An Efficient Recursive Inverse Adaptive Filtering Algorithm for Channel Equalization

Mohammad Shukri Ahmad[1,2], Osman Kukrer[1] and Aykut Hocanin[1]
[1] Department of Electrical and Electronic Engineering,
Eastern Mediterranean University,
Gazimagusa, via Mersin 10, North Cyprus.
Email: {mohammad.ahmad, osman.kukrer, aykut.hocanin}@emu.edu.tr

***Abstract*—Recursive Inverse (RI) adaptive filtering algorithm which uses a variable step-size and the instantaneous value of the autocorrelation matrix in the coefficient update equation was proposed in [1]. The algorithm was shown to have a higher performance compared with the RLS and RRLS algorithms. In this paper, a more efficient version with lower computational complexity is presented. The performance of the algorithm has been tested in a channel equalization setting and compared with those of the Recursive Least Squares (RLS) and Stabilized Fast Transversal Recursive Least Squares (SFTRLS) algorithms in Additive White Gaussian Noise (AWGN), Additive Correlated Gaussian Noise (ACGN), Additive White Impulsive Noise (AWIN) and Additive Correlated Impulsive Noise (ACIN) environments. Simulation results show that the Fast RI algorithm performs better than RLS and requires less computations. Additionally, the performance of the Fast RI algorithm is shown to be superior to that of the SFTRLS algorithm under the same conditions.**

***Index Terms*—Recursive Inverse, RLS, SFTRLS.**

I. Introduction

Adaptive filtering is one of the well-established topics in signal processing [2], [3]. The RLS algorithm [4], [5], [6] offers superior speed of convergence compared to the LMS algorithm and its variants, especially in highly correlated and nonstationary environments. In this paper, we introduce a fast implementation method of the recently proposed RI algorithm, [1], and compare its performance with that of the RLS and SFTRLS algorithms.

II. The Recursive Inverse Algorithm

The Wiener-Hopf equation [2] leads to the optimum solution for the FIR filter coefficients. The coefficients are given by

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

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 filter input autocorrelation matrix, and $\mathbf{p}(k)$ is the estimate of the cross-correlation vector between the filter's input and the filter's desired response. The solution of (1) is required at each iteration where the filter coefficients are updated. As an additional requirement, the autocorrelation matrix should be nonsingular at each iteration, [2].

Reconsidering (1) where the correlations are estimated recursively [3] as;

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

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

and β is the forgetting factor which is usually very close to one. Substituting (2) and (3) in (1) and by using the matrix inversion lemma [7] yields,

$$\mathbf{w}(k) = \mathbf{w}(k-1) + \mu(k)\mathbf{R}^{-1}(k-1)\mathbf{x}(k)e(k) \quad (4)$$

The update equation in (4) is the Newton-LMS algorithm where the a-priori filtering error is,

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

and the variable step-size is

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

Newton-LMS is equivalent to the Wiener solution with exponential-forgetting window estimation of the autocorrelation and cross-correlation. Newton-LMS requires the inverse of the autocorrelation matrix. But in the RI algorithm, this is avoided.

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

$$\mathbf{w}_{n+1}(k) = [\mathbf{I} - \mu\mathbf{R}(k)]\mathbf{w}_n(k) + \mu\mathbf{p}(k), \quad n = 0, 1, 2, \ldots \quad (5)$$

if μ satisfies the convergence criterion [2]

$$\mu < \frac{2}{\lambda_{max}(\mathbf{R}(k))}. \quad (6)$$

[2] Mohammad Shukri Ahmad,
Electrical and Electronic Engineering Department,
European University of Lefke,
Gemikonagi, Lefke, via Mersin 10, North Cyprus.
Email: msalman@eul.edu.tr

