



Title	A wavelet based partial update fast LMS/Newton algorithm
Author(s)	Zhou, Y; Chan, SC; Ho, KL
Citation	Proceedings of 2005 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS 2005), Hong Kong, China, 13-16 December 2005. In Conference Proceedings, 2005, p. 817-820
Issued Date	2005
URL	http://hdl.handle.net/10722/45923
Rights	Creative Commons: Attribution 3.0 Hong Kong License

A WAVELET BASED PARTIAL UPDATE FAST LMS/NEWTON ALGORITHM

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 studies a wavelet based partial update fast LMS/Newton algorithm. Different from the conventional fast LMS/Newton algorithm, the proposed algorithm first uses a shorter-order, partial Haar transform-based NLMS adaptive filter to estimate the peak position of the long, sparse channel impulse response, and then employs the fast LMS/Newton algorithm integrated with partial update technique to fulfill the rest convergence task. The experimental results demonstrate the proposed algorithm outperforms its conventional counterpart in convergence performance and possesses a significantly lower computational complexity.

1. INTRODUCTION

Many adaptive filtering algorithms have been proposed [1] in literature and among them the least mean squares (LMS) algorithm and the recursive least squares (RLS) algorithm are two typical families. One very efficient class of LMS algorithms is the fast LMS/Newton algorithm [2]. In this algorithm, 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. It has a low arithmetic complexity of $O(2L_c + 6M)$ (where L_c represents the order of the adaptive filter), and a higher numerical stability than other LS-based variants because of its LMS characteristics. It also 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 failure to exploit the sparseness of the echo path which is specific to echo cancellation problems and has attracted considerable interests recently.

It is widely known that adaptive filters can be carried out in the transform domain (by employing DCT, DFT or DWT, etc) to achieve better convergence properties [1]. Among the applicable schemes, the discrete wavelet transform (DWT), a powerful tool for analysis of non-stationary signals, has received considerable attention recently in adaptive filtering because of its two very good properties: the time and spectral localization [3]. The former can result in computation cost reduction, and the latter can lead to input decorrelation so as to fasten the convergence. In literature, a few reported results show an improvement in

signal modeling and satisfactory convergence speed for a variety of conditions. In [4], Hosur and Tewfik studied the special sparse structure of the wavelet transformed correlation matrix and proposed the wavelet transform domain algorithms. It is also shown in [5] that fewer coefficients are needed for the Haar transform-based adaptive filter than that working in time domain. Later in [6], the authors exploited the hierarchical structure of the Haar transform to adaptively determine which significant coefficients to adapt and thus realize the rapid identification of the sparse channel impulse response as well as the reduced additional steady state error induced by the adaptation of those trivial coefficients. More recently in [7], the partial Haar transform is employed to estimate the peak position of the channel impulse response and another shorter time-domain adaptive filter is used centering in time about the estimated peak to adapt upon a delayed version of the input. This work directly motivated us to improve the present fast LMS/Newton algorithm by first employing a low-order, partial Haar transform-based NLMS adaptive filter to locate the peak position of the long, sparse channel impulse response and then continue the update centering the estimated position with an low-order fast LMS/Newton adaptive filter, upon which we additionally apply the efficient partial update mechanism [8], [9] to further lower the computational complexity. These manipulations result in the proposed wavelet based partial update fast LMS/Newton algorithm which has both a faster convergence performance and a significantly reduced computational complexity. This paper is organized as follows: the conventional fast LMS/Newton algorithm is reviewed in section 2. The proposed wavelet based partial update fast LMS/Newton algorithm is presented in section 3. Experimental results and comparisons are presented in section 4. Finally, conclusions are drawn in section 5.

