

# A NEW PROPORTIONATE FAST LMS/NEWTON ALGORITHM FOR ADAPTIVE FILTERING

*Y. Zhou, S. C. Chan and K. L. Ho*

*{yizhou, scchan, klho}@eee.hku.hk*

Department of Electrical and Electronic Engineering  
The University of Hong Kong, Pokfulam Road, Hong Kong

## ABSTRACT

This paper proposes a new proportionate adaptive filtering algorithm which exploits the advantageous features of the generalized proportionate NLMS (GP-NLMS) algorithm and the fast LMS/Newton algorithm. By means of an efficient switching mechanism, the new algorithm works alternately between the GP-NLMS and the fast LMS/Newton algorithms in order to combine their respective advantages. The overall converging speed and steady state performance for both sparse and dispersive channels as well as tracking performance are thus significantly improved. Computer simulations on an echo cancellation problem verify the superior performance of the new algorithm over both the GP-NLMS algorithm and the conventional fast LMS/Newton algorithm.

## 1. INTRODUCTION

Adaptive filtering is frequently employed in communications, control, and many other applications. Many adaptive filtering algorithms have been proposed [1] and they can broadly be classified into two different classes: the least mean squares (LMS) algorithm and the recursive least squares (RLS) algorithm. These two algorithms form the benchmarks for adaptive filtering study and different approaches have been proposed for improving their performances and reducing the arithmetic complexities [1]. One very efficient class of LMS algorithms is the fast Newton algorithms, which include the fast Newton transversal filters (FNMF) [2] and the fast LMS/Newton algorithm [3]. In these algorithms, the input signal to the adaptive filter is modeled as a low,  $M$ -order auto-regressive (AR) process so that the Kalman gain vector in the Newton algorithm can be efficiently approximated. These two algorithms have a similar

arithmetic complexity of  $O(2L + 5M)$  and  $O(2L + 6M)$  respectively (where  $L$  represents the order of the adaptive filter), but the latter is more numerically stable than the former because of its LMS characteristics. It also considerably outperforms the Normalized LMS algorithm (NLMS) in convergence speed and possesses the attractive properties of regular hardware implementation. Unfortunately, it also suffers from some deficiencies such as slow initial convergence (due to the estimation delay of the AR parameters of the input signal) and the inferior tracking performance as compared to the NLMS algorithm. The latter is of particular importance in many applications involving time-varying channels such as echo cancellation problems [4], where the echo path might be non-stationary. Another problem, which is also specific to echo cancellation problems, is that it does not exploit the sparsity of the echo path, which has attracted considerable interests recently. The proportionate stepsize NLMS (PNLMS) algorithm [5] improves the initial convergence of the NLMS algorithm by adapting the stepsizes for each coefficient in the weight vector. Unfortunately, it converges rather slowly when the impulse response is dispersive and when the input is colored. Afterwards, two improved PNLMS algorithms, called the PNLMS++ algorithm [6] and the IP-NLMS algorithm [7] were proposed. Alternation between the NLMS and PLMS and their linearly combinations were proposed to improve the convergence performance. More recently, Hoshuyama et al proposed a Generalized Proportionate Affine Projection Algorithm (GP-APA) [8] where approximate gradient of the weight vector are used to adapt the stepsizes. Faster initial converging and tracking speeds over the PNLMS were reported. The GP-APA reduces to the GP-NLMS algorithm when one input vector is processed at a time, which also has the lowest arithmetic complexity in the family. Since the PNLMS algorithm and its variants are NLMS-type algorithm, their performances will be considerably affected by colored input signals, especially for dispersive channels with

considerable number of non-zero coefficients.

