Underdetermined LS Algorithms for Adaptive 2D FIR Filtering

GEORGE-OTHON GLENTIS and KRISTINA GEORGOULAKIS

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

Abstract:- In this paper, Underdetermined Least Squares algorithms are derived, for Two-Dimensional adaptive linear filtering and prediction. The derivation of the proposed algorithms is based on the spatial shift invariance properties that the 2D discrete time signals possess. The proposed algorithms have low computational complexity. The convergence speed and the tracking ability of the proposed schemes, is comparable to that of that of the higher complexity 2D RLS algorithms. The performance of the proposed algorithms is illustrated by simulation.

Keywords: Two-Dimensional Adaptive Filtering, Least-Squares Algorithms

1 Introduction

Two-Dimensional (2D) Adaptive Least Squares (LS) filtering and system identification are of great importance in a wide range of applications. Typical examples include image restoration, image enhancement, image compression, 2D spectral estimation, stochastic texture modeling, edge detection etc, [1].

Let $x(m,n)$ be the input of a linear 2D FIR filter. The filter's output $y(m,n)$ is a linear combination of past input values $x(m-i,n-j)$ weighted by the *filter coefficients* $c_{i,j}$, over a support region or *filter mask*, $\mathcal{R}_F$, i.e.,

$$y(m,n) = \sum_{(i,j)\in\mathcal{R}_F} c_{i,j}x(m-i,n-j) \qquad (1)$$

The linear regression 2D model of eq. (1) can be used to describe general support regions, such as strongly causal, causal, semicausal, or noncausal. To keep notation simple, 2D FIR models with strongly causal support regions, of a rectangular shape, will be considered. 2D FIR models with arbitrarily shaped support regions, of general convex shapes, [4], [10], can be handled in a similar way.

Two different approaches, leading to two widely used algorithmic families, have been used for the adaptive estimation of the optimum LS 2D FIR filter, on the basis of the available data set. The first one, is based on a stochastic approximation of the steepest descent method and is known as the 2D LMS family, [2], [8], [9]. The later, is based on a stochastic approximation of the Gauss-Newton method and is known as the 2D RLS family, [3],[4],[6], [11]. Although 2D LMS type algorithms have small computational cost, they suffer from slow convergence rate, especially when the input signal autocorrelation matrix has a large eigenvalue spread. 2D RLS type algorithms does not suffer from such drawback, they have, however, increased computational complexity.

In this paper, underdetermined LS adaptive algorithms for 2D FIR filtering and linear prediction will be considered. The proposed algorithms are interpreted as deterministic counterparts of stochastic Quasi-Newton adaptive algorithms. First, a unified LS criterion will be introduced, and will subsequently be utilized for the derivation of underdetermined adaptive schemes. Underdetermined sliding window, as well as exponential window, adaptive 2D filtering algorithms are proposed. Fast implementation schemes are derived, for both cases, taken into account the spatial shift invariance property that the 2D regressor vector possesses. The proposed algorithms have low computational complexity, comparable to that of the 2D LMS algorithm. The convergence properties of the proposed schemes, is comparable to that of the higher complexity 2D RLS algorithms. The performance of the proposed methods is illustrated by computer experiments.

2 Two-Dimensional FIR filtering

Let us consider a strongly causal 2D support region. To make analysis more tractable, we restrict $\mathcal{R}_F$ to be a rectangular mask, of size $p \times q$. Then, eq. (1) takes the form $y(m,m) = \sum_{i=0}^{q-1} \sum_{j=0}^{p-1} c_{i,j} x(m-i, n-j)$. This equation can be written as a linear regression

$$y(m,n) = \mathbf{X}^T(m,n)\mathbf{C} \tag{2}$$

$\mathbf{X}(m,n)$ is the regressor vector, defined as $\mathbf{X}(m,n) = [\underline{\mathbf{X}}^T(m,n) \ \underline{\mathbf{X}}^T(m-1,n) \dots \underline{\mathbf{X}}^T(m-q+1,n)]$. Entries $\underