Recursive Inverse Adaptive Filtering Algorithm

Mohammad Shukri Ahmad, Osman Kukrer and Aykut Hocanin

***Abstract*—In this paper, a new FIR adaptive filtering algorithm is introduced. This algorithm is based on the Quasi-Newton (QN) optimization algorithm. The approach uses a variable step-size in the coefficient update equation that leads to an improved performance. The simulation results show that the algorithm has very similar performance to the Robust Recursive Least Squares Algorithm (RRLS) while performing better than the Transform Domain LMS with Variable Step-Size (TDVSS) in stationary environments. The algorithm is tested in Additive White Gaussian Noise (AWGN) and Correlated Noise environments.**

***Index Terms*—Adaptive Filters, Recursive Inverse, RRLS, TD-VSS.**

I. Introduction

Adaptive filtering techniques are very well-known topics in signal processing [1], [2], [3]. The RLS algorithm is one of the most popular techniques [4], [5], [6]. It offers superior speed of convergence compared to LMS algorithm and its variations, especially in highly correlated environments. The main obstacle in using the RLS algorithm is its cost (i.e. it is computationally complex). Variations of the RLS algorithm has been proposed in the literature, but the complexity of these variations still relatively high.

In this paper we propose a new recursive algorithm where its performance in terms of convergence speed, and mean-Square-Error (MSE) is very comparable to the RLS algorithm in one hand, and it is less computationally complex on the other hand.

II. Recursive Inverse Algorithm

The Wiener-Hopf equation leads to the optimum solution for the FIR filter coefficients. This equation is;

$$\mathbf{w}(k) = \mathbf{R}^{-1}(k)\mathbf{p}(k), \tag{1}$$

where $k = 1, 2, \ldots, N-1$, N is the filter length, $\mathbf{w}(k)$ is the filter weight vector calculated at time k, $\mathbf{R}(k)$ is the estimate of the autocorrelation matrix of the filter's input, and $\mathbf{p}(k)$ is the estimate of the cross-correlation vector between the filter's input and the filter's desired output.

The solution of (1) is required at each iteration where the filter coefficients are updated. Additionally, the autocorrelation matrix should be nonsingular at each iteration, [7].

Reconsidering (1) where the correlations are estimated recursively as [**?**],

$$\mathbf{R}(k) = \beta\mathbf{R}(k-1) + \mathbf{x}(k)\mathbf{x}^T(k), \tag{2}$$

$$\mathbf{p}(k) = \beta\mathbf{p}(k-1) + d(k)\mathbf{x}(k), \tag{3}$$

where β is the forgetting factor and it is usually very close to one. Now, substituting (2) and (3) in (1) yields,

$$\mathbf{w}(k) = \{\beta\mathbf{R}(k-1)+\mathbf{x}(k)\mathbf{x}^T(k)\}^{-1}[\beta\mathbf{p}(k-1)+d(k)\mathbf{x}(k)], \tag{4}$$

by using the matrix inversion lemma [8], equation (4) becomes;

$$\begin{aligned}\mathbf{w}(k) &= \frac{1}{\beta}\left[\mathbf{R}^{-1}(k-1)\right] \\ &- \frac{1}{\beta}\left[\frac{\mathbf{R}^{-1}(k-1)\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{R}^{-1}(k-1)}{\beta+\mathbf{x}^T(k)\mathbf{R}^{-1}(k-1)\mathbf{x}(k)}\right](\beta\mathbf{p}(k-1) \\ &+ d(k)\mathbf{x}(k)) \\ &= \left[\mathbf{I}-\frac{\mathbf{R}^{-1}(k-1)\mathbf{x}(k)\mathbf{x}^T(k)}{\beta+\mathbf{x}^T(k)\mathbf{R}^{-1}(k-1)\mathbf{x}(k)}\right]\mathbf{w}(k-1) \\ &+ \frac{1}{\beta}\left[\frac{\mathbf{I}\{\beta+\mathbf{x}^T(k)\mathbf{R}^{-1}(k-1)\mathbf{x}(k)\}\mathbf{R}^{-1}(k-1)\mathbf{x}(k)}{\beta+\mathbf{x}^T(k)\mathbf{R}^{-1}(k-1)\mathbf{x}(k)}\right]d(k) \\ &+ \frac{1}{\beta}\left[\frac{\mathbf{R}^{-1}(k-1)\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{R}^{-1}(k-1)\mathbf{x}(k)}{\beta+\mathbf{x}^T(k)\mathbf{R}^{-1}(k-1)\mathbf{x}(k)}\right]d(k)\end{aligned} \tag{5}$$

rearranging (5);

$$\begin{aligned}\mathbf{w}(k) &= \left[\mathbf{I}-\frac{\mathbf{R}^{-1}(k-1)\mathbf{x}(k)\mathbf{x}^T(k)}{\beta+\mathbf{x}^T(k)\mathbf{R}^{-1}(k-1)\mathbf{x}(k)}\right]\mathbf{w}(k-1) \\ &+ \left[\frac{\mathbf{R}^{-1}(k-1)\mathbf{x}(k)}{\beta+\mathbf{x}^T(k)\mathbf{R}^{-1}(k-1)\mathbf{x}(k)}\right]d(k) \\ &= \mathbf{w}(k-1)+\mu(k)\mathbf{R}^{-1}(k-1)\mathbf{x}(k)e(k)\end{aligned} \tag{6}$$

In (6) the a-priori filtering error is,

$$e(k) = d(k) - \mathbf{x}^T(k)\mathbf{w}(k-1)$$

and

$$\mu(k) = \frac{1}{\beta+\mathbf{x}^T(k)\mathbf{R}^{-1}(k-1)\mathbf{x}(k)}$$

Newton-LMS is equivalent to the Wiener solution with exponential-forgetting window estimation of the autocorrelation and cross-correlation.

The RRLS algorithm is similar, except that the updating of the correlations are not performed directly using (2) and (3). Instead, the inverse autocorrelation matrix is updated to avoid inverting it at each step.

Consider solving the Wiener equation (1) iteratively at each time step k. Specifically, the following iteration converges to the Wiener Solution,

$$w_{n+1}(k) = [I-\mu R(k)]w_n(k)+\mu p(k), \quad n = 0, 1, 2, \ldots \tag{7}$$

if μ satisfies the convergence criterion;

$$\mu < \frac{2}{\lambda_{max}(\mathbf{R}(k))}. \tag{8}$$

by considering the update equations for the correlations, and taking the expectation of (2);

$$\bar{\mathbf{R}}(k+1) = \beta\bar{\mathbf{R}}(k) + \mathbf{R}_{xx}, \tag{9}$$

where $\mathbf{R}_{xx} = E\left\{\mathbf{x}(k)\mathbf{x}^T(k)\right\}$ and $\bar{\mathbf{R}}(k) = E\{\mathbf{R}(k)\}$. Solving (8) yields,

$$\bar{\mathbf{R}}(k) = \frac{1-\beta^k}{1-\beta}\mathbf{R}_{xx}, \tag{10}$$

and as $k \to \infty$

$$\bar{\mathbf{R}}(\infty) = \frac{1}{1-\beta}\mathbf{R}_{xx}, \tag{11}$$

Equation (10) implies that the eigenvalues of the estimated autocorrelation matrix increase exponentially, and in the limit become $\frac{1}{1-\beta}$ times that of the original matrix. The implication is that since μ must be chosen to satisfy (8) in the limit as well, we get:

$$\mu < \frac{2(1-\beta)}{\lambda_{max}(\mathbf{R}_{xx})}, \tag{12}$$

Equation (12) restricts μ to values much smaller than the one required if the autocorrelation matrix $\mathbf{R}_{xx}$ were available. Hence, it would be advantageous to make the step-size μ variable so that,

$$\mu(k) < \frac{2}{\lambda_{max}(\mathbf{R}(k))} = \left(\frac{1-\beta}{1-\beta^k}\right)\left(\frac{2}{\lambda_{max}(\mathbf{R}_{xx})}\right) = \frac{\mu_{max}}{1-\beta^k}, \tag{13}$$

or

$$\mu(k) = \frac{\mu_0}{1-\beta^k} \qquad \text{where } \mu_0 < \mu_{max}, \tag{14}$$

The iteration in (7) has high computational cost. Therefore, with the variable step-size, only one iteration at each time step may be sufficient. The weight update equation becomes;

$$\mathbf{w}(k) = [\mathbf{I} - \mu(k)\mathbf{R}(k)]\mathbf{w}(k-1) + \mu(k)\mathbf{p}(k), \tag{15}$$

Equation(15) describes the update equation of the suggested Recursive Inverse (RI) algorithm. The RI algorithm has a major advantage compared to the RRLS that it does not require the updating of the inverse autocorrelation matrix where the RRLS is known to suffer from this. Also, the positive definiteness property of the inverse autocorrelation matrix may be lost as the iterations advance.

III. Simulation Results

In the simulations, we are comparing the performance of RI algorithm vs. the RRLS [9] and TDVSS [10] algorithms in the system identification problem described in [10] in terms of mse. All the algorithms were implemented using a filter length of $N = 16$ taps.

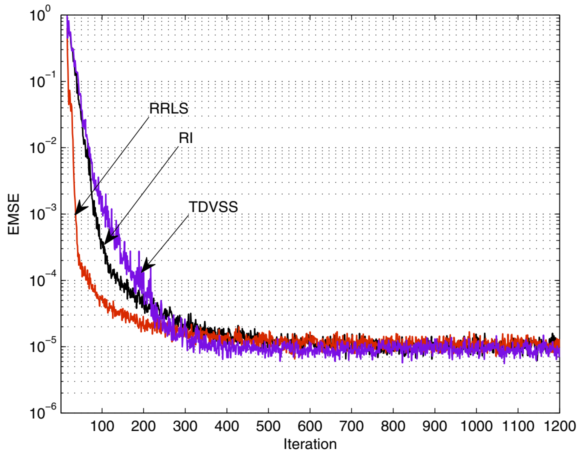

Fig. 1. The ensemble MSE for RI, RRLS and TDVSS, in AWGN, $N = 16$. RI: SNR= 37dB, $\beta = 0.991$, $\mu_0 = 0.00146$. RRLS: SNR= 37dB, $\beta = 0.991$. TDVSS: SNR= 47dB, $\alpha = 0.99$, $\beta = 0.9$, $\epsilon = 0.025$, $\mu_{min} = 0.0047$, $\mu_{max} = 0.05$, $\gamma = 0.001$, $L = 10$.

A. Additive White Gaussian Noise

In this experiment, the mse is held constant, the signal is assumed to be corrupted with an additive white Gaussian noise (AWGN) process. Simulations were done with the following parameters: For the RI algorithm, SNR= 37dB, $\beta = 0.991$, $\mu_0 = 0.00146$. For the RRLS algorithm, SNR= 37dB, $\beta = 0.991$. For the TDVSS algorithm, SNR= 37dB, $\alpha = 0.99$, $\beta = 0.9$, $\epsilon = 0.025$, $\mu_{min} = 0.0047$, $\mu_{max} = 0.05$, $\gamma = 0.001$, $L = 10$. Fig.1 shows that, both the RI and RRLS algorithms converge to the same mse at the same SNR= 37dB, whereas the TDVSS converges to same mse at SNR= 47dB. Even though the RRLS and the proposed algorithm have similar performances in AWGN, the computational complexity of the proposed algorithm is much lower.

B. Additive Correlated Gaussian Noise

In order to test the convergence rate of the algorithms mentioned above, the mse is kept constant and the signal $x(n)$ is assumed to be corrupted by an ACGN. A correlated Gaussian noise process is generated by using the first-order autoregressive model (AR(1)) $v(k+1) = \rho v(k) + v_0(k)$, where $v_0(k)$ is a white Gaussian noise process and ρ is the correlation parameter ($\rho = 0.7$). Simulations were done with the following parameters: For the RI algorithm, SNR= 30dB, $\beta = 0.991$, $\mu_0 = 0.00146$. For the RRLS algorithm, SNR= 30dB, $\beta = 0.991$. For the TDVSS algorithm, SNR= 40dB, $\alpha = 0.99$, $\beta = 0.9$, $\epsilon = 0.025$, $\mu_{min} = 0.0047$, $\mu_{max} = 0.05$, $\gamma = 0.001$, $L = 10$. Fig.2 shows that both the RI and RRLS algorithms converge to the same mse at the same SNR, whereas the TDVSS converges to same mse at a much higher SNR. The advantage of the RI over the TDVSS algorithm in terms of mse is clear.Even though the RRLS and the proposed algorithm have similar performances, the computational complexity of the proposed algorithm is much lower.

