Performance Analysis of the Frequency Domain LMS Adaptive Filter Using the Sliding DFT

GEORGE-OTHON GLENTIS and KRISTINA GEORGOULAKIS

Technological Education Institute of Crete
Branch at Chania, Department of Electronics
C GREECE.

Abstract:- In this paper the performance of the Frequency Domain LMS adaptive filter is investigated, for the case when the Sliding Discrete Fourier Transform is utilized for the frequency domain data transformation. A statistical performance analysis in terms of the mean, and the mean squared error of the filter parameters is presented. The convergence speed of the algorithm is analyzed in terms of the eigenvalue spread of the input signal autocorrelation matrix. The theoretical analysis results are verified by computer simulations.

Key-Words:- Adaptive Filtering, Frequency Domain LMS, Sliding Window DFT.

1 Introduction

The design of adaptive filters and system identification algorithms with optimum learning, in the sense of minimizing the accumulated squared error between the output signal and a desired response, has been the subject of major research for a long time. Typical examples include the design of decision feedback equalizers in digital communications, the design of acoustical echo cancelers in hands-free telephony and in teleconferencing, the design of filters that will recover and enhance a signal transmitted through a communication channel, [1],[2].

One of the most common algorithm for adaptive filtering is Widrow's Least Mean Squared error (LMS) algorithm, [1]. LMS like algorithms are popular due to the low computational complexity and the simplicity in the hardware realization of the underline algorithmic structure. However, the convergence rate of the algorithm heavily depends on the eigenvalue spread of the correlation matrix of the input data.

In an attempt to improve the performance of the LMS algorithm, unitary transformations on the input data vector, have been used, [4]-[6]. The resulting algorithms may have increased convergence rate for some classes on input signals, yet the computational complexity remains similar to that of the original LMS scheme. When the Discrete Fourier Transform (DFT) is used as the unitary transform, the input data are continuously transformed into the Frequency Domain, (FD). The Sliding Window DFT (SDFT), can be utilized to perform this task, [6], [9]. The SDFT estimates the DFT transform of a rectangular window of the signal, which is continuously updated with new samples as the oldest ones are discarded. The Sampling Frequency (FS) structure, [3], a method for implementing the SDFT based on a set of filter banks, is very popular in adaptive filtering, [6], due to the low computational complexity, the regularity and modularity, facts that are of great importance when high speed implementation on VLSI ASIC, is under consideration, [7].

The objective of this paper is to provide a performance analysis of the FD-LMS algorithm, for the case when the stabilized SF structure is utilized for the computation of the S-DFT of the input data. It is shown that the algorithm converges in the mean, to a transformed Wiener solution. The minimum mean squared error attained, equals to that of the original LMS scheme, provided that the adaptation step size has been properly chosen. The convergence speed of the frequency domain algorithm is proved to be always faster than that of the conventional LMS time domain adaptive scheme. The theoretical results are validated by means of computer experiments.

2 The Sliding DFT

Given a discrete time signal, $x(n)$, the SDFT is defined as the DFT of a running rectangular window of the signal, of size M, that moves at a rate of one sample point each time. Consider the data vector $\mathbf{x}(n) = [x(n)\ x(n-1)\ \ldots\ x(n-M+1)]^T$. The corresponding SDFT is the vector $\mathbf{U}(n) = [u_0(n)\ u_1(n)\ \ldots\ u_{M-1}(n)]^T$ where each element $u_k(n)$, $k = 0, 1 \ldots M-1$, is estimated by the form $u_k(n) = \sum_{\ell=0}^{M-1} x(n-\ell) w_M^{k\ell}/\sqrt{M}$. $w_M = e^{-j\frac{2\pi}{M}}$ is the so called twiddle factor. Thus, $\mathbf{U}(n) = \mathbf{W}\mathbf{x}(n)$, where $\mathbf{W}$ is the DFT matrix, i.e., $\mathbf{W}_{k,l} = w_M^{kl}/\sqrt{M}$. Several techniques have been developed so far for the estimation of the SDFT of real or complex signals. All these methods are based on the recursive estimation of the SDFT algorithm, relating the transform of the current time instant to the transform already estimated the previous time instant, [3],[8], [9]. The main advantage of these algorithms is the reduced computational complexity, which is proportional to the window's length. The FS-SDFT, [3], and the steady-flow LMS-DFT algorithm, [8], can be applied to estimate sliding DFT's of arbitrarily selected window sizes. On the other hand, when efficient algorithms based on FFT-like recursive schemes are applied, [9], the window's size is restricted to be a power of two.

