the tap weight vector update. The vector k(n) is known as the gain vector and is included in order to simplify the calculation.

$$\begin{aligned}
\Psi_\lambda^{-1}(n) &= \lambda\Psi_\lambda^{-1}(n-1) + x(n)x^T(n) \\
&= \lambda^{-1}\Psi_\lambda^{-1}(n-1) - \frac{\lambda^{-2}\Psi_\lambda^{-1}(n-1)x(n)x^T(n)\Psi_\lambda^{-1}(n-1)}{1+\lambda^{-1}x^T(n)\Psi_\lambda^{-1}(n-1)x(n)} \\
&= \lambda^{-1}(\Psi_\lambda^{-1}(n-1) - k(n)x^T(n)\Psi_\lambda^{-1}(n-1))
\end{aligned} \tag{9}$$

where, $k(n) = \dfrac{\lambda^{-1}\Psi_\lambda^{-1}(n-1)x(n)}{1+\lambda^{-1}x^T(n)\Psi_\lambda^{-1}(n-1)x(n)}$ (10)

The vector $\theta_\lambda(n)$ of Eq. (7) can also be expressed in a recursive form as shown in following equation:

$$\theta_\lambda(n) = \lambda\theta_\lambda(n-1) + x(n)d(n) \tag{11}$$

From Eq. (9), (10) and (11), we can fsinally arrive at the filter weight update vector for the RLS algorithm:

$$\overline{w}(n) = \overline{w}(n-1) + k(n)\overline{e}_{n-1} \tag{12}$$

where, $\overline{e}_{n-1}(n) = d(n) - \overline{w}^T(n-1)x(n)$

The use of forgetting factor is intended to ensure that data in the distant path are forgotten in order to afford the possibility of following the statistical variations of the observable data when filter operates in a non-stationary environment. In RLS the forgetting factor remains constant. If the forgetting factor is varied as per the variability of noise in the signal then RLS algorithm may perform better for minimization of error from the signal.

Vffrls: The variable forgetting factor was considered to improve the performance of the RLS algorithm. The forgetting factor increases gradually with the increase of the number of training so that it can enhance the tracking capability which results in decrement in estimated error. Function expression (Wang, 2009) for variable forgetting factor is presented as follows:

$$\lambda = \lambda_{\min} + (1 - \lambda_{\min})VFF \tag{13}$$

where, VFF is variable forgetting factor function used to convert forgetting factor λ from constant to variable value. The VFF functions are log-sigmoid function and double log-sigmoid function. Log-sigmoid function models the "S-shaped" curve which shows the growth of dependent variable *p(t)* which is function of independent variable *t*. The initial stage of growth is approximately exponential, which slow down on the beginning of saturation and stops at maturity. A log-sigmoid function is defined by the formula:

$$p(t) = \frac{1}{1 + \exp(-t)} \tag{14}$$

The proposed double log-sigmoid function is similar to the logistic function with numerous applications. A double log-sigmoid function is defined by the formula:

$$p(t) = \operatorname{sgn}(t-d)\left(1 - \exp\left(-\left(\frac{t-d}{s}\right)^2\right)\right) \tag{15}$$

where, *d* is its center and *s* is the steepness factor.

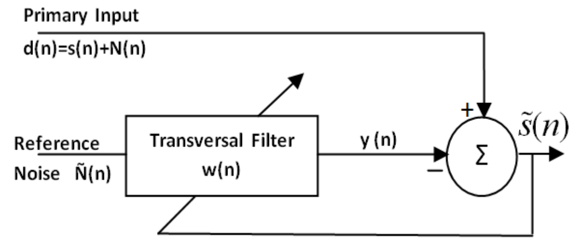

Fig. 1: Block diagram of an adaptive noise cancellation system

The signum function of a real number *x* is defined as follows:

$$\operatorname{sgn}(x) = \begin{cases} 1, x > 0 \\ 0, x = 0 \\ -1, x < 0 \end{cases} \tag{16}$$

It is based on the Gaussian curve and graphically similar to two identical log-sigmoid bonded together at the point $x = d$. Both in log-sigmoid RLS algorithm (SRLS) and Double log-Sigmoid RLS (DSRLS), λ vary as per Eq. (13). The VFF function is taken from Eq. (14) and (15) for SRLS and DSRLS, respectively.