Motivated by these important properties of the fast LMS/Newton algorithm and the GP-NLMS algorithm, a new proportionate fast LMS/Newton (PLMS/Newton) algorithm is proposed in this paper to improve the initial convergence and tracking performances of its conventional counterpart. The basic idea is to separate the updating of the input AR model and the NLMS part of the algorithm during tracking and initial convergence. This allows the NLMS part to operate essentially as a GP-NLMS algorithm in the initial converging stage and during time-varying environments and hence achieve a better overall performance. On the other hand, when the weight vector has converged to a value sufficiently close to its steady state value or when the channel response has settled down to a relatively constant value after considerable variation, the input AR modeling can be coupled again to the NLMS part to further lower the steady state error over the GP-NLMS algorithm especially when the input is colored and the channel response is non-sparse. One important ingredient of the new algorithm is an efficient switching mechanism, which determines when the input AR modeling should be coupled to or decoupled from the NLMS part of the algorithm. Experimental results show that a reliable decision can be derived from the variations of the weight vector, which can be readily derived from the GP-NLMS algorithm. Moreover, since the structure of the new algorithm is very similar to its conventional counterpart, except for the additional arithmetic operations in determining the switching decision, the complexity of the new algorithm is similar to the GP-NLMS algorithm. This paper is organized as follows: the conventional fast LMS/Newton algorithm is reviewed in section 2. The proposed PLMS/Newton algorithm is presented in section 3. Experimental results and comparisons with the LMS/Newton, GP-NLMS and NLMS algorithms are presented in section 4. Finally, conclusions are drawn in section 5.

## 2. THE FAST LMS/NEWTON ALGORITHM

Consider the adaptive system identification problem depicted in Fig. 1. The input signal  $x(n)$  passes through the unknown channel with an impulse response  $\mathbf{W}^*$  and generates the output  $d_0(n)$ , which is assumed to be corrected by additive noise  $\eta_0(n)$  to form the desired signal  $d(n)$ . An adaptive filter with an impulse response  $\mathbf{W}(n)$  is usually employed to identify the unknown channel, where the weight vector is continuously adjusted to minimize certain performance criterion such as the mean-square-error (MSE) of the instantaneous error

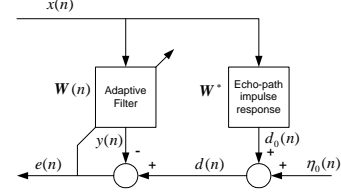


Figure 1. Adaptive system identification

estimation  $e(n)$ . The latter is equal to the difference between the filter output  $y(n)$  and the desired input  $d(n)$ . In the Newton algorithm, the weight update equations are given by

$$e(n) = d(n) - \mathbf{X}^T(n)\mathbf{W}(n) \quad (1)$$

$$\mathbf{W}(n+1) = \mathbf{W}(n) + \mu \cdot e(n)\hat{\mathbf{R}}^{-1}(n)\mathbf{X}(n) \quad (2)$$

where  $\hat{\mathbf{R}}^{-1}(n)$  is the inverse of the estimated input covariance matrix and  $\mu$  is the stepsize which controls the converging and tracking speed, and the steady state error of the algorithm. In the fast LMS/Newton algorithm, the input  $x(n)$  is modeled as an  $M$ -order AR process (usually  $M \ll L$ ) so that  $\hat{\mathbf{R}}^{-1}(n)$  can be efficiently approximated using linear prediction method. As a result, the computational complexity of the basic Newton method can be significantly reduced, similar to the LMS algorithm, while offering significant performance improvement. More precisely, in the fast LMS/Newton algorithm,  $\hat{\mathbf{R}}^{-1}(n)$  is factored into the following form:

$$\hat{\mathbf{R}}^{-1}(n) = \mathbf{L}_M^T(n)\mathbf{D}^{-1}(n)\mathbf{L}_M(n) \quad (3)$$

where  $\mathbf{L}_M(n)$  is an  $(L \times L)$  lower triangular matrix consisting of the coefficients of the backward predictors. Due to the AR model assumption of the input, it can be simplified to

$$\mathbf{L}_M(n) = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ a_{1,1}(n) & 1 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{M,M}(n) & a_{M,M-1}(n) & \dots & 1 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & a_{M,M}(n-1) & \dots & a_{M,1}(n-1) & 1 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & a_{M,M}(n-N+M+1) & a_{M,M-1}(n-N+M+1) & \dots & 1 \end{bmatrix} \quad (4)$$