Following the FS-DFT scheme, $u_k(n)$ is efficiently be updated by the set of first order recursions $u_k(n) = w_M^k u_k(n-1) + (x(n) - x(n-M))\sqrt{M}$, which can be implemented by the following filter-bank scheme, [6]

$$u_k(n) = \frac{1}{\sqrt{M}} \frac{1 - z^{-M}}{(1 - w_M^k z^{-1})} x(n) \qquad (1)$$

The marginal instability of the original FS filters, due to the fact that the poles of the FS filters lies exactly on the unit circle, may cause amplification of round-off errors. This problem is alleviated by moving the corresponding poles and zeros slightly inside the unit circle, by replacing z^{-1} by ρz^{-1}. ρ is a stabilization factor, $\rho \in (0, 1]$, [6]. Thus, the following stabilized FS filters are utilized

$$v_k(n) = \frac{1}{\sqrt{M}} \frac{1 - \rho^M z^{-M}}{(1 - \rho w_M^k z^{-1})} x(n) \qquad (2)$$

Following [10], it is possible to associate the recursive structure (2) to the DFT of the an exponential weighted window of the incoming data $x(n)$ as, $v_k(n) = \sum_{\ell=0}^{M-1} \rho^\ell x(n-\ell) w_M^{k\ell}/\sqrt{M}$. In a matrix notation the above formula reads

$$\mathbf{V}(n) = \mathbf{W}\mathbf{L}\mathbf{x}(n) \qquad (3)$$

where, $\mathbf{V}(n) = [v_0(n)\ v_1(n)\ \ldots v_{M-1}(n)]^T$, and $\mathbf{L} = \mathrm{diag}[1\ \rho \ldots \rho^{M-1}]$. The power of the error between the stabilized and the original FS-SDFT transform, over all frequency bins, has be used as a metric of performance. It can explicitly be estimated, in terms of ρ and M, and can be kept arbitrarily small, by proper choice of ρ, [10].

3 Frequency Domain Adaptive Filtering

The Frequency Domain LMS algorithm, [4], can be interpreted as a preconditioned version of the original LMS scheme, [7]. The DFT is efficiently utilized to speed up the convergence of the LMS algorithm. This preprocessing improves the eigenvalue distribution of the input signal autocorrelation matrix, and thus, affects the convergence speed of the LMS algorithm. The performance of the FD-LMS adaptive filter is analyzed, for the case when the stabilized FS-SDFT is utilized for the computation of the frequency domain data transformation.

3.1 The stabilized FS-SDFT FD-LMS

Let $x(n)$ be an input signal and $y(n)$ be a desired response signal. The FD-LMS algorithm based on the modified data transformation, namely the stabilized FS-SDFT defined by eq. (3), takes the form

$$e_V(n) = y(n) - \mathbf{C}_V^H(n-1)_V(n) \quad (4)$$

$$\mathbf{C}_V(n) = \mathbf{C}_V(n-1) + \mu_V \mathbf{P}_V^{-1}\mathbf{V}(n)e^*(n) \quad (5)$$

$\mathbf{P}_V$ is a diagonal matrix with entries the signal power associated to each frequency bin. It has the form $\mathbf{P}_V = \text{diag}\left[\sigma_{V,0}^2,\ \sigma_{V,1}^2,\ \ldots\sigma_{V,M-1}^2\right]$, where, $\sigma_{V,k}^2$ is the signal power at the k-th frequency bin, $\sigma_{V,k}^2 = \mathcal{E}\left[|v_k(n)|^2\right]$. In practice, $\mathbf{P}_V$ is a time varying matrix, whose elements, are calculated in terms of available data, using for example, an exponentially weighted power estimator, implemented by the recursive difference equation $\sigma_{V,k}^2(n) = \lambda\sigma_{V,k}^2(n-1) + (1-\lambda)|v_k(n)|^2$, $\lambda \in (0,1)$. However, in order to make the performace analysis of the FD-LMS algorithm, more tracktable, we will consider $\mathbf{P}_V$ to be a constant diagonal matrix, [4],[5],[14].