By solving the update equation for the autocorrelation matrix, and taking the expectation of $\mathbf{R}(k)$ in (2) as $k \to \infty$, and substituting the result in (6) we get;

$$\mu(k) < \frac{2}{\lambda_{max}(\bar{\mathbf{R}}(k))} = \left(\frac{1-\beta}{1-\beta^k}\right)\left(\frac{2}{\lambda_{max}(\mathbf{R}_{xx})}\right) = \frac{\mu_{max}}{1-\beta^k}, \quad (7)$$

where $\mathbf{R}_{xx} = E\{\mathbf{x}(k)\mathbf{x}^T(k)\}$ and $\bar{\mathbf{R}}(k) = E\{\mathbf{R}(k)\}$. Or equivalently,

$$\mu(k) = \frac{\mu_0}{1-\beta^k} \qquad \text{where } \mu_0 < \mu_{max}. \quad (8)$$

The iteration in (5) has a high computational cost. Therefore, with the variable step-size, only one iteration at each time step may be sufficient. Finally, the weight update equation for the proposed RI algorithm becomes:

$$\mathbf{w}(k) = [\mathbf{I} - \mu(k)\mathbf{R}(k)]\mathbf{w}(k-1) + \mu(k)\mathbf{p}(k). \quad (9)$$

The RI algorithm has a major advantage compared to the RLS algorithm in that it does not require the updating of the inverse autocorrelation matrix. Also, its computational complexity is less than that of the RLS as will be shown in Table I. Due to the update of inverse autocorrelation matrix, RLS type algorithms may face numerical stability problems [3], which is not the case for the RI algorithm. Convergence analysis of the RI algorithm [1] shows that the algorithm converges in the mean and the mean square sense.

TABLE I
COMPUTATIONAL COMPLEXITY OF RI, FAST RI, RLS AND SFTRLS

	Mult./Div.	Add./Sub.
RI	$\frac{5}{2}N^2 + \frac{7}{2}N$	$2N^2 + N$
RI_{fast}	$N^2 + \left(5 + \frac{3}{2}\log_2 N\right)N - 1$	$N^2 + \left(\frac{3}{2} + \frac{9}{2}\log_2 N\right)N + \frac{1}{2}$
RLS	$3N^2 + 11N + 9$	$3N^2 + 7N + 4$
SFTRLS	$9N + 13$	$9N + 1$

III. FAST IMPLEMENTATION ISSUES

In its current form, even though the RI algorithm has computational complexity less than that of the RLS algorithm, its computational complexity can be reduced further. Note that the term $\mu(k)\mathbf{R}(k)\mathbf{w}(k-1)$ in (9) requires computational complexity of $O(N^2)$, which makes the algorithm inefficient in its current form. Fast calculations of the product is possible if the autocorrelation matrix $\mathbf{R}(k)$ is assumed to be toeplitz. Toeplitized autocorrelation matrix $\widehat{\mathbf{R}}(k)$ can be obtained by replacing each diagonal of the original autocorrelation matrix by its average value. If we note that the elements of $\widehat{\mathbf{R}}(k)$ can be generated from the symmetric sequence g_q:

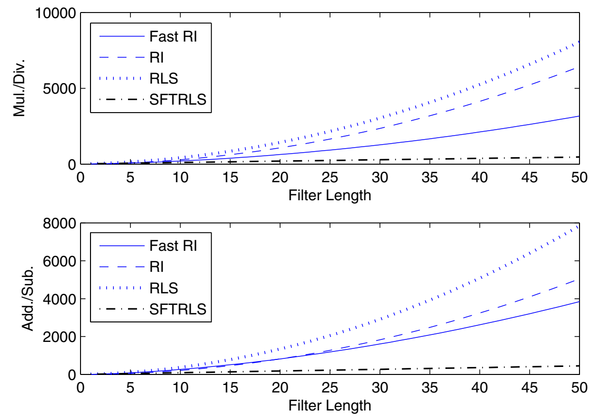

Fig. 1. Computational Complexity of RI, Fast RI, RLS and SFTRLS.

$$g_q = \begin{cases} r_q, & 0 \le q \le (N-1) \\ r_q^*, & -(N-1) \le q < 0 \end{cases} \quad (10)$$

where $r_q^* = r_{-q}$. The n^{th} element of the vector $\mathbf{w}_f(k) = \widehat{\mathbf{R}}(k)\mathbf{w}(k)$ can be written as

$$w_{f,n}(k) = \sum_{m=1}^{N} r_{n,m} w_{m-1}(k), \quad n = 1, 2, \ldots, N. \quad (11)$$

Rewriting (11) in terms of the sequence in (10) gives

$$w_{f,n}(k) = \sum_{m=0}^{N-1} g_{n-m-1} w_m(k), \quad n = 1, 2, \ldots, N, \quad (12)$$

equation (12) represents the convolution sum. Now, taking $(2N-1)$-point DFT of both sides of (12) at time k

$$W_{fe}(l) = G(l)W_e(l), \quad l = 1, 2, \ldots, 2N-1, \quad (13)$$

where $W_{fe}(l)$ is the DFT of the zero-padded sequence $\{w_{fe,n}(k); \quad n = 1, 2, \ldots, 2N-1\}$:

$$w_{fe,n}(k) = \begin{cases} w_{f,n}(k), & n = 1, 2, \ldots, N \\ 0, & n = N+1, \ldots, 2N-1, \end{cases} \quad (14)$$

and $W_e(l)$ is the DFT of $\mathbf{w}_e(k) = [\underline{\mathbf{0}} \;\; \mathbf{w}(k)]$ where $\underline{\mathbf{0}}$ is an $(N-1)$-dimensional zero vector. The sequence $\{w_{f,n}(k); \quad n = 1, 2, \ldots, N\}$ can now be recovered from the inverse DFT of $W_{fe}(l)$.

In Fig. 1, it is shown that the number of Mul./Div. and the Add./Sub., respectively, for the Fast RI algorithm is considerably less than that of the RI and the RLS algorithms. This is especially true for large filter lengths.

IV. ADAPTIVE EQUALIZATION MODEL

In this section, the channel equalization model of a linear dispersive communication channel is described. The block diagram of the model is depicted in Fig. 2. The two random-number generators (1 and 2) in the model are used to generate

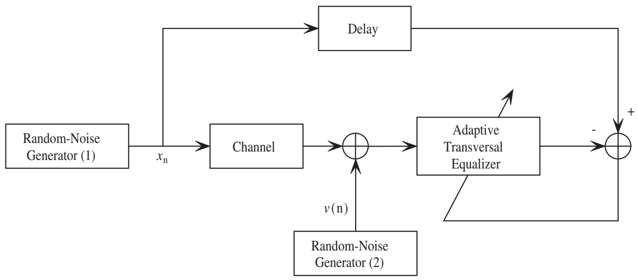

Fig. 2. Block diagram of the adaptive equalization model.

the transmitted signal x_n and the additive noise at the receiver input, respectively. The sequence x_n is a Bernoulli sequence with $x_n = \pm 1$; the random variable x_n has a zero mean and variance 1, and $v(n)$ has a zero mean and variance dependent on the desired SNR. The impulse response of the channel is defined by:

$$h(n) = \begin{cases} \frac{1}{2}\left[1 + \cos\left(\frac{2\pi}{W}(n-2)\right)\right], & n = 1, 2, 3, \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

where W controls the eigenvalue spread of the autocorrelation matrix.

V. Simulation Results

In the simulations, the performance of the Fast RI algorithm is compared to that of the RLS and SFTRLS algorithms in the channel equalization problem described in section IV. All the experiments were done with the parameters: filter length of $N = 11$ taps, SNR= 30dB, $W = 3.3$ and delay of $\Delta = 7$.

A. Additive White Gaussian Noise

In this experiment, the input signal x_n is assumed to be corrupted with an AWGN process after passing through the channel. Simulations were done with the following parameters: For the Fast RI algorithm: $\beta = 0.985$, $\mu_0 = 0.004$. For the RLS algorithm: $\beta = 0.985$. For the SFTRLS algorithm, [9]: $\lambda = 0.991$, $\kappa_1 = 1.5$, $\kappa_2 = 2.5$ and $\kappa_3 = 1$. Fig. 3 shows that the Fast RI and RLS algorithms converge approximately at the same time but the Fast RI algorithm converges to a lower mse (mse=-28dB for Fast RI and mse=-24dB for RLS), whereas, the SFTRLS algorithm converges to a much higher mse of -9dB. Furthermore, the computational complexity of the proposed algorithm is much lower than that of the RLS algorithm as shown in Table. I. Also, it should be noted that the Fast RI algorithm does not require the inversion of the autocorrelation matrix which will ensure its numerical stability. However, the RLS algorithm may face numerical stability problems due to the loss of Hermitian symmetry and loss of positive definiteness of the inverse autocorrelation matrix, [10].

B. Additive Correlated Gaussian Noise

The signal is assumed to be corrupted with an ACGN process. 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 Fast RI algorithm: $\beta = 0.985$, $\mu_0 = 0.004$. For the RLS algorithm: $\beta = 0.985$. For the SFTRLS algorithm, [9]: $\lambda = 0.991$, $\kappa_1 = 1.5$, $\kappa_2 = 2.5$ and $\kappa_3 = 1$. Fig. 4 shows that the Fast RI and RLS algorithms converges approximately at the same time but the Fast RI algorithm converges to a slightly lower mse (mse=-29dB for Fast RI). Even though the SFTRLS algorithm has computational complexity $O(N)$, it converges to a much higher mse of -9dB and slower than the Fast RI and the RLS algorithms. Additionally, the forgetting factors λ in SFTRLS algorithm and β in RLS algorithm, has to to be chosen such that their values are very close to 1. However, this poses a limitation for the algorithms since small values of these parameters may be required in nonstationary environments. On the other hand, in the case of the Fast RI algorithm, there is no restriction on β (In the current experiment, it is selected to be the same as the one in the RLS algorithm). In the case of the SFTRLS algorithm, with $\lambda < 0.991$, the algorithm faces

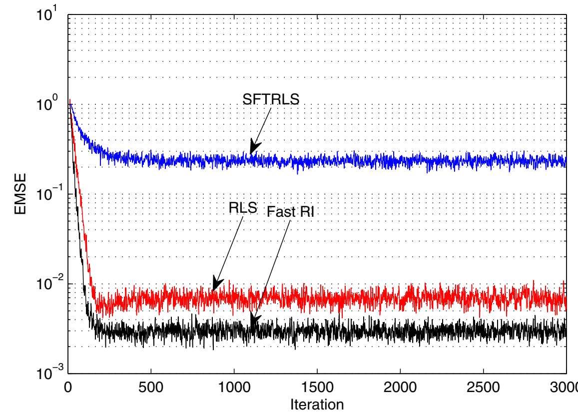

Fig. 3. The ensemble MSE for Fast RI, RLS and SFTRLS in AWGN. $N = 11$, SNR= 30dB, $\Delta = 7$. Fast RI: $\beta = 0.985$, $\mu_0 = 0.004$. RLS: $\beta = 0.985$. SFTRLS: $\lambda = 0.991$, $\kappa_1 = 0.5$, $\kappa_2 = 0.5$ and $\kappa_3 = 0.5$.

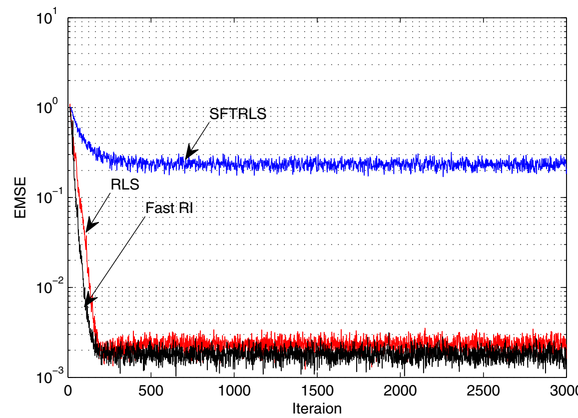

Fig. 4. The ensemble MSE for Fast RI, RLS and SFTRLS in ACGN. $N = 11$, SNR= 30dB, $\Delta = 7$. Fast RI: $\beta = 0.985$, $\mu_0 = 0.004$. RLS: $\beta = 0.985$. SFTRLS: $\lambda = 0.991$, $\kappa_1 = 0.5$, $\kappa_2 = 0.5$ and $\kappa_3 = 0.5$.

stability problems.

C. Additive White Impulsive Noise

Due to the man-made noise, underwater acoustic noise, atmospheric noise, etc, the noise added to the received signal may not be modeled using Gaussian distribution. This type of noise which has a heavy-tailed distribution is characterized by outliers and may be better modeled using a Gaussian mixture model. In order to study the effects of the impulsive components (outliers) of the noise process in the channel equalization setting described in section IV, an impulsive noise process is generated by the probability density function [11], [12]: $f = (1-\epsilon)G(0,\sigma_n^2) + \epsilon G(0,\kappa\sigma_n^2)$ with variance σ_f^2 is given as: $\sigma_f^2 = (1-\epsilon)\sigma_n^2 + \epsilon\kappa\sigma_n^2$, where $G(0,\sigma_n^2)$ is a Gaussian probability density function with zero mean and variance σ_n^2 that represents the nominal background noise. $G(0,\kappa\sigma_n^2)$ represents the impulsive component of the noise model where ϵ is the probability and $\kappa \geq 1$ is the strength of impulsive components, respectively. The white impulsive noise process is generated with $\epsilon = 0.2$, $\kappa = 100$. Simulations were done with the following parameters: For the Fast RI algorithm: $\beta = 0.985$, $\mu_0 = 0.004$. For the RLS algorithm: $\beta = 0.985$. For the SFTRLS algorithm: $\lambda = 0.991$, $\kappa_1 = 1.5$, $\kappa_2 = 2.5$ and $\kappa_3 = 1$. Fig. 5 shows that the Fast RI and RLS algorithms converges approximately at the same time but the Fast RI algorithm converges to a lower mse (mse=-27dB for Fast RI and mse=-21dB for RLS), whereas, the SFTRLS algorithm converges to a much higher mse of -9dB.

D. Additive Correlated Impulsive Noise

A correlated impulsive 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 impulsive noise process created by the process described in section V-C with the same parameters, and ρ is the correlation parameter ($\rho = 0.7$). Simulations were done with the following parameters: For the Fast RI algorithm: $\beta = 0.985$, $\mu_0 = 0.004$. For the RLS algorithm: $\beta = 0.985$. For the SFTRLS algorithm: $\lambda = 0.991$, $\kappa_1 = 1.5$, $\kappa_2 = 2.5$ and $\kappa_3 = 1$. Fig. 6 shows that the Fast RI and RLS algorithms converges approximately at the same time but the Fast RI algorithm converges to a lower mse (mse=-27dB for Fasr RI and mse=-21dB for RLS), whereas, the SFTRLS algorithm converges to a much higher mse of -9dB.

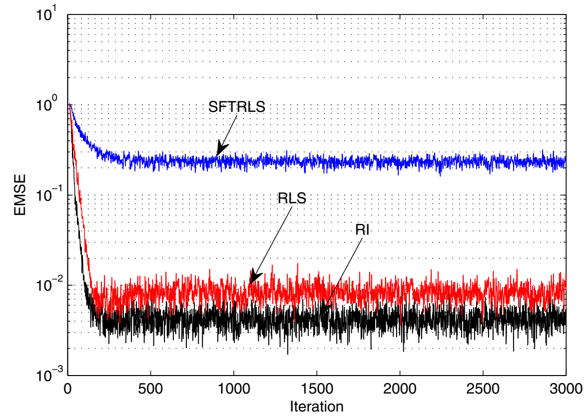

Fig. 5. The ensemble MSE for Fast RI, RLS and SFTRLS in AWIN. $N = 11$, SNR= 30dB, $\Delta = 7$, $\epsilon = 0.2$, $\kappa = 100$. Fast RI: $\beta = 0.985$, $\mu_0 = 0.004$. RLS: $\beta = 0.985$. SFTRLS: $\lambda = 0.991$, $\kappa_1 = 0.5$, $\kappa_2 = 0.5$ and $\kappa_3 = 0.5$.

