Audio Processing by Lattice RLS Algorithm Based Linear Adaptive Filter

Amit Prakash, Kumar Shashi Kant
Department of Electronics and Communication Engineering,
National Institute of Technology, Jamshedpur, Jharkhand, INDIA-831014

Abstract

This RLS Lattice algorithm is developed by using vector space treatment with the introduction of Forgetting Factor such that it offers fast convergence and good error performance in the presence of noise. A linear Adaptive filter using this algorithm is developed which has advantage that we can directly add the next stage to the existing previous filter stage without changing the previous filter coefficients to obtain the higher order filter. An audio signal with Gauss White noise is simulated in the noise cancellation system on MATLAB platform. This algorithm has very less computational complexity since it does not include any inverse matrix calculation which is highly efficient. The experimental resultdemonstrates the importance of using newly updated iterations for the sensitivity of the noise estimation algorithm performance by introducing the forgetting factor in this algorithm.

Keywords— Adaptive filters, RLS Algorithm, Adaptive noise cancellation, Vector space

1.Introduction

As compared to other adaptive algorithms Recursive Least Square (RLS) algorithm has rapid and exact convergence with a better noise handling capability across frequencies even when the Eigen value spread of the input signal correlation matrix is large. It extends the conventional scheme by adopting a numerical linear algebra random variable analysis without any mean operator [1]-[3]. A step further, RLS Lattice (RLSL) algorithm based adaptive filter is much more useful in audio processing and noise cancellation since the data processing at any instant of time for *(p+1)th* order requires only to add the new factor with the previous output signals of *pth* order as an input to the next order. Further, the use of numerical linear algebra analysis gives better numerical performance of the algorithm and its stability, such that the poles lies within the unity circle as explained in [4].

To handle the rapid change in the statistical properties, forgetting factor has been introduced in this algorithm. It gives rise to exponentially weighted growing window which further enhances the performance of the algorithm by handling true error covariance matrix. In [5], the adaptive technique for noise reduction during non - speech segment is explained but its performance is degraded during speech segment. The performance of the signal can be improved by using forgetting factor in the algorithm that leads to fast tracking. It exploits the concept of forgetting factor that may be required, due to model uncertainty, presence of unknown external disturbances and time-varying nature of the observed signal or non-stationary behaviour of observation noise [6].

In this paper, a RLS Lattice algorithm has been presented by using vector space treatment with the introduction of forgetting factor, such that it offers fast convergence and good error performance in the presence of noise. This algorithm is better than the other widely used adaptive algorithm, i.e., Least Mean Square (LMS) algorithm for noise cancellation in [7]. In addition, an improved block diagram for Adaptive Noise Canceller (ANC) is introduced to achieve noise cancellation of an audio signal.

2.Description of the Algorithm

2.1. RLS Algorithm

For the exponentially weighted least square method the tap-weight vector $\hat{w}(n)$ at iteration $'n'$ is given by [1] as,

$$\widehat{w}(n) = \emptyset^{-1}(n)z(n) \tag{1}$$

where $\emptyset(n)$ is the correlation matrix of the input vector u(n) given by $\sum_{i=1}^{n} \lambda^{n-i} u(i)u^H(i)$, $z(n)$ is the cross-correlation vector between input of the traversal filter and the desired response d(n) is given by $\sum_{i=1}^{n} \lambda^{n-i} u(i)d^*(i)$.

Here the matrix $\emptyset(n)$ for tap weight update is given by Matrix Inverse Lemma [1], which states that

$$A^{-1} = B - BC(D + C^H BC)^{-1} C^H B$$

where A and B are positive definite M-by-M matrices, C is M-by-N matrix and D is positive definite N-by-N matrix.

Since this algorithm uses inverse calculation of a matrix it has more computational complexity and it is difficult for hardware implementations.

2.2. RLS Lattice (RLSL) Algorithm

We consider, in general the pre - windowed with exponentially weighted least square case, the input samples vector to the microphone be-

$$\bar{x}(n) = \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(n) \end{bmatrix}$$

$$W_{k,p,n} = span\{z^{-k}\bar{x}(n), z^{-(k+1)}\bar{x}(n), \ldots \ldots, z^{-p}\bar{x}(n)\}$$

$W_{k,p,n}$ is the p-dimensional space spanned by the column input vectors up to *n-th* index where $z^{-k}\bar{x}(n)$ is given by

$$z^{-k}\bar{x}(n) = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ x(0) \\ \vdots \\ x(n) \end{bmatrix}$$

$P_{k,p,n}$: Orthogonal projection operator with respect to space $W_{k,p,n}$ such that $P_{1,p,n}d(n) = \sum_{i=1}^{p} c_i \mathrm{z}^{-i}\,\bar{x}(n)$ where c_i is the optimal coefficient which linearly combines the vector on the space $W_{1,p,n}$.

Order update of forward error and backward error prediction:

$f_{p.n}$: *p-th* order forward prediction error operator on the space $W_{1,p,n}$ with n-th index as the current index

Now, $f_{p+1.n}$ is the forward error prediction operator on the space $W_{1,p+1,n}$ i.e.

$$f_{p+1.n} = \bar{x}(n) - P_{1,p+1,n}\bar{x}(n) \tag{2}$$

$W_{1,p+1,n}$ can be orthogonally decomposed into two spaces i.e., $W_{1,p,n}$ and the space spanned by the *(p+1)th*element vector $z^{-(p+1)}\bar{x}(n)$. The orthogonal projection error of the vector $z^{-(p+1)}\bar{x}(n)$ has the form $\begin{bmatrix} 0 \\ \cdots \\ b_{p,n-1} \end{bmatrix}$ denoted by $\bar{b}_{p,n-1}$.

$b_{p,n-1}$ is the *p-th* order backward error prediction operator on the space $W_{1,p,n-1}$ with *(n-1)th* index as the current index.

Hence $f_{p+1.n}$ = sum of the forward error prediction operator on the space $W_{1,p,n}$ and the forward error prediction operator on the space spanned by $z^{-(p+1)}\bar{x}(n)$ which is given by

$$f_{p+1.n} = \bar{x}(n) - \left[P_{1,p,n}\bar{x}(n) + P_{\bar{b}_{p,n-1}}\bar{x}(n)\right]$$

$$= f_{p.n} - \frac{\langle \bar{x}(n), \bar{b}_{p,n-1} \rangle}{\|\bar{b}_{p,n-1}\|^2} \bar{b}_{p,n-1} \tag{3}$$

$\frac{\langle \bar{x}(n), \bar{b}_{p,n-1} \rangle}{\|\bar{b}_{p,n-1}\|^2} \bar{b}_{p,n-1}$ is the projection error of $\bar{x}(n)$ on $\bar{b}_{p,n-1}$.

$\langle \bar{x}(n), \bar{b}_{p,n-1} \rangle$ is the inner product of the vectors $\bar{x}(n)$ *and* $\bar{b}_{p,n-1}$ which gives the correlation between the two vectors denoted by $\Delta_{p,n}$. Thus

$$\Delta_{p,n} = \langle \bar{x}(n), \bar{b}_{p,n-1} \rangle \tag{4}$$

$$\|\bar{b}_{p,n-1}\|^2 = \sigma_{p,n-1}^{b^2} \tag{5}$$

$\sigma_{p,n-1}^{b^2}$ is the backward error variance. There may be chance that this variance becomes zero and therefore this should be initialized with a small positive value in the algorithm.

Now (3) transforms to

$$f_{p+1.n} = f_{p.n} - \frac{\Delta_{p,n}}{\sigma_{p,n-1}^{b^2}} \tilde{b}_{p,n-1} \quad (6)$$

The present component of the vector $f_{p+1.n}$ comes out to be

$$f_{p+1}(n) = f_p(n) - \frac{\Delta_{p,n}}{\sigma_{p,n-1}^{b^2}} b_p(n-1) \quad (7)$$

Similarly, *(p+1)th* order backward error prediction is given by

$$b_{p+1.n} = \tilde{b}_{p,n-1} - \frac{\Delta_{p,n}}{\sigma_{p,n}^{f^2}} f_{p.n} \quad (8)$$

$$\sigma_{p,n}^{f^2} = \left\| f_{p,n} \right\|^2 \quad (9)$$

$\sigma_{p,n}^{f^2}$is the forward error variance and will never approach zero since it is the square of the error which is projected by the present component on the past p-components.

The present component of the vector $b_{p+1.n}$comes out to be

$$b_{p+1}(n) = b_p(n-1) - \frac{\Delta_{p,n}}{\sigma_{p,n}^{f^2}} f_p(n) \quad (10)$$

With $p = 0$ in (6) and (8), and comparing the vectors $f_{0.n}$ *and* $b_{0.n}$ will be

$$f_{0.n} = b_{0.n} = \bar{x}(n) \quad (11)$$

For $p = 0$,the present component or the last component of vectors $f_{p.n}$ *and* $b_{p.n}$ is $f_0(n)$ and $b_0(n)$respectively which from (11) leads to

$$\begin{aligned} f_0(n) &= b_0(n) \\ &= x(n) \end{aligned} \quad (12)$$

For $n \geq p-1$, the structure of RLSL filter follows from (7), (10) and (12) is given in Fig. 1.

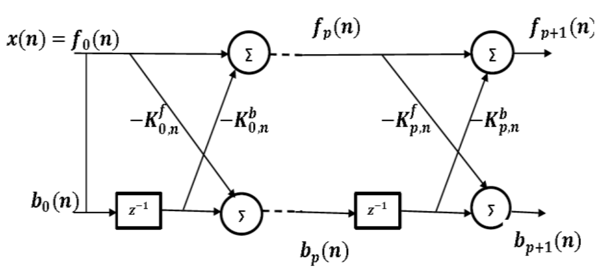


Fig.1 Schematic of RLSL Filter

The values of $\boldsymbol{K_{p,n}^f}$*and* $\boldsymbol{K_{p,n}^b}$ (Fig.1) are both function of order and time that needs to be updated respectively.

$$\boldsymbol{K_{p,n}^f} = \frac{\Delta_{p,n}}{\sigma_{p,n}^{f^2}} \quad (13)$$

and,$\boldsymbol{K_{p,n}^b} = \frac{\Delta_{p,n}}{\sigma_{p,n-1}^{b^2}}$ (14)

From (13) and (14) forward error variance and backward error variance requires time and order update for the recursive values of $\boldsymbol{K_{p,n}^f}$*and* $\boldsymbol{K_{p,n}^b}$respectively.

Time update of variance:

From (5), (9) and (11)

$$\sigma_{0,n}^{f^2} = \sigma_{0,n}^{b^2} = \|\bar{x}(n)\|^2 \quad (15)$$

$\|\bar{x}(n)\|^2$is the auto-correlation matrix of $\bar{x}(n)$ with exponential weighting factor or forgetting factor "λ" defined in pp.437 of [1].

$$\begin{aligned} \|\bar{x}(n)\|^2 &= [\lambda^n x^2(0) + \lambda^{n-1} x^2(1) + \cdots + \lambda x^2(n-1) \\ &\quad + x^2(n)] \\ &= \lambda[\lambda^{n-1} x^2(0) + \lambda^{n-2} x^2(1) + \cdots + x^2(n-1)] \\ &\quad + x^2(n) \end{aligned}$$

Hence,

$$\sigma_{0,n}^{f^2} = \sigma_{0,n}^{b^2} = \lambda \sigma_{0,n-1}^{f^2} + x^2(n) \quad (16)$$

Order update of variance

$$\sigma_{p+1,n}^{f^2} = \left\| f_{p+1,n} \right\|^2$$

$\sigma_{p+1,n}^{f^2}$is the inner product of $f_{p+1,n}$which gives the auto-correlation vector of $f_{p+1,n}$ explained in (17)

$$\begin{aligned} \sigma_{p+1,n}^{f^2} &= \sigma_{p,n}^{f^2} - \frac{\Delta_{p,n}}{\sigma_{p,n-1}^{b^2}} (\Delta_{p,n}) \\ &= \sigma_{p,n}^{f^2} - \frac{\Delta_{p,n}^2}{\sigma_{p,n-1}^{b^2}} \end{aligned} \quad (17)$$

Similarly,

$$\sigma_{p+1,n}^{b^2} = \sigma_{p,n-1}^{b^2} - \frac{\Delta_{p,n}^2}{\sigma_{p,n}^{f^2}} \quad (18)$$

Time update for$\Delta_{p,n}$:

Consider the space $U_{\pi,n}$ such that

$$\begin{aligned} &U_{p,\pi,n} \\ &= span\{z^{-1}\bar{x}(n), z^{-2}\bar{x}(n), \dots \dots, z^{-p}\bar{x}(n), \pi(n)\} \end{aligned}$$

where$\pi(n)$is the pinning vector defined as $[0 \quad 0 \quad \dots \quad 1]^{\mathrm{T}}$

and,

$$U_{p,\pi,n-1} = \left[\begin{bmatrix} z^{-1}\bar{x}(n-1) \\ \dots \\ 0 \end{bmatrix}, \quad \dots \begin{bmatrix} z^{-p}\bar{x}(n-1) \\ \dots \\ 0 \end{bmatrix}, \pi(n) \right]$$

The *p-th* order backward error projection $b_{p,\pi,n}$is given by

$$b_{\pi,n} = \begin{bmatrix} \tilde{b}_{p,n-2} \\ \dots \\ 0 \end{bmatrix} = \bar{b}_{p,n-1} - \frac{b_{p,n-1}}{\vartheta_{p,n}}\rho \qquad (19)$$

$$where, \vartheta_{p,n} = error\ variance\ of\ \pi(n)$$

$$\rho = \ projection\ error\ of\ \pi(n)$$

From (19),

$$\bar{b}_{p,n-1} = \begin{bmatrix} \tilde{b}_{p,n-2} \\ \dots \\ 0 \end{bmatrix} + \frac{b_{p,n-1}}{\vartheta_{p,n}}\rho$$

From (4) and (19),

$$\Delta_{p,n} = \left\langle \bar{x}(n), \begin{bmatrix} \tilde{b}_{p,n-2} \\ \dots \\ 0 \end{bmatrix} + \frac{b_{p,n-1}}{\vartheta_{p,n}}\rho \right\rangle$$

$$= \lambda\langle \bar{x}(n-1), \bar{b}_{p,n-2}\rangle + \frac{f_p(n)b_p(n-1)}{\vartheta_{p,n}}$$

$$= \lambda\Delta_{p,n-1} + \frac{f_p(n)b_p(n-1)}{\vartheta_{p,n}} \qquad (20)$$

The present component or the last component ($\Delta_p(n)$) of vector $\Delta_{p,n}$ can be derived from (19) as

$$\Delta_p(n) = \lambda\Delta_p(n-1) + \frac{f_p(n)b_p(n-1)}{\vartheta_p(n)} \qquad (21)$$

The order update for $\vartheta_p(n)$ which is correlation of the error variance of $\pi(n)$ is given by

$$\vartheta_{p+1}(n) = \vartheta_p(n) + \frac{b_p^2(n-1)}{\sigma_{p,n}^{f\,2}} \qquad (22)$$

The summary of the algorithm is depicted in Table.1.

3. Experimental Results

As compared to other widely used adaptive algorithms RLSL algorithm gives better result as shown in Fig. 2. It shows the curve between Mean Square Error (MSE) and the number of iterations. MSE is the difference between the output signal after filtering and the original input signal.

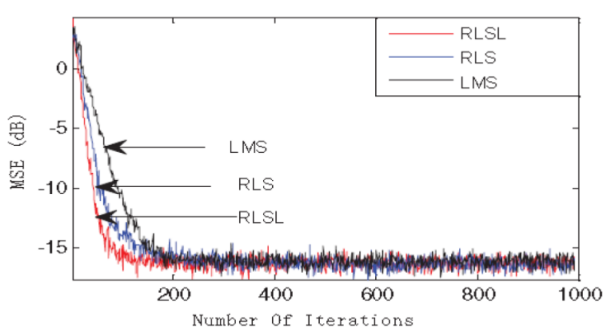


Fig.2 Learning curves of algorithms compared

The figure shows that RLSL has faster convergence rate than the other conventional algorithms due to introduction of forgetting factor and efficient updating of filter weights.

Simulation

The primary input to the ANC is the desired signal x(n) mixed with the White Gaussian noise signal v(n). The mixed signal is passed through the adaptive filter based on RLS Lattice algorithm which recursively adjusts the filter coefficients to get the noise free output y(n) which matches with the x(n) desired signal. The block diagram representation of the ANC is shown in Fig. 3.

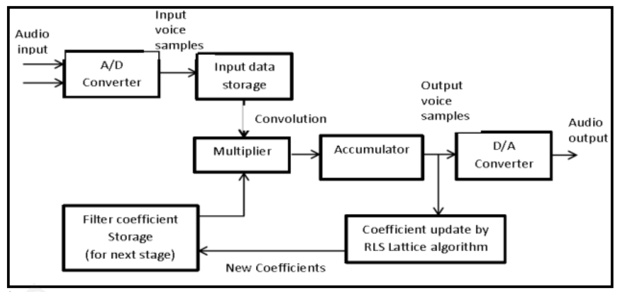


Fig. 3. Block Diagram representation of ANC

The simulation is done through MATLAB platform for sinusoidal signal in first case and for audio signal in second case. In both the cases, the order of filter $p = 8$,and the forgetting factor$\lambda = 0.89$, $\delta = 0.01$.

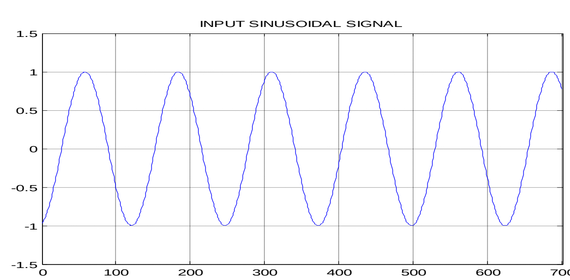
3.1 Simulation of Sinusoidal Signal

*The input signal:*x(n)=sin(0.05**pi*n*)

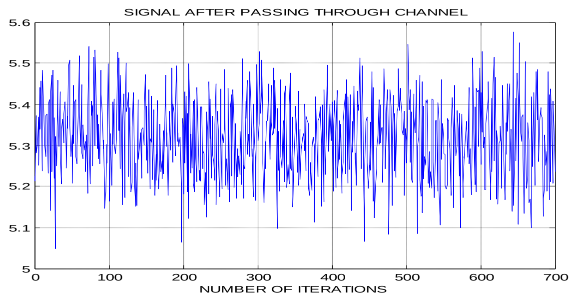
Noise: v(n)=randn(1,n)

Simulation:

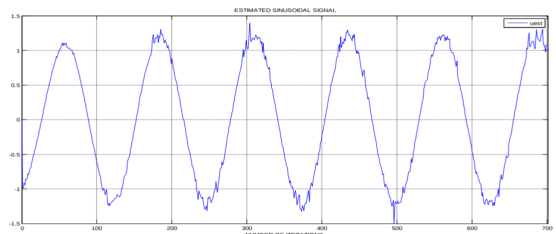
a)



b)



c)



d)

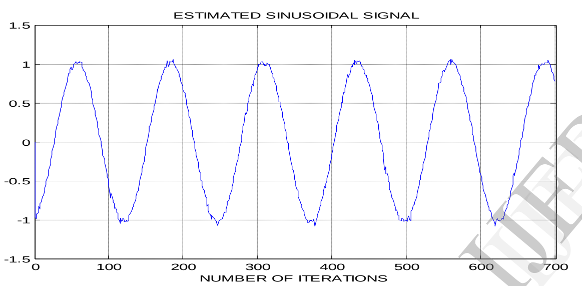


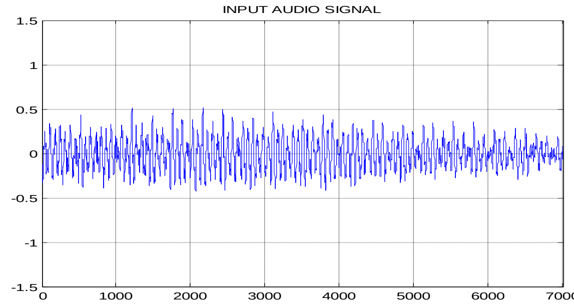
Fig. 4. Simulation of Sinusoidal Signal
(a)input sinusoidal signal (b) input signal mixed with noise (c)output of the RLSL filter (d) output of RLS filter

3.2Simulation of an Audio Signal

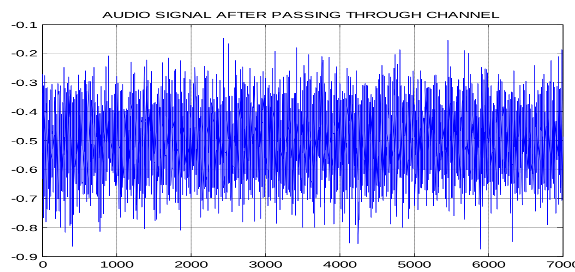
In this experiment an audio file is recorded from the sound card as a *.wav*file. This audio signal is then mixed with the White Gaussian noise and passed through the RLSL algorithm based filter to get the desired output. The RLSL filter updates its coefficients until the noise from the mixed signal is eliminated.

Simulation:

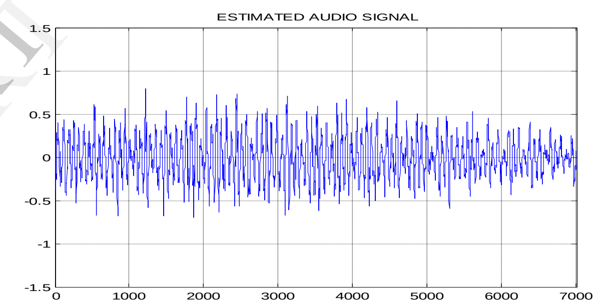
a)



b)



c)



d)

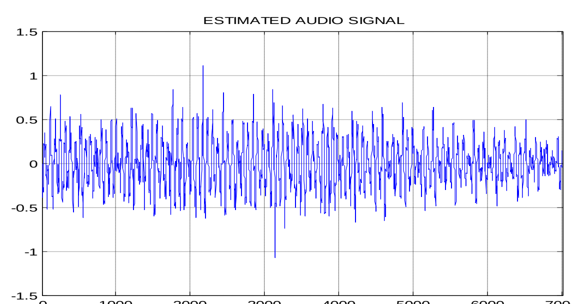


Fig. 5. Simulation of Audio Signal
(a)input audio signal (b) input signal mixed with noise (c)output of the RLS filter (d)output of the RLSL filter

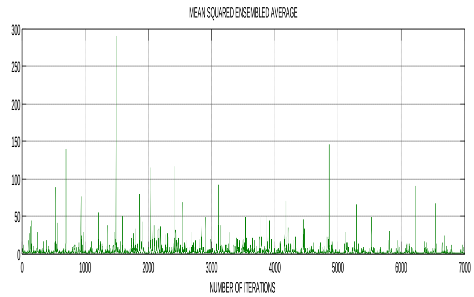


Fig. 6. Mean square Ensemble Average for RLS filter

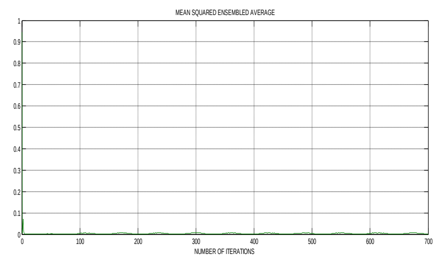


Fig. 7. Mean square Ensemble Average for RLSL filter

The RLSL gives exact convergence as demonsrated in Fig. 7. and hence the output of the RLSL filter is exaclty matched with the desired signal. The SNR(dB) calculatedat input and output of both RLS and RLSLfilters respectively aregiven below.