The statistical properties of the FD-LMS algorithm described by eqs. (4)-(5), will analyzed using the standard independence assumptions, [1], rephrased in the frequency domain as, [13], *Independence assumptions:* [A.1]. The input vectors $\mathbf{V}(1)$, $\mathbf{V}(2)$, $\ldots$ constitute a sequence of statistically independent vectors. [A.2] The input vector $\mathbf{V}(n)$ is statistically independent of all previous samples of the desired response, $y(1), y(2), \ldots y(n-1)$. [A.3] The desired response signal $y(n)$ is dependent on $\mathbf{V}(n)$, but statistically independent of all previous samples of the desired response signal, i.e., $y(1), \ldots y(n-1)$.

3.2 Mean tap-weight behavior

Let us consider the average tap-weight behavior, $\mathcal{E}\left[\mathbf{C}_V(n)\right]$. Taking the expected value of both sides of eq. (5) we get $\mathcal{E}\left[\mathbf{C}_V(n)\right] = \mathcal{E}\left[\mathbf{C}_V(n-1)\right] + \mu\mathbf{P}_V^{-1}\mathcal{E}\left[\mathbf{V}(n)\mathbf{V}^H(n)\mathbf{C}(n-1)\right] + \mu\mathbf{P}_V^{-1}\mathcal{E}\left[\mathbf{V}(n)y^*(n)\right]$. Using the independence assumptions,A.1-A.3, we are allowed to write $\mathcal{E}\left[\mathbf{V}(n)\mathbf{V}^H(n)\mathbf{C}_V(n-1)\right] = \mathcal{E}\left[\mathbf{V}(n)\mathbf{V}^H(n)\right]$ $\mathcal{E}\left[\mathbf{C}_V(n-1)\right]$. Define the correlation parameters $\mathbf{R}_V = \mathcal{E}\left[\mathbf{V}(n)\mathbf{V}^H(n)\right]$, $\mathbf{D}_V = \mathcal{E}\left[\mathbf{V}(n)y^*(n)\right]$ Thus, $\mathcal{E}\left[\mathbf{C}_V(n)\right] = \left(\mathbf{I} - \mu\mathbf{P}_V^{-1}\mathbf{R}_V\right)\mathcal{E}\left[\mathbf{C}_V(n-1)\right]$ $+\mu\mathbf{P}_V^{-1}\mathbf{D}_V$. Using standard arguments, [1], it can be shown that the steady state solution, $\mathcal{E}\left[\mathbf{C}_V(\infty)\right] \equiv \mathbf{C}_V^\circ$, of the above equation is given by the solution of the system of linear equations $\mathbf{R}_V\mathbf{C}_V^\circ = \mathbf{D}_V$, provided that $0 < \mu_V < 2/\text{tr}(\mathbf{P}_V^{-1}\mathbf{R}_V) = 2/M$. The relationship between $\mathbf{C}_V^\circ$ and the optimum Wiener solution, can be established. Using eq. (3), we get

$$\mathbf{R}_V = \mathbf{WLRLW}^H, \quad \mathbf{D}_V = \mathbf{WLD} \quad (6)$$

where, $\mathbf{R} = \mathcal{E}\left(\mathbf{x}(n)\mathbf{x}^H(n)\right)$, $\mathbf{d} = \mathcal{E}\left(\mathbf{x}(n)y^*(n)\right)$, are the autocorrelation matrix of the input signal, and the cross correlation vector between the input signal and the desired response signal. Finally, we have

$$\mathbf{LW}^H\mathbf{C}_V^\circ = \mathbf{c}^\circ \quad (7)$$

where $\mathbf{c}^\circ$ is the optimum Wiener filter given by the solution of the normal equations $\mathbf{Rc}^\circ = \mathbf{d}$.