where the element  $a_{p,i}(n)$  is the  $i$ -th coefficient of the  $p$ -th order backward predictor for  $x(n)$ , and  $\mathbf{D}(n)$  is a diagonal matrix whose  $i$ -th element is the estimated power of the  $i$ -th backward prediction error. In [3], two algorithms with different complexities are presented. The algorithm employed in this paper is based on Algorithm 2, which has a much simpler structure than Algorithm 1, and hence more suitable for hardware implementation. Note that the  $(M+1)$ -th through the  $L$ -th rows of  $\mathbf{L}_M(n)$  are shifted version of each other and define the extended input and coefficient vectors of  $\mathbf{X}(n)$  and  $\mathbf{W}(n)$  as:

$$\mathbf{X}_E(n) = [x(n+M), \dots, x(n), \dots, x(n-L-M+1)]^T \quad (5)$$

$$\mathbf{W}_E(n) = [w_{-M}(n), \dots, w_0(n), \dots, w_{N+M-1}(n)]^T. \quad (6)$$

By freezing the first  $M$  and last  $M$  unnecessary elements of  $\mathbf{W}_E(n)$  to zero during all iterations and denoting the resultant vector as  $\mathbf{W}(n)$ , the fast LMS/Newton algorithm can be written as follows:

$$e(n) = d(n-M) - \mathbf{X}^T(n-M)\mathbf{W}(n) \quad (7)$$

$$\mathbf{W}(n+1) = \mathbf{W}(n) + 2\mu e(n)\mathbf{u}_a(n) \quad (8)$$

$$\mathbf{u}_a(n) = \mathbf{L}_2(n)\tilde{\mathbf{D}}^{-1}(n)\mathbf{L}_1(n)\mathbf{X}_E(n), \quad (9)$$

where  $\mathbf{L}_1(n)$  and  $\mathbf{L}_2(n)$  are respectively  $(L+M)$ -by- $(L+2M)$  and  $L$ -by- $(L+M)$  matrices whose rows consist of consecutively shifted and delayed coefficients of the  $M$ -order forward and backward predictors  $[a_{M,M}(n), a_{M,M-1}(n), \dots, 1]$  and  $[a_{M,M}(n), a_{M,M-1}(n), \dots, 1]$ . By exploiting the shifting property of  $\mathbf{u}_a(n)$  and  $\mathbf{b}_E(n) = \mathbf{L}_1(n)\mathbf{X}_E(n)$ , it is possible to reduce the computational complexity of the algorithm to  $2L+6M$  multiplications and additions for each iteration. The predictor parameters can be efficiently calculated using a lattice predictor and the Levinson-Durbin algorithm.

### 3. THE PROPORTIONATE FAST LMS/NEWTON ALGORITHM

Although the fast LMS/Newton algorithm efficiently exploits the colored nature of the input speech signal and gives an overall fast converging speed, its initial converging speed is however slightly slow because of the time consumed for estimating the AR model parameters. Besides, it does not take into account the sparse nature of the impulse response of the echo path. As mentioned earlier, the GP-APA algorithm [8] controls the step size for each filter tap individually based on their approximated time derivatives. In this paper, we shall consider a special case of the GP-APA algorithm with the dimensionality of the input data space equal to 1. It then reduces to a GP-NLMS algorithm with the least complexity:

$$\mathbf{W}(n+1) = \mathbf{W}(n) + \alpha e(n)\mathbf{U}(n)\mathbf{X}(n) / (\mathbf{X}^T(n)\mathbf{U}(n)\mathbf{X}(n) + \delta) \quad (10)$$

$$\mathbf{U}(n) = \text{diag}(u_1(n), u_2(n), \dots, u_L(n)), \quad (11)$$

$$u_i(n) = \frac{1-\beta}{2} \frac{\hat{c}_i(n-1)}{\|\hat{\mathbf{c}}(n-1)\|_1} + \frac{\beta}{2L}, \quad (12)$$

$$\|\hat{\mathbf{c}}(n-1)\|_1 = \sum_{i=1}^L |\hat{c}_i(n-1)|$$

$$\hat{c}_i(n) = w_i(n-1) - \hat{w}_i(n-1), \quad (13)$$

$$\hat{w}_i(n) = \eta \hat{w}_i(n-1) + (1-\eta)w_i(n-1), \quad i = 1, 2, \dots, L, \quad (14)$$