2. THE FAST LMS/NEWTON ALGORITHM

Figure 1 depicts the adaptive system identification problem. The unknown channel impulse response W^* is to be identified using an adaptive filter with weight vector $W(n)$. $x(n)$ is the input signal and $e(n)$ is the instantaneous estimation error between the channel and adaptive filter outputs $d(n)$ and $y(n)$. $\eta_0(n)$ is the background noise. $W(n)$ is usually continuously adjusted to minimize certain performance criterion such as the

mean-square-error (MSE) of $e(n)$. 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 μ is the stepsize controlling the converging and tracking speed, and the steady state error of the algorithm. In the fast LMS/Newton algorithm [3], the input $x(n)$ is modeled as an M -order AR process (usually $M \ll L_C$) so that $\hat{\mathbf{R}}^{-1}(n)$ can be efficiently approximated using linear prediction method. Consequently, 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_C -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:

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

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

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 $(L_C+M) \times (L_C+2M)$ and $L_C \times (L_C+M)$ 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

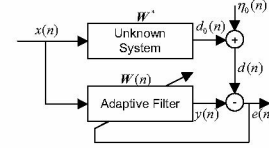


Figure 1. System identification structure

computational complexity of the algorithm to $2L_C + 6M$ multiplications and additions for each iteration. The predictor parameters can be efficiently calculated using a lattice predictor and the Levinson-Durbin algorithm.

3. THE WAVELET BASED PARTIAL UPDATE FAST LMS/NEWTON ALGORITHM

Although the fast LMS/Newton algorithm is very efficient, its $2L$ computational complexity may be still prohibitive for some applications such as Echo Cancellation [10] in which the number of adaptive filter coefficients usually amount to either a few hundreds, or more harshly, to some thousands. On the other hand, it does not exploit the sparseness of the echo path, which widely exists in real situations and has attracted considerable interests recently. Among these researches, the wavelet-transform based schemes are found to be very promising. Wavelet transform has very good time localization property and thus only a small number of coefficients are needed to efficiently identify the sparse channel impulse response. It also has the additional benefit of good spectral localization and thus can diagonalize many types of input correlation matrices and hence provides a certain amount of whitening for colored input. Some relevant works can be found in literature such as [5], [6] and [7]. It is the work in [7] that triggered our research. More specifically, in [7], the partial Haar transform is employed to estimate the peak of the channel impulse response and another shorter-order time-domain adaptive filter then centers in time about the estimated peak to adapt upon a delayed version of the input. This has naturally helped us raise an idea of using a similar shorter-order, partial Haar transform-based NLMS adaptive filter to locate the peak position of the long, sparse channel impulse in the very short beginning period, and then switching back to fast LMS/Newton algorithm for later-stage update. Next, we will show how this idea is implemented and, with the introduction of partial update (PU) technique [8], [9], how the computational complexity of the resultant algorithm can be further reduced.

The Haar wavelets are the simplest of wavelet functions which are discrete-time orthonormal sequences $\psi_{mn}(t)$ defined by

$$\psi_{mn}(t) = \psi_{m0}(t - 2^m n)$$

$$\text{where } \psi_{m0}(t) = 2^{-\frac{m}{2}}, \text{ for } 0 \leq t \leq 2^{m-1} - 1$$

$$= -2^{-\frac{m}{2}}, \text{ for } 2^{m-1} \leq t \leq 2^m - 1$$

$$\text{or } = 0 \text{ elsewhere}$$

where the indices m (a natural number) and n (an integer) correspond to the scale and translation respectively. Thus the $N \times N$ Haar wavelet transform matrix, where $N = 2^K$ has the following structure (exemplified with $N = 8$):

$$H_8 = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix}$$

When the full matrix is utilized to transform the input vector, the Haar wavelet transform-based adaptive filter is analogous to other transform domain (like DCT, DFT) adaptive filter which can thus exactly model any channel impulse response. In contrast, any subset/dimensionality truncation, which results in the partial Haar transform, will inevitably increase the estimation error and thus fail to correctly identify the channel. However, as pointed out in [7], if we merely aim at estimating the location of the peak of the channel impulse response, we do not have to stick to the full Haar transform because the partial transform is less costly.