3.3 Mean Squared Error Analysis

The instantaneous error $e_V(n)$, defined by eq. (4), can be expressed in terms of the optimum transformed filter parameter as

$$e_V(n) = e_V^\circ(n) - \Delta^H(n)\mathbf{V}(n) \quad (8)$$

$\mathbf{\Delta}(n)$ is the parameter's error vector, $\mathbf{\Delta}(n) = \mathbf{C}_V(n) - \mathbf{C}_V^\circ$. $e_V^\circ(n)$ is the optimum filtering error attained, when the input signal is filtered by the optimum filter parameters, $e_V^\circ(n) = y(n) - \mathbf{C}_V^{\circ H}\mathbf{V}(n)$. The MSE, E_V, is estimated as

$$E_V = \boldsymbol{\mathcal{E}}\left[|e_V(n)|^2\right] = E_V^\circ + E_{ex} \qquad (9)$$

E_V° is the optimum MSE given by

$$E_V^\circ = \boldsymbol{\mathcal{E}}\left[|e_V^\circ(n)|^2\right] = \boldsymbol{\mathcal{E}}\left[|y(n)|^2\right] - \mathbf{D}_V^H\mathbf{C}_V^\circ \quad (10)$$

Using eqs. (6) and (7) it can be easily shown that the optimum MSE, E_V°, equals to the MSE attained by the Wiener solution, i.e., $E_V^\circ = E^\circ$. E_{ex}, the excess MSE error, which is estimated to be $E_{ex} = \mathrm{tr}\,[\mathbf{R}_V\mathbf{K}]$. Matrix $\mathbf{K}$, appeared above, is the covariance of the parameters error vector, $\mathbf{K} = \boldsymbol{\mathcal{E}}\,[\mathbf{\Delta}(n)\mathbf{\Delta}^H(n)]$.

In order the get an explicit expression for E_{ex}, the covariance matrix $\mathbf{K}$ should first be estimated. To this end, $\mathbf{\Delta}(n)$ is expressed as

$$\mathbf{\Delta}(n) = \left(\mathbf{I} - \mu\mathbf{P}_V^{-1}\mathbf{V}(n)\mathbf{V}^H(n)\right)\mathbf{\Delta}(n-1) + \mu\mathbf{P}_V^{-1}\mathbf{V}(n)e_V^\circ(n)$$

Using results from averaging analysis, [12], the solution of the above stochastic difference equation is, for sufficient small μ, close to the solution of another stochastic difference, obtained by replacing $(\mathbf{I} - \mu\mathbf{P}_V^{-1}\mathbf{V}(n)\mathbf{V}^H(n))$ by its average $(\mathbf{I} - \mu\mathbf{P}_V^{-1}\mathbf{R}_V)$. Using the above results, the covariance matrix $\mathbf{K}(n)$ is estimated as

$$\mathbf{K}(n) = \left(\mathbf{I} - \mu\mathbf{P}_V^{-1}\mathbf{R}_V\right)\mathbf{K}(n-1)\left(\mathbf{I} - \mu\mathbf{R}_V\mathbf{P}_V^{-1}\right) + \mu^2\mathbf{P}_V^{-1}\mathbf{R}_V\mathbf{P}_V^{-1}E_V^\circ$$

The steady state solution $\mathbf{K} = \mathbf{K}(\infty)$ is then estimated from the above equation. A first order approximation with respect to μ of $\mathbf{K}$ can be obtained by dropping the term $\mu^2\mathbf{P}_V^{-1}\mathbf{R}_V\mathbf{P}_V^{-1}$, resulting to the more tractable expression

$$\mathbf{P}_V^{-1}\mathbf{R}_V\mathbf{K} + \mathbf{K}\mathbf{R}_V\mathbf{P}_V^{-1} = \mu\mathbf{P}_V^{-1}\mathbf{R}_V\mathbf{P}_V^{-1}E_V^\circ \quad (12)$$

or,

$$\mathbf{R}_V\mathbf{K} + \mathbf{P}_V\mathbf{K}\mathbf{R}_V\mathbf{P}_V^{-1} = \mu\mathbf{R}_V\mathbf{P}_V^{-1}E_V^\circ$$