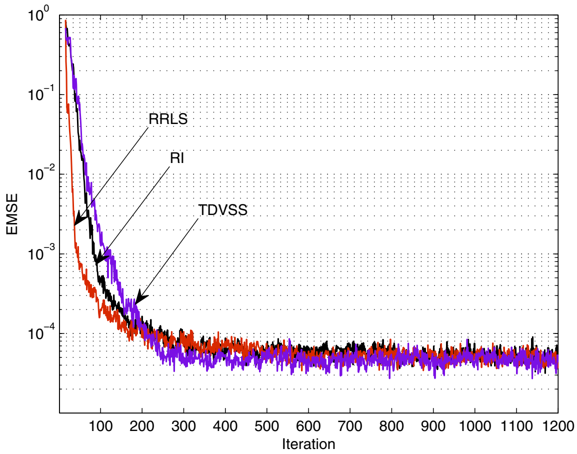

Fig. 2. The ensemble MSE for RI, RRLS and TDVSS, in ACGN, $N = 16$. RI: SNR= 30dB, $\beta = 0.991$, $\mu_0 = 0.00146$. RRLS: SNR= 30dB, $\beta = 0.991$. TDVSS: SNR= 40dB, $\alpha = 0.99$, $\beta = 0.9$, $\epsilon = 0.025$, $\mu_{min} = 0.0047$, $\mu_{max} = 0.05$, $\gamma = 0.001$, $L = 10$.

IV. Conclusion

A new FIR adaptive filtering algorithm is introduced. The approach uses a variable step-size in the coefficient update equation. We have compared the performance of the RI, RRLS and TDVSS algorithms in both AWGN and ACGN environments. Under the same conditions the RI algorithm performs very similar to the RRLS algorithm but with much less computational complexity. On the other hand, the RI algorithm outperforms the TDVSS algorithm in both AWGN and ACGN environments.

References

[1] S. Zhao, Z. Man, S. Khoo and H. R. Wu, “Stability and Convergence Analysis of Transform-Domain LMS Adaptive Filters With Second-Order Autoregressive Process”, *IEEE Transactions on Signal Processings*, vol. 57, no. 1, pp. 119 - 130, January. 2009.

[2] L. Rugini and G. Leus, “Basis Expansion Adaptive Filters for Time-Varying System Identification”, *IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, pp. 153 - 156, 12-14 December. 2007.

[3] B. Hassibi and T. Kailath, “Adaptive filtering with a H^{∞} criterion”, *Asilomar Conference on Signals, Systems and Computers*, vol.2, pp. 1483 - 1487, 31 October-2 November. 1994.

[4] S. Qiao, “Fast adaptive RLS algorithms: a generalized inverse approach and analysis”, *IEEE Transactions on Signal Processings*, vol. 39, no. 6, pp. 1455 - 1459, June. 1991.

[5] S. Haykin, A. H. Sayed, J. R. Zeidler, P. Yee and P. C. Wei, “Adaptive tracking of linear time-variant systems by extended RLS algorithms”, *IEEE Transactions on Signal Processings*, vol. 45, no. 5, pp. 1118 - 1128, May. 1997.

[6] S. Makino and Y. Kaneda, “A new RLS algorithm based on the variation characteristics of a room impulse response”, *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 373 - 376, 19-22 April. 1994.

[7] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, Upper Saddle River, NJ, 2002.

[8] D. J. Tylavsky and G. R. L. Sohie, “Generalization of the Matrix Inversion Lemma”, *Proceedings of the IEEE*, vol. 74, no. 7, pp. 1050-1052, July. 1986.

[9] M. M. Chansarkar and U. B. Desai, “A Robust Recursive Least Squares Algorithm”, *IEEE Transactions on Signal Processings*, vol. 45, no. 7, pp. 1726-1735, July. 1997.

[10] R. C. Bilcu, P. Kuosmanen and K. Egiazarian, “A Transform Domain LMS Adaptive Filter with Variable Step-Size”, *IEEE Signal Processing Letters*, vol. 9, no. 2, pp. 51-53, February. 2002.