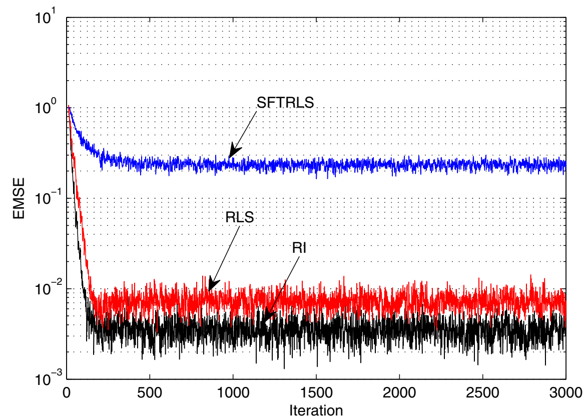

Fig. 6. The ensemble MSE for Fast RI, RLS and SFTRLS in ACIN. $N = 11$, SNR= 30dB, $\Delta = 7$, $\epsilon = 0.2$, $\kappa = 100$. Fast RI: $\beta = 0.985$, $\mu_0 = 0.004$. RLS: $\beta = 0.985$. SFTRLS: $\lambda = 0.991$, $\kappa_1 = 0.5$, $\kappa_2 = 0.5$ and $\kappa_3 = 0.5$.

VI. Conclusion

Recursive Inverse (RI) adaptive filtering algorithm which uses a variable step-size and the instantaneous value of the autocorrelation matrix in the coefficient update equation was proposed in [1]. The algorithm was shown to have a higher performance compared with the RLS and RRLS algorithms. In this paper, a more efficient version with lower computational complexity is presented. The performance of the algorithm has been tested in a channel equalization setting and compared with those of the RLS and SFTRLS algorithms in AWGN, ACGN, AWIN and ACIN environments. Simulation results show the advantage of the Fast RI algorithm in reducing the effects of the impulsive components (outliers) over the RLS algorithm, and the significant reduction in the number of Mul./ Div. and Add./Sub. operations required. The performance of the Fast RI algorithm is shown to be superior to that of the SFTRLS algorithm under the same conditions. On the other hand, it should be noted that the Fast RI algorithm does not require the inversion of the autocorrelation matrix which will ensure its numerical stability. However, the RLS algorithm and its variants may face numerical stability problems due to the loss of Hermitian symmetry and loss of positive definiteness of the inverse autocorrelation matrix.

References

[1] M. S. Ahmad, O. Kukrer and A. Hocanin, "Recursive Inverse Adaptive Filtering Algorithm", *IEEE Signal Processing Letters*, submitted: October, 10^{th}, 2009, (under review).

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

[3] G. O. Glentis, K. Berberidis and S. Theodoridis, "Efficient Least Squares Adaptive Algorithms for FIR Transversal Filtering", *IEEE Signal Processing Magazine*, pp. 13-41, July 1999.

[4] S. Qiao, “Fast adaptive RLS algorithms: a generalized inverse approach and analysis”, *IEEE Transactions on Signal Processing*, 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 Processing*, 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] 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.

[8] S. Haykin, *Kalman Filtering and Neural Networks*, John Wiley & Sons, NY, 2001.

[9] P. S. R. Diniz, *Adaptive Filtering Algorithms and Practical Implementation*, 3^{rd} Ed., Springer, LLC, NY, 2008.

[10] A. H. Sayed, *Adaptive Filters*, John Wily & Sons, NJ, 2008.

[11] H. Deliç and A. Hocanın, “Robust Detection in DS/CDMA”, *IEEE Transactions on Vehicular Technology*, Vol. 51, No. 1, January 2002.

[12] B. Seyfe and S. Valaee, “A New Choice of Penalty Function for Robust Multiuser Detection Based on M-Estimation,” *IEEE Transactions on Communications Theory*, Vol. 53, pp. 224-227 , February 2005.