Thus, $\mathrm{tr}\,(\mathbf{R}_V\mathbf{K}) = 1/2\mu\mathrm{tr}\left(\mathbf{P}_V^{-1}\mathbf{R}_V\right)E^\circ$, since $\mathrm{tr}(AB) = \mathrm{tr}(BA)$. Finally, we get

$$E_V^{ex} = (1/2)\mu M E_V^\circ \qquad (13)$$

3.4 Convergence analysis

The convergence performance of the stabilized FS-SDFT FD-LMS of subsection 3.1, depends on the eigenvalue spread of matrix $\mathbf{P}_V^{-1}\mathbf{R}_V$. On the other hand, the performance of the conventional LMS algorithm depends on the eigenvalue spread of the input autocorrelation matrix $\mathbf{R}$. Therefore, in comparing the performance between these two algorithms, it is sufficient to compare the eigenvalue spread of the corresponding matrices $\mathbf{P}_V^{-1}\mathbf{R}_V$ and $\mathbf{R}$. Following [14], we adopt a useful index that measure the eigenvalue spread of a positive semidefinite matrix, i.e., a function of the ratio of the arithmetic mean over the geometric mean of the eigenvalues, defined as

$$\alpha\left(\mathbf{P}_V^{-1}\mathbf{R}_V\right) = \left(\lambda_a/\lambda_g\right)^M \qquad (14)$$

where, λ_a and λ_g are the arithmetic and the geometric averages of the eigenvalues of matrix $\mathbf{P}_V^{-1}\mathbf{R}_V$, respectively. Large spread of eigenvalues result in large values for $\alpha\,(\mathbf{P}_V^{-1}\mathbf{R}_V)$. For equal eigenvalues $\alpha\,(\mathbf{P}_V^{-1}\mathbf{R}_V)$ is equal to one.

Eq.(14) can be evaluated by a more tractable expression as,

$$\alpha\left(\mathbf{P}_V^{-1}\mathbf{R}_V\right) = \frac{\left(\mathrm{tr}\left[\mathbf{P}_V^{-1}\mathbf{R}_V\right]/M\right)^M}{\det\left[\mathbf{P}_V^{-1}\mathbf{R}_V\right]}$$

Notice that $\mathrm{tr}\,[\mathbf{P}_V^{-1}\mathbf{R}_V] = M$, $\det\,[\mathbf{P}_V^{-1}\mathbf{R}_V] = \det\,[\mathbf{P}_V^{-1}]\det\,[\mathbf{R}_V]$. Thus, $\alpha\,(\mathbf{P}_V^{-1}\mathbf{R}_V) = \frac{\det[\mathbf{P}_V)]}{\det[\mathbf{R}_V]}$. Eq.(6) results to $\det\,[\mathbf{R}_V] = \det\,[\mathbf{L}]^2\det\,[\mathbf{R}]$. Using the equation $\det\,[\mathbf{L}] = \prod_{i=0}^{M-1}\rho^i = \rho^{\frac{M(M-1)}{2}}$, we get $\det\,[\mathbf{R}_V] = \rho^{M(M-1)}\det\,[\mathbf{R}]$. Moreover, $\mathrm{tr}\,[\mathbf{P}_V] = \mathrm{tr}\,[\mathbf{R}_V] = \mathrm{tr}\,[\mathbf{LRL}]$. From the above facts, $\mathrm{tr}\,[\mathbf{P}_V]$ is estimated as

$$\mathrm{tr}\,[\mathbf{P}_V] = \sum_{m=0}^{M-1}\rho^{2m} = \frac{1-\rho^{2M}}{1-\rho^2}\frac{\mathrm{tr}\,[\mathbf{R}]}{M}$$

Thus, $\alpha\left(\mathbf{P}_V^{-1}\mathbf{R}_V\right)$ can be expressed in the form

$$\alpha\left(\mathbf{P}_V^{-1}\mathbf{R}_V\right) = \frac{\det\left[\mathbf{P}_V\right]}{\left(\mathrm{tr}\left[\mathbf{P}_V\right]/M\right)^M}\frac{\left(\mathrm{tr}\left[\mathbf{R}_V\right]/M\right)^M}{\det(\mathbf{R}_V)}$$

Finally, we get $\alpha\left(\mathbf{P}_V^{-1}\mathbf{R}_V\right) = \phi(\rho)\frac{\alpha(\mathbf{R})}{\alpha(\mathbf{P}_V)}$ where, $\phi(\rho) = \left(\frac{1-\rho^{2M}}{M\rho^{(M-1)}(1-\rho^2)}\right)^M$. Notice that, when ρ is close to one, which is the basic assumption that guarantees that the stabilized transform is close to the SDFT, we have $\phi(\rho) \leq 1$. Thus, we can conclude that

$$\alpha\left(\mathbf{P}_V^{-1}\mathbf{R}_V\right) \leq \alpha\left(\mathbf{R}\right) \qquad (15)$$

since, $\alpha(\mathbf{P}_V)$ is, by definition, always greater than, or equal, to one. Thus, the eigenvalue spread of the stabilized FS-SDFT FD-LMS algorithm is expected to be smaller that the eigenvalued spread of the classic LMS scheme, which implies that the former algorithm converges faster than the later one.

4 Simulation

The performance of the stabilized FS-SDFT FD-LMS adaptive algorithm is illustrated by a typical system identification experiment. A stationary AR process of order 2, driven by a white noise signal, was used as an input to an FIR filter. Identification of FIR filters of various orders has been considered, namely, $M_1 = 32$, $M_2 = 64$, $M_3 = 128$, and $M_4 = 256$. The filter coefficients were randomly selected. At the output of the FIR system white gaussian noise was added, resulting to an SNR equal to about 30dB, for all cases.

Five different values for the stabilization factor, ρ, has been tested, namely, $\rho_1 = 1$, $\rho_2 = .99999$, $\rho_3 = .9999$, $\rho_4 = .999$, and $\rho_5 = .99$. The smoothed MSE, E_V for each case has been estimated. It is depicted on Figure 1. Notice that for all four filter tested, the stabilized factor of size $\rho_5 = .99$ failed to give a considerable improvement of the convergence speed. Identification experiments for ρ_1 up to ρ_4, perform almost undistinguishable. The dependence of the ratio $\frac{\lambda_{\max}}{\lambda_{\min}}$ of matrix $\mathbf{P}_V^{-1}\mathbf{R}_V$, on the stabilization factor ρ, is depicted on Figure 2, for different filter sizes.

5 Conclusions

A performance analysis of the FD-LMS adaptive filter has been presented for the case when the Sliding DFT is utilized for the frequency domain data transformation. A statistical performance analysis in terms of the mean, and the mean squared error of the filter parameters has been presented. Although the stabilized FS that is used for the recursive estimation of the SDFT of the input data, results to a non-orthogonal data transform, the performance of the adaptive filter is superior of the performance of the original LMS algorithm, implemented in the time domain. It has been shown that the algorithm converges in the mean, to a transformed Wiener solution. The minimum Mean Squared Error attained, equals to that of the original LMS scheme, provided that the adaptation step size has been properly chosen. The convergence speed of the frequency domain algorithm is always faster than that of the conventional LMS time domain adaptive scheme. Finally, the theoretical results has been validated by means of computer experiments.

Acknowledgement:

This work has been partially supported by E-PEAEK II, Measure 2.5, Action 2.5.1.

References:

[1] B. Widrow and S. Stearns, Adaptive Signal Processing. Prentice Hall, New Jersey, 1985

[2] S. Benedetto, E. Biglieri, Principles of Digital Transmission, Kluwer Academic 1999.

[3] L.R. Rabiner, R.W. Schafer, 'Recursive and nonrecursive realization of digital filters designed by Frequency Sampling Techniques,' IEEE Trans. on Audio ElectroAc., AU-19, pp. 200-207, 1971

[4] S. Narayan, A.M. Peterson, and M.J. Narasimba, 'Transform domain LMS algorithm,' IEEE Trans. Acoust. Speech, Signal Proc., vol. ASSP-31, pp. 609-615, June 1983.

[5] J.C. Lee, and C.K. Un, 'Performance of transform domain LMS adaptive digital filters,' IEEE Trans. Acoust. Speech, Signal Proc., vol. ASSP-34, pp. 499-510, June 1986.

[6] H. Shynk, 'Frequency-Domain and multirate adaptive filtering,' IEEE Signal Proc. Magazine, vol. 9, pp. 14-39, Jan. 1992.