where  $u_i(n)$  and  $\hat{c}_i(n)$  are respectively the stepsize and approximated time derivative of the  $i$ -th filter tap.  $\beta$

serves as the minimum step size, and  $\eta$  is the forgetting factor for calculating smoothed tap weight  $\hat{w}_i(n)$  and  $\delta$  is a constant.  $\|\cdot\|_1$  denotes the  $l_1$  norm of a vector. From (10)~(14), we can see that the advantages of the GP-NLMS algorithm are two folds: First, in time-invariant echo path with sparse impulse response, the time derivatives  $\hat{c}_i(n)$  allow significant tap weights to be given a larger stepsize and vice versa. This results in a faster initial converging speed. Secondly, since  $\hat{c}_i(n)$  tends to reflect the time variations of the filter weights, it yields a faster tracking speed in slowly time-varying channels. However, as demonstrated by simulation results in section 4, these two advantages will become less profound when the echo path is dispersive. Moreover, due to the noise introduced in calculating the gains, the steady state error of the GP-NLMS algorithm will become noisier as well. Motivated by these observations, it is desirable to employ the GP-NLMS algorithm during initial convergence and time-varying environment, while employing a fast LMS/Newton algorithm near the steady state in order to achieve a lower steady state error and faster converging speed for colored inputs and various echo paths. In other words, we need an efficient switching mechanism to switch alternately between these two algorithms according to a certain measure so as to increase the overall converging and tracking speed. We now study a measure that is derived from the approximated time derivatives of the weight vector  $\hat{c}_i(n)$  in (13). Consider the  $l_1$  norm of the vector  $\|\hat{\mathbf{c}}(n-1)\|_1$  and from the GP-NLMS algorithm, we find that it will converge gradually from its initial value to a very small value when the algorithm is about to converge to its steady state. In addition, its value will be quite unstable during tracking of time-varying echo path responses. Therefore, it forms a good measure to implement the switching mechanism, where we can switch to the fast LMS/Newton algorithm when  $\|\hat{\mathbf{c}}(n-1)\|_1$  becomes smaller than a certain threshold. More precisely, let us denote the instant gradient of  $\|\hat{\mathbf{c}}(n-1)\|_1$  by  $G_c(n)$  and define the decaying ratio  $\chi(n)$  as  $G_c(n)/G_c^0$ , where  $G_c^0$  represents the initial value of  $G_c(n)$  obtained by averaging the first  $P$  data, i. e.  $G_c(k)$  from  $k = 1, 2, \dots, P$ . A small value of  $\chi(n)$  indicates a considerably diminished variations in the weight vector, and hence the filter has approached the end of its initial converging period. By choosing appropriately a threshold, say  $\hat{\chi}$ , it is possible to compare  $\chi(n)$  against this threshold to determine whether switching is necessary. When  $\chi(n)$  is larger than  $\hat{\chi}$ , the algorithm is likely to be in its initial converging stage or in the stage of tracking a time-varying echo path. The GP-NLMS will be chosen to give a fast initial converging and tracking performance. Whereas, when  $\chi(n)$  falls

below  $\hat{\chi}$ , the algorithm is likely to converge slowly and the fast LMS/Newton algorithm should be invoked to accelerate the convergence and further lower the steady state error. To guarantee that  $\chi(n)$  has actually decreased below the threshold, the switching decision should be made as  $\chi(n)$  is less than the threshold for  $Q$  consecutive observations, where  $Q$  denotes the decision window length. The parameters  $\hat{\chi}$ ,  $P$  and  $Q$  can be chosen experimentally in practical applications and simulation results show that the performance of the algorithm is not too sensitive to these values if they are reasonably chosen. The detail of the resulting PLMS/Newton algorithm is summarized in table 1, where for simplicity, the division operation is treated as multiplication and the windowing operations are not included and its implementation block diagram is plotted in Fig. 2. It can be seen that the AR estimation part of the fast LMS/Newton algorithm, which usually constitutes a small part of the algorithm, is implemented in each iteration.