The basic idea of our work is to divide the whole convergence process into two stages. The similarities of the weight vector update part of the NLMS algorithm [1] and that of the fast LMS/Newton algorithm (Equ. (8)) enables us to switch the update part flexibly from the Haar transform domain NLMS type to the fast LMS/Newton type. More precisely, at Stage 1, a partial Haar transform-based NLMS adaptive filter $\mathbf{W}_{S1} (L_{S1} \times 1, L_{S1} < L_C)$ is employed to estimate the peak position of the sparse channel impulse response, and at the same time the input AR model $\mathbf{u}_a(n) (L_{S2} \times 1, L_{S2} < L_C)$ is also updated but temporarily separated from the weight vector update part. Stage 2 comes after a short time duration when the peak is supposed to be identified and the input AR model can be coupled again to the weight vector update part and the convergence process continues with adaptive filter $\mathbf{W}_{S2} (L_{S2} \times 1)$ centering in time about the peak of the weights. In order to tolerate the estimation error, N_T taps prior to the estimated peak position can be included into updating in Stage 2, where N_T is a small number which can be experimentally selected. Without loss of generality, the length of \mathbf{W}_{S2}, L_{S2} , can be selected as $L_{S2} = 1/2L_C$ and its initial value can be set as the inverse partial Haar transform of the weight vector \mathbf{W}_{S1} at the end of Stage 1. In this way, the computational complexity of the proposed partial Haar transform-based fast LMS/Newton algorithm can be reduced to different amounts.

Nevertheless, due to the lack of a priori knowledge, although L_{S2} will have been halved, it still can not be selected as sufficiently low as the real number of the significant coefficients of the sparse channel. Some coefficients are also very close to zero and the calculation redundancy still exists. To circumvent this problem and further lower the computational complexity, we applied the partial update (PU) technique introduced in [8, 9], which is effective and easy to be implemented, into the fast LMS/Newton algorithm in Stage 2. The difference between [8, 9] and our algorithm is that the former executes the PU part

directly upon the input vector $\mathbf{X}(n)$ while in our case, the weight coefficients are partially updated with regard to the whitened input AR model $\mathbf{u}_a(n)$. To this end, together with the part in Stage 1, the resultant PU-fast LMS/Newton algorithm in Stage 2 comprise the whole Partial Haar transform-based Partial Update fast LMS/Newton algorithm (PHPU-fast LMS/Newton) which is summarized in Table 1. Despite the light computation overhead for estimating the peak position with PH-based NLMS algorithm, the proposed algorithm only needs $L_{S2} + N_B L + 2 + 6M$ multiplications, one extra division, and $O(B) + B \log_2 B$ comparisons (the conventional algorithm needs $2L_{S2} + 6M$ multiplications). A block diagram depicting the hardware implementation detail is plotted in Fig.2.

TABLE I
THE PHPU-FAST LMS/NEWTON

1. Initialization
$\mathbf{W}_{S1}(0) = 0$;
2. Adaptation
Stage 1:
For $n = 1, 2, \dots, n_0$
Input AR model $\mathbf{u}_a(n)$ update
$\mathbf{X}_E(n) = [x(n+M), \dots, x(n), \dots, x(n-L-M+1)]^T$
$\mathbf{W}_E(n) = [w_{-M}(n), \dots, w_0(n), \dots, w_{N+M-1}(n)]^T$
$\mathbf{u}_a(n) = \mathbf{L}_2(n) \tilde{\mathbf{D}}^{-1}(n) \mathbf{L}_1(n) \mathbf{X}_E(n)$
Partial Haar transform-based NLMS
$\bar{\mathbf{X}}(n) = \mathbf{H}_{PH} \mathbf{X}(n)$, $e(n) = d(n) - \mathbf{W}_{S1}^T(n) \bar{\mathbf{X}}(n)$
$\mathbf{W}_{S1}(n+1) = \mathbf{W}_{S1}(n) + \mu e(n) \bar{\mathbf{X}}(n)$
At $n = n_0$, calculate the peak estimated position p_c
Stage 2:
$\mathbf{W}_{S2}(n_0) = \mathbf{H}_{PH}^T \mathbf{W}_{S1}(n_0)$, For $n = n_0, n_0 + 1, n_0 + 2, \dots$
$e(n) = d(n-M) - \mathbf{X}^T(n-M) \mathbf{W}_{S2}(n)$
$\mathbf{X}_E(n) = [x(n+M), \dots, x(n), \dots, x(n-L-M+1)]^T$
$\mathbf{u}_a(n) = \mathbf{L}_2(n) \tilde{\mathbf{D}}^{-1}(n) \mathbf{L}_1(n) \mathbf{X}_E(n)$
Partitioning $\mathbf{u}_a(n)$ and $\mathbf{W}(n)$ into B blocks of $L = L_C / B$ coefficients each,
$\mathbf{W}_{S2}(n) = [\mathbf{w}_1^T(n) \quad \mathbf{w}_2^T(n) \quad \dots \quad \mathbf{w}_B^T(n)]^T$
$\mathbf{u}_a(n) = \mathbf{L}_2(n) \tilde{\mathbf{D}}^{-1}(n) \mathbf{L}_1(n) \mathbf{X}_E(n)$
$= [\mathbf{u}_1^T(n) \quad \mathbf{u}_2^T(n) \quad \dots \quad \mathbf{u}_B^T(n)]^T$
$\mathbf{W}_{S2}(n+1) = \mathbf{W}_{S2}(n) + \mu e(n) \mathbf{A}_{N_B(n)} \mathbf{u}_a(n)$
where $\mathbf{A}_{N_B(n)}$ is a $L_{S2} \times L_{S2}$ matrix used for selecting N_B block of coefficients from $\mathbf{u}_a(n)$ to be updated. It has N_B identity matrix $\mathbf{I}_{L \times L}$ on its diagonal and zeros elsewhere. The positions of these matrices are in accordance with those of the largest N_B blocks of coefficients of $\mathbf{W}_{S2}(n)$ with the largest $\ \mathbf{u}_i(n)\ ^2, i = 0, 1, \dots, B$. This result can be obtained through certain sorting algorithm and calculation of the largest norm blocks which are specified in [8, 9].