[7] G.O. Glentis, K. Berberidis, and S. Theodoridis, Efficient least squares adaptive algorithms for FIR transversal filtering: a unified view, IEEE Signal Processing Magazine, pp. 13-42, July 1999

[8] B. Widrow, P. Baudrenghien, M. Vetterli, and P. Titchener, 'Fundamental relations between the LMS algorithm and the DFT,' IEEE Trans. on Circuits and Systems, CAS-34, pp. 814-820, July 1987.

[9] B. Farhang-Boroujeny, Y. Lee, and C.C. Ko, 'Sliding transforms for efficient implementation of transform domain adaptive filters,' Signal Processing, 52, pp. 83-96, 1996.

[10] S.C. Douglas and J.K. Soh, 'A numerically stable sliding window estimator and its applications to adaptive filtering,' Proc. 31st Annual Asilomar Conf. on Signals, Systems and Computers, pp. 111-115, Nov. 1997, USA.

[11] G.O. Glentis, 'Pipelined architectures for the TD-LMS adaptive filter,' in Proceeding of the ICASSP-01, Utah, USA, May 2001.

[12] D. Slock, Fast transversal Algorithms, in Adaptive system identification and signal processing algorithms, N. Kalouptsidis, and S. Theodoridis, eds, Prentice Hall 1993.

[13] D. Kim, and P. De Wilde, 'Performance analysis of the DCT-LMS adaptive filtering algorithm, Signal Processing, pp. 1629-1654, 2000.

[14] B. Farhang-Boroujeny, and S. Gazor, 'Selection of orthonormal transforms for improving the performance of the transform domain normalized LMS algorithm,' IEE Proceeding-F, Vol. 139, pp. 327-335, Oct. 1992.

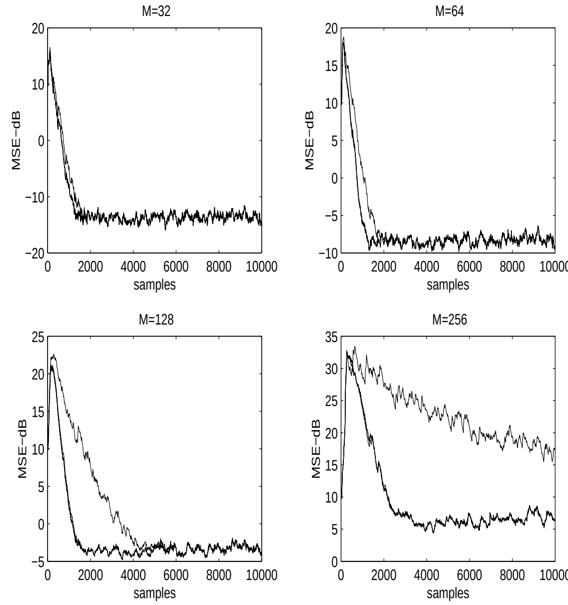

Figure 1. Learning curves of the stabilized FS-SDFT TD-LMS adaptive algorithm for various filter sizes and values of the stabilization parameter ρ.

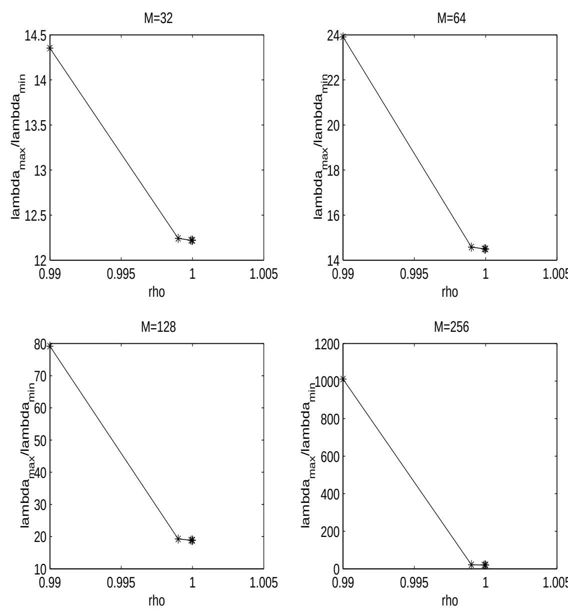

Figure 2. Eigenvalue spread of the stabilized FS-SDFT TD-LMS adaptive algorithm for various filter sizes and values of the stabilization parameter ρ.