When the fast LMS/Newton algorithm is invoked, the NLMS update in (7) to (9) based on the whitened input  $u_a(n)$  will be used. Otherwise, the GP-NLMS update in (10)~(14) will be employed based on the input  $x(n)$ . Because of this sharing, the arithmetic complexity is nearly the same as the GP-NLMS algorithm, except for the low order AR estimation part.

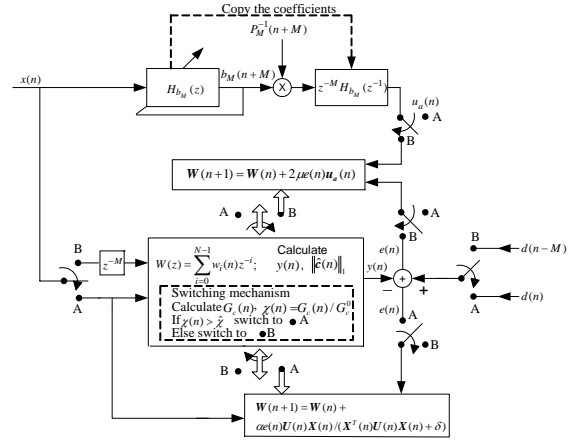


Figure 2. The implementation block diagram for the PLMS/Newton algorithm

#### 4. SIMULATION RESULTS

We now investigate the performance of the proposed algorithm through computer simulations of an echo canceller for both sparse and dispersive environments. The system model is similar to that shown in Fig. 1. The colored input signal is modeled as a speech signal using an AR process with coefficients [1 -0.65 0.693 -0.22 0.309 -0.177] as given in [3]. The resultant signal possesses a spectrum closely resembling those of speech signals. The sparse echo path is a realistic one given by the ITU-T recommendation G.168 [9] with a length of 128. Its impulse response plotted in Fig. 3(a) shows the significant coefficients are clustered only in a small region. The impulse response of the dispersive echo path is plotted in Fig. 3(b) and it is randomly generated and normalized to have unit power. The power of the additive white Gaussian noise is set to be 0.0001. Four algorithms, the fast LMS/Newton, the PNLMS, the GP-NLMS and the proposed PLMS/Newton were compared. For the GP-NLMS algorithm,  $\eta = 0.999$ ,  $\gamma = 1$ ,  $\alpha = 0.2$ ,  $\delta = 0.00001$ , and  $\beta = 0.5$ . For the fast LMS/Newton algorithm and the PNLMS algorithm, the stepsizes were chosen so that the steady state MSE of all the algorithms is approximately -40dB. For the PLMS/Newton algorithm,  $\hat{\chi} = 0.5$ ,  $P = 20$ , and  $Q = 100$ . Three experiments have been conducted. **Exp.1:** Sparse echo path. MSE is employed as the converging performance measure of all the algorithms (same in **Exp.2**). From Fig. 4, we can see the proposed hybrid algorithm was switched from the GP-NLMS algorithm to the fast LMS/Newton algorithm at around the 1300<sup>th</sup> iteration and thus it outperformed the other algorithms. **Exp.2:** The dispersive echo path. It can be seen from Fig. 5 that the initial converging behavior of the GP-NLMS and the PNLMS algorithms considerably degraded due to the presence of the dispersive impulse response. In contrast, the proposed PLMS/Newton