SIMULATION RESULTS AND DISCUSSION

For evaluating the performance of the algorithms, the first requirement is the availability of proper noisy signal. Noisy signal was prepared for Hindi digit shunya by adding machine gun, F16 and speech noises from NOISEX-92 database (Varga *et al*., 1992) to clean Hindi digit shunya signal at -5dB, 0dB, 5dB and 10dB SNR levels. The clean sample of Hindi digit shunya was recorded at 16 kHz sampling frequency and 16 bits resolution with coolwave software in laboratory. The noisy samples are prepared at 16 kHz sampling frequency and 16 bits resolution. The noisy signal of Hindi digit shunya at different SNR levels was fed into the mathematical simulation of RLS and VFFRLS algorithms with the help of MATLAB at different filter orders.

The noisy signal *d(n)* is fed into noise cancellation system shown in Fig. 1. The input *x(n)*in this system is estimated noise calculated from the noisy signal. For estimation of noise from noisy signal it is observed that around 5-10% initial samples of the noisy signal is silence period, where only noise is present. In our case, 900 initial samples (around 9% of complete noisy signal) from the noisy signal of length 10002 samples are taken as noise samples after hearing test. The amount of noise is estimated by repeating the signal of 900 initial samples to get the length of noise equal to the length of noisy signal. The estimated clean signal power at the output (*SP*) and

Table 1:Comparative performance of RLS, SRLS and DSRLS

Filter order		5				10			
Algorithm	Parameter	-5db	0dB	5dB	10dB	-5dB	0dB	5dB	10dB
Machine gun noise									
RLS	SNR	8.89	10.86	13.40	15.66	9.28	11.13	13.68	16.06
	MSE	0.0430	0.0390	0.0350	0.0316	0.0431	0.0392	0.0354	0.0321
SRLS	SNR	15.81	16.53	17.69	18.83	16.72	17.30	18.30	19.36
	MSE	0.0440	0.0404	0.0367	0.0332	0.0442	0.0406	0.0371	0.0339
DSRLS	SNR	19.06	18.86	19.04	19.55	19.47	19.36	19.57	20.02
	MSE	0.0418	0.0373	0.0330	0.0295	0.0418	0.0373	0.0330	0.0296
F16 noise									
RLS	SNR	3.19	5.71	9.271	13.01	3.71	6.18	9.71	13.51
	MSE	0.0428	0.0377	0.0327	0.0299	0.0428	0.0377	0.0326	0.0298
SRLS	SNR	5.38	7.67	11.02	14.64	5.84	8.12	11.47	15.14
	MSE	0.0429	0.038	0.0330	0.0303	0.0428	0.0379	0.0329	0.0302
DSRLS	SNR	5.81	7.93	11.17	14.73	6.12	8.21	11.41	14.99
	MSE	0.0427	0.0376	0.0323	0.0293	0.0427	0.0375	0.0323	0.0293
Speech noise									
RLS	SNR	6.75	8.72	11.69	14.74	6.95	8.94	11.93	15.03
	MSE	0.0422	0.0371	0.0331	0.0312	0.0424	0.0373	0.0333	0.0314
SRLS	SNR	11.50	13.14	15.54	17.83	11.59	13.26	15.70	18.02
	MSE	0.0427	0.0379	0.0341	0.0322	0.0431	0.0383	0.0345	0.0326
DSRLS	SNR	12.54	14.05	16.25	18.34	12.41	13.93	16.14	18.28
	MSE	0.0419	0.0365	0.0324	0.0304	0.0419	0.0365	0.0324	0.0304

estimated noise power from estimated clean signal (*NP*) is calculated and used in Eq. (17) to calculate output SNR. The output SNR is calculated at all input SNR level signals for all three noises.

$$SNR = 10\log10\left({(SP - NP)}/{NP} \right) \tag{17}$$

The value of λ is taken 0.99 in case of RLS and λ_{min} is taken 0.95 in case of SRLS and DSRLS. Then the resulting outputs were analyzed in order to study the behavior of these algorithms. In SRLS, path variation of λ is according to the shape of log-sigmoid function. In DSRLS, path varies as per double log-sigmoid function which follows the variation in noise more closely as compared to SRLS. The performance of algorithms was compared in terms of improvement in SNR and decrement in MSE of the input noisy signal at -5dB, 0dB, 5dB and 10 dB for all three noises as shown in Table 1. As observed from Table 1 that SRLS algorithm has shown improvement in SNR as compared to RLS algorithm

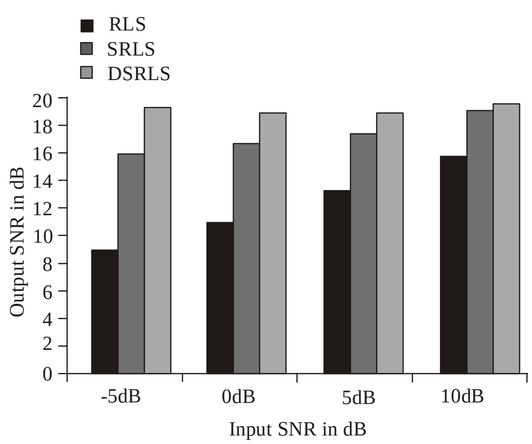

Fig. 2: Comparative analysis of improvements in SNR for machine gun noise at filter order 5