	Input SNR(dB)	Output SNR(dB)
RLS	−24.1894	18.1987
RLSL	−24.1894	21.4204

4. Conclusion

In this research work, we developed lattice form of RLS Adaptive Filter based on the vector space treatment. The main conclusion is that the standard lattice recursions of Table. 1represents the only numerically reliable algorithm. The algorithm demonstrates the importance of using the newly updated iterations for the sensitivity of the noise estimation algorithm performance by using the forgetting factor in above given algorithm which has been verified by several simulations to validate the algorithm in finite precision and the performance is compared with other algorithms.

Acknowledgement

The authors are thankful to the incessant support and motivation from the Department of Electronics and Communication Engineering at NIT Jamshedpur. The contribution and guidance provided by the Head & supervisor needs a special mention.

Initialization

For $p = 0 \; to \; P$ set:

$$\Delta_p(-1) = 0$$
$$b_p(-1) = 0$$
$$\sigma^{f^2}_{p,-1} = 0$$

$\sigma^{b^2}_{p,-1} = \delta$; such that $\delta > 0$ & $\delta \approx 0$

For $n \geq 0$, repeat:

$$f_0(n) = b_0(n) = x(n)$$
$$\sigma^{f^2}_{0,n} = \sigma^{b^2}_{0,n} = \lambda\sigma^{f^2}_{0,n-1} + x^2(n)$$
$$\vartheta_0(n) = 1$$

For $p = 0 \; to \; P - 1$, repeat:

$$\Delta_p(n) = \lambda\Delta_p(n-1) + \frac{f_p(n)b_p(n-1)}{\vartheta_p(n)}$$
$$f_{p+1}(n) = f_p(n) - \frac{\Delta_{p,n}}{\sigma^{b^2}_{p,n-1}} b_p(n-1)$$
$$b_{p+1}(n) = b_p(n-1) - \frac{\Delta_{p,n}}{\sigma^{f^2}_{p,n}} f_p(n)$$
$$\vartheta_{p+1}(n) = \vartheta_p(n) + \frac{b_p^2(n-1)}{\sigma^{f^2}_{p,n}}$$
$$\sigma^{f^2}_{p+1,n} = \sigma^{f^2}_{p,n} - \frac{\Delta^2_{p,n}}{\sigma^{b^2}_{p,n-1}}$$
$$\sigma^{b^2}_{p+1,n} = \sigma^{b^2}_{p,n-1} - \frac{\Delta^2_{p,n}}{\sigma^{f^2}_{p,n}}$$

Table 1: The standard RLS Lattice Algorithm

References

[1] Simon Haykin, “Order-recursive adaptive filters,” in *Adaptive Fiter Theory*, 4th ed., India, pp. 536-563.

[2] B.Farhang-Boroujeny, “Lattice filters” in *Adaptive Filters*, *USA*, pp. 357-410.

[3] Glenn E. Johnson, Robert **A.** Muir, Joseph *M.* Scanlan, and William M. Steedly, “Practical Comparison of Adaptive Filter Algorithms” Proc. IEEE, pp. 259-263, 1995.

[4] J.R. Bunch, R.C.Le Borne and I.K. Proudler, “Tracking ill-conditioning for the RLS-lattice

algorithms" IEE Proc.-Vis. Image Signal Process., Vol. 145, pp. 1-5 , February 1998.

[5] R. B. Wallace and R. A. Goubran, "Noise Cancellation Using Parallel Adaptive Filters" IEEE transactions on circuits and systems-II: Analog and Digital signal processing, vol. 39, pp. 239-242, April 1992.

[6] Md. Zulfiquar Ali Bhotto and Andreas Antoniou "Robust Recursive Least-Squares Adaptive-Filtering for Impulsive- Noise Environments" IEEE Signal processing letters, vol. 18, pp. 185-188, March 2011.

[7] Ma Shengqian, XuGuowei, Ma Zhifeng, Wei Shuping, Fan Manhong, "Research on Adaptive Noise Canceller of an Improvement LMS algorithm" Proc. IEEE, pp. 1611-1614, 2011.