TABLE 1. THE PROPORTIONATE FAST LMS/NEWTON ALGORITHM

1. Initialization	
$W(0) = 0$ ; Given initial value averaging window length $P$ , calculate $G_e^0$	
2. Adaptation	
$X_E(n) = [x(n+M), \dots, x(n), \dots, x(n-L-M+1)]^T$	} 6M Mult. 6M Add. (Employ Levinson-Durbin algorithm)
$W_E(n) = [w_{-M}(n), \dots, w_0(n), \dots, w_{N+M-1}(n)]^T$	
$u_a(n) = L_2(n) \tilde{D}^{-1}(n) L_1(n) X_E(n)$	
FOR $i=1$ to $L$ LOOP	
$\hat{w}_i(n) = \eta \hat{w}_i(n-1) + (1-\eta) w_i(n-1)$ ,	} 2L Add.
$\hat{c}_i(n) = w_i(n-1) - \hat{w}_i(n-1)$	
END OF LOOP	
$\ \hat{e}(n)\  = \sum_{i=1}^L  \hat{c}_i(n) $	} L Add.
Calculate $G_e(n) \cdot \chi(n) = G_e(n) / G_e^0$	
IF $\chi(n) > \hat{\chi}$ (GP-NLMS)	
FOR $i=1$ to $L$ LOOP	
$u_i(n) = \frac{1-\beta}{2} \frac{\hat{c}_i(n-1)}{\ \hat{e}(n-1)\ } + \frac{\beta}{2L}$	} 5L+1 Mult. 2L-1 Add.
END OF LOOP	
$U(n) = \text{diag}(u_1(n), u_2(n), \dots, u_L(n))$	
$W(n+1) = W(n) + \alpha e(n) U(n) X(n) / (X^T(n) U(n) X(n) + \delta)$	
ELSE (Fast LMS/Newton)	
$e(n) = d(n-M) - X^T(n-M) W(n)$	} 2L Mult. 2L Add.
$W(n+1) = W(n) + 2\mu e(n) u_a(n)$	
END IF	
Total: 5L+6M+1Mult. 5L+6M-1Add. working with GP-NLMS 2L+6M Mult. 5L+6M Add. working with fast LMS/Newton	

algorithm still performs very well and has the best performance among the other algorithms. The switching happened roughly at the 1200<sup>th</sup> iteration. **Exp.3:** Tracking in sparse echo path environment. The mean square of the echo path impulse response estimation difference (MSD) is employed as the performance measure. The slowly varying echo path follows the model given in [3] as  $w_i(n+1) = w_i(n) + \varepsilon |w_i(n)| v_i(n)$ ,  $i = 1, 2, \dots, L$ , where  $\varepsilon$  is a small constant equal to 0.01 and  $v_i(n)$ 's are a set of independent Gaussian white noise sequences with unit variance. Fig. 6 shows the tracking performance of all the algorithms. Because the value of  $G_c(n)$  varied dramatically, the PLMS/Newton algorithm quickly switched to the GP-NLMS mode. As a result, it has a performance similar to the GP-NLMS algorithm. They both outperformed the PNLMS and the fast LMS/Newton algorithms. Apparently, this resemblance of the PLMS/Newton and the GP-NLMS algorithms will increase with the value of the observation window length  $Q$ .

## 5. CONCLUSION

A new PLMS/Newton algorithm for adaptive filtering is presented. It efficiently combines the advantages of the GP-NLMS and the fast LMS/Newton algorithms by means of an efficient switching mechanism. Computer simulations on an echo canceller application verify the superior performance of the new algorithm over the conventional fast LMS/Newton algorithm in terms of convergence speed and steady state performance for both sparse and dispersive channels as well as colored input signals.

## 6. REFERENCES

[1] S. Haykin, *Adaptive Filter Theory*, 4<sup>th</sup> ed. Prentice Hall, 2001.

[2] G. V. Moustakides and S. Theodoridis, "Fast newton transversal filters-a new class of adaptive estimation algorithms," *IEEE Trans. on Signal Processing*, vol. 39, No. 10, pp. 2184-2193, Oct. 1991.

[3] B. F. Boroujeny, "Fast LMS/Newton algorithms based on autoregressive modeling and their application to acoustic echo cancellation," *IEEE Trans. On Signal Processing*, vol. 45, pp. 1987-2000, Aug. 1997.

[4] J. Benesty et al. *Advances in Network and Acoustic Echo Cancellation*, Springer, 2001.

[5] D. L. Duttweiler, "Proportionate normalized least-mean-squares adaptation in echo cancelers," *IEEE Trans. On Speech and Audio Processing*, vol. 8, pp. 508-518, Sept. 2000.

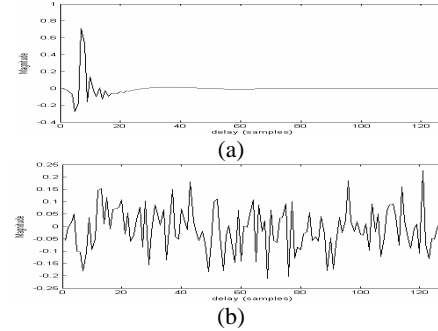


Figure 3. Echo path impulse response: (a) sparse (b) dispersive.