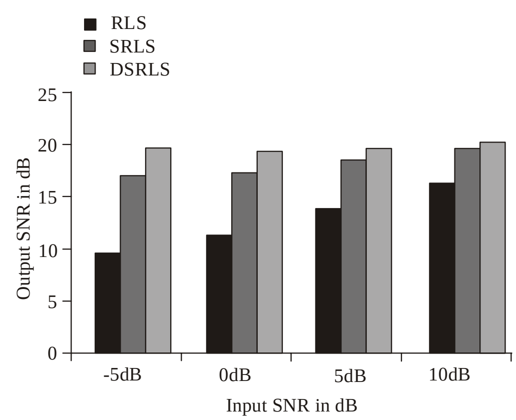

Fig. 3: Comparative analysis of improvements in SNR for machine gun noise at filter order 10

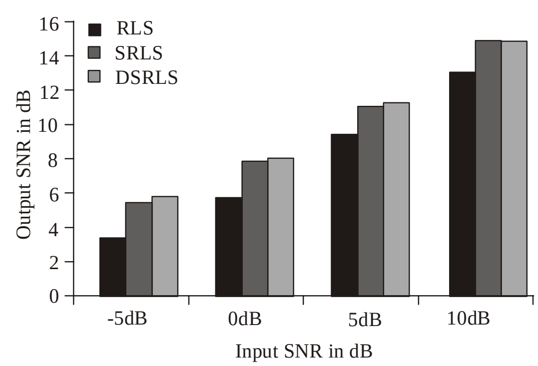

Fig. 4: Comparative analysis of improvements in SNR for F16 noise at filter order 5

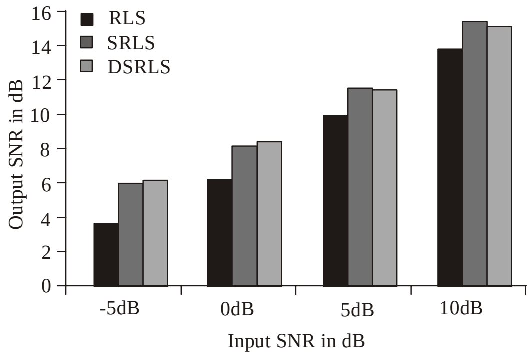

Fig. 5: Comparative analysis of improvements in SNR for F16 noise at filter order 10

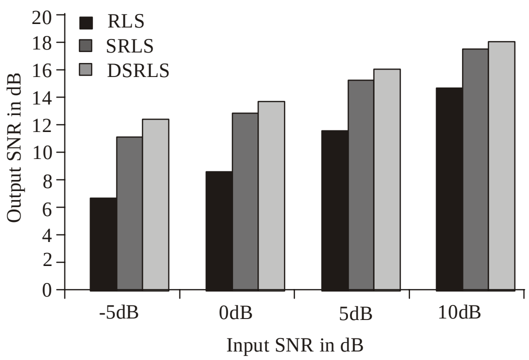

Fig. 6: Comparative analysis of improvements in SNR for speech noise at filter order 5

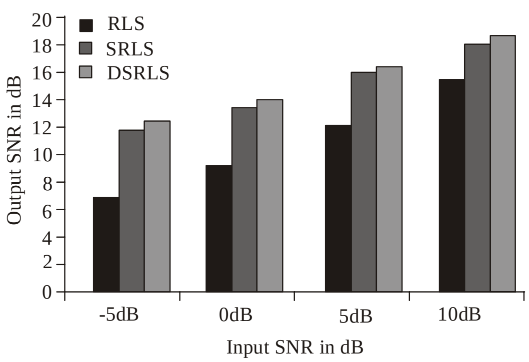

Fig. 7: Comparative analysis of improvements in SNR for speech noise at filter order 10

while DSRLS algorithm performed best among all the algorithms at all filter orders. It was also observed from Table 1 that SRLS algorithm has shown improvement in SNR with an increase in MSE as compared to RLS algorithm at all filter orders for all three noises. This means that noise power also increases with the increase in signal power. The increment in MSE is the drawback of SRLS algorithm. The DSRLS algorithm performed best among all the algorithms in terms of improvement in SNR and decrement in MSE. The comparative performance of RLS, SRLS and DSRLS are shown in Fig. 2 to 9. It is