4. SIMULATION RESULTS

We now verify the efficiency of the proposed PHPU-fast LMS/Newton algorithm using computer simulations of an acoustic echo cancellation problem. The system model is depicted in Fig. 3. The input signal $x(n)$ 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 [2]. The echo path impulse response, as shown in Fig. 4 (a), is a realistic one with 256 coefficients given as $m_1(k)$ by the ITU-T recommendation G.168 [10]. The background noise $\eta_0(n)$ is a white Gaussian random sequence with variance $\delta_{\eta_0}^2(n) = 0.0001$. For simplicity, no double-talk is assumed to present. The partial Haar transform H_{sp} is a 64×256 matrix which consists of the 64th through 128th rows of the full Haar transform matrix H_s . Hence, $L_{s1} = 64$. The selection of the stepsize of the NLMS algorithm is not critical and thus is set to be 0.002 to guarantee convergence stability. The switching time index n_0 is set to be 100. As can be observed in Fig. 4 (b), the peak position is estimated to be at around the 11th tap. In our experiment, we introduce $N_T = 5$ taps to tolerate the estimation error. $L_{s2} = 128$, and the global stepsize of the Partial Update fast LMS/Newton algorithm is set to be 0.5 to ensure the final steady state error is around -40dB. Various inner block coefficient lengths L and the number of blocks to be updated N_B are tested. We tested three cases: $L = 1, N_B = 64$, $L = 2, N_B = 32$, and $L = 4, N_B = 16$. The curves are obtained through the average of the results of 100 independent runs. From Fig. 5, it can be observed that the PHPU-fast LMS/Newton algorithm with all the above parameters converged faster than its conventional counterpart. The higher the computational complexity is, the faster the convergence speed will be. Especially, when $L = 1, N_B = 64$, the performance of the proposed algorithm is very close to the Partial Haar-based fast LMS/Newton algorithm (PHPU-fast LMS/Newton without PU part).

5. CONCLUSION

In this paper, a wavelet based partial update fast LMS/Newton algorithm is proposed. By employing partial Haar transform and partial update technique, the proposed algorithm will exhibit a faster convergence speed and a significantly reduced computational complexity than its conventional counterpart performance. The efficiency of the proposed algorithm has been verified with an adaptive echo cancellation problem.

6. REFERENCES

[1] S. Haykin, *Adaptive Filter Theory*, 4th ed. Prentice Hall, 2001.
[2] B. F. Boroujeny, "Fast LMS/Newton algorithms based on autoregressive modeling and their application to acoustic echo cancellation," *IEEE Trans. Signal Processing*, vol. 45, pp. 1987-2000, Aug. 1997.
[3] Stephane Mallat, *A Wavelet Tour of Signal Processing*, 2nd ed. San Diego, Academic Press, c1999.
[4] S. Hosur and A. H. Tewfik, "Wavelet transform domain adaptive FIR filtering," *IEEE Trans. Signal Processing*, vol. 45, No. 3, pp. 617-629, Mar. 1997.
[5] M. Doroslovacki and H. Fan, "On-line identification of echo-path impulse responses by Haar-wavelet-based adaptive filter," in *Proc. IEEE Conf. Acoust., Speech Signal Process.*, 1999, pp. 1065-1068.
[6] K. C. Ho and S. D. Blunt, "Rapid identification of a sparse impulse response using an adaptive algorithm in the Haar domain," *IEEE Trans. Signal Processing*, vol. 51, No. 3, pp. 628-638, Mar. 2003.
[7] N. J. Bershad and A. Bist, "Fast Coupled Adaptation for Sparse Impulse Responses Using a Partial Haar Transform," *IEEE Trans. Signal Processing*, vol. 53, No. 3, pp. 966-976, Mar. 2005.

[8] K. Dogancay and O. Tanrikulu, "Adaptive Filtering Algorithms With Selective Partial Updates," *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing*, vol. 48, No. 8, pp. 762-769, Aug. 2001.
[9] S. Werner, M. L. R. Campos and P. S. R. Diniz, "Partial-Update NLMS Algorithms with Data-Selective Updating," *IEEE Trans. Signal Processing*, vol. 52, No. 4, pp. 938-949, April 2005.
[10] "Digital Network Echo Cancellers," ITU-T Recommendation G.168, 2000.

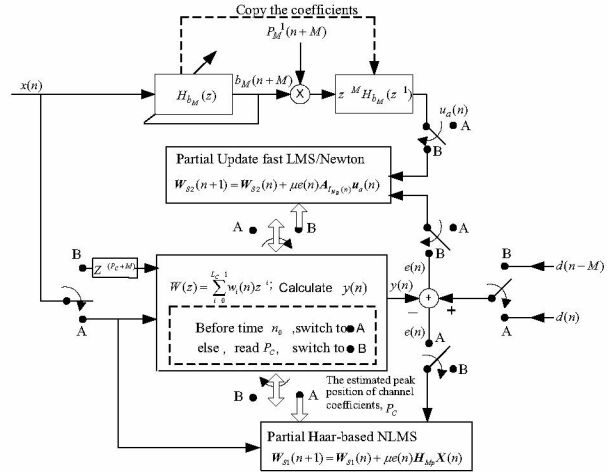


Figure 2. The implementation block diagrams for PHPU-fast LMS/Newton algorithm.

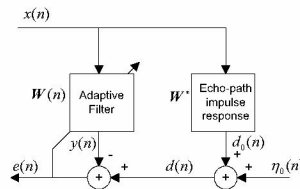


Figure 3. Adaptive Echo Cancellation

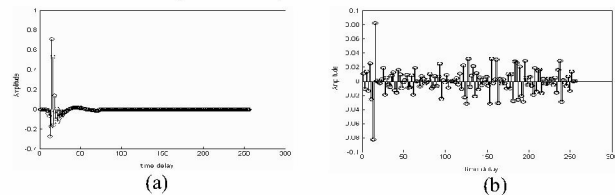


Figure 4 (a) the real channel impulse response; (b) the peak position identified by partial haar transform-based adaptive filter

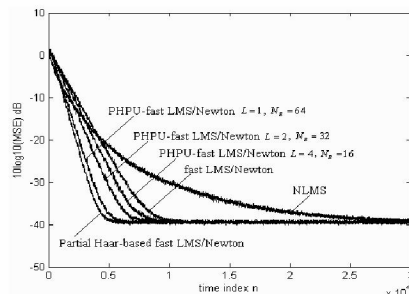


Figure 5. Performance of PHPU-fast LMS/Newton algorithm. MSE vs. time n , with various L and N_B .