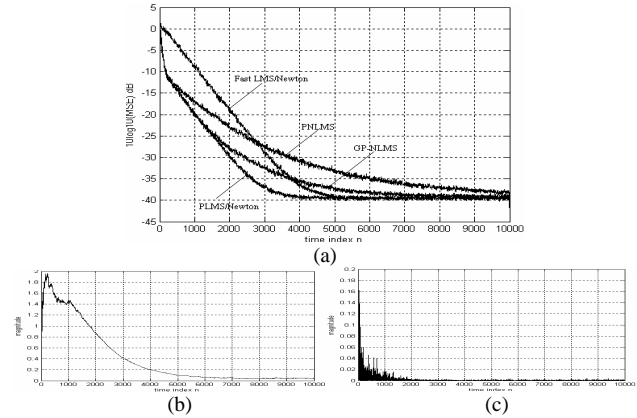


Figure 4. Example of sparse echo path: (a) MSE vs. time, (b)  $\|\hat{c}(n-1)\|$ , (c)  $G_c(n)$

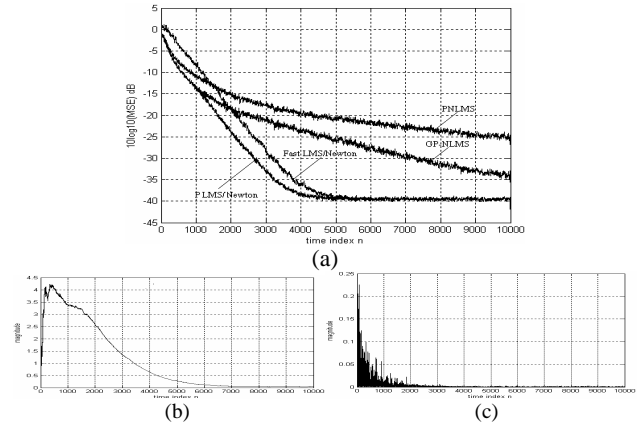
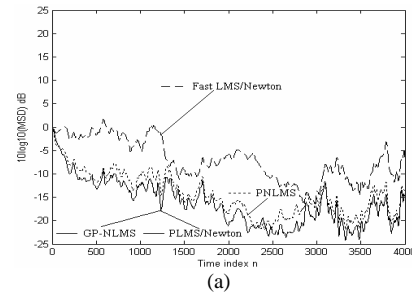


Figure 5. Example of dispersive echo path: (a) MSE vs. time, (b)  $\|\hat{c}(n-1)\|$ , (c)  $G_c(n)$



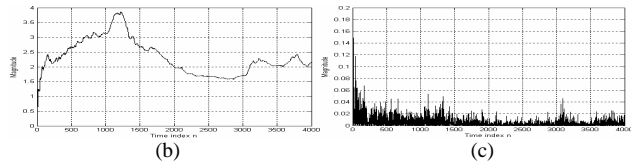


Figure 6. Example of tracking slowly-varying sparse echo path: (a) MSD vs. time, (b)  $\|\hat{c}(n-1)\|$ , (c)  $G_c(n)$

[6] S. L. Gay, "An efficient, fast converging adaptive filter for network echo cancellation," *Signals, Systems & Computers*, 1998. Conference Record of the 32<sup>nd</sup> Asilomar Conference on, vol. 1, 1-4, pp. 394-398, Nov. 1998.

[7] J. Benesty, S. L. Gay, "An improved PNLMS algorithm," *IEEE ICASSP-02*, vol. 2, pp. 1881-1884, 2004.

[8] O. Hoshuyama, R. A. Goubran and A. Sugiyama, "A generalized proportionate variable step-size algorithm for fast changing acoustic environments," *IEEE ICASSP-04*, vol. 4, pp. 161-164, May 2004.

[9] "Digital Network Echo Cancellers," ITU-T Recommendation G.168, 2000.