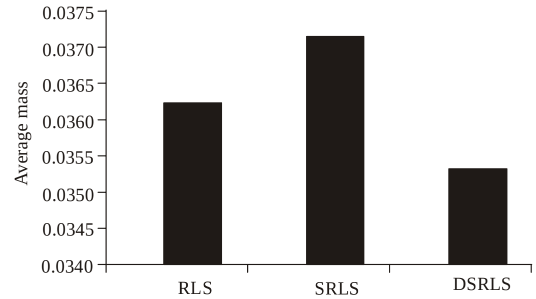

Fig. 8: Average MSE for all three noises at filter order 5

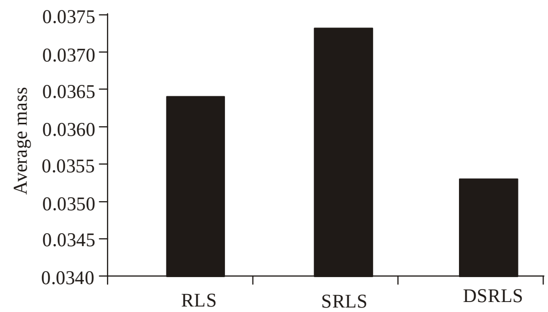

Fig. 9: Average MSE for all three noises at filter order 10

observed from Fig. 2 to 7 that DSRLS performs best among all three algorithms in terms of improvement in SNR except for F16 noise at 5dB and 10dB input SNR levels for filter order 10. Figure 8 and 9 shows average MSE for all three noises at filter order 5 and filter order 10.

The improvement in SNR with SRLS and DSRLS is due to the enhanced tracking capability of algorithms for noise variations. The improvement depends upon the choice of path variations provided to λ in SRLS and DSRLS. As noise level increases the process becomes more non-stationary and more path variation is needed to track the statistical variations of the noisy signal. In such situations DSRLS is able to follow the variations more closely and provides more improvement in SNR and decrement in MSE as compared to both RLS and SRLS algorithms.

CONCLUSION

RLS algorithm with variable forgetting factors has shown its superiority over RLS algorithm for noise minimization. It is due to the fact that the use of variable forgetting factor with RLS is able to track the variations in noisy signal more closely as compared to RLS algorithm. The SRLS algorithm only shows improvement in SNR but not in MSE over RLS. The proposed DSRLS algorithm has shown improvements in both the parameters, SNR and MSE as compared to RLS and

SRLS algorithms. These improvements are obtained due to better enhanced tracking capability for noise variations of DSRLS algorithm.

REFERENCES

Bhotto, M.Z.A. and A. Antoniou, 2011. Robust recursive least-squares adaptive-filtering algorithm for impulsive-noise environments. IEEE Signal Proce. Lett., 18(3): 185-188.

Eleftheriou, E. and D.D. Falconer, 1986. Tracking properties and steady-state performance of RLS adaptive filter algorithms. IEEE T. Acoust. Speech Signal Proce., 34(5): 1097-1109.

Ewada, E., 1994. Comparison of RLS, LMS and sign algorithms for tracking randomly time-varying channels. IEEE T. Signal Process., 42(11): 2937-2944.

Farhang-Boroujeny, B., 1999. Adaptive Filters, Theory and Applications. John Wiley and Sons, New York.

Haykin, S., 2002. Adaptive Filter Theory. 4th Edn., Pearson Education, India.

Kuo, S.M. and D.R. Morgan, 1999. Active noise control: A tutorial review. Proceed. IEEE, 87(6): 943-973.

Kuo, S.M., X. Kong and W.S. Gan, 2003. Applications of adaptive feedback active noise control system. IEEE Trans. Contr. Syst. Technol., 11(2): 216-220.

Leung, S.H. and C.F. So, 2005. Gradient-based variable forgetting factor RLS algorithm in time-varying environments. IEEE T. Signal Process., 53(8): 3141-3150.

Slock, D.T.M. and T. Kailath, 1991. Numerically stable transversal filters for recursive least squares adaptive filtering. IEEE T. Signal Process., 39(1): 92-114.

Ting, Y.T. and D.G. Childers, 1990. Speech Analysis using the Weighted Recursive Least Squares Algorithm with a Variable Forgetting Factor. ICASSP, Albuquerque, NM, 1: 389-392.

Varga, A., H.J.M. Steeneken and D. Jones, 1992. The noisex-92 study on the effect of additive noise on automatic speech recognition system. Reports of NATO Research Study Group (RSG.10).

Wang, J., 2009. A Variable Forgetting Factor Rls Adaptive Filtering Algorithm, International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communication, Beijing, China, pp: 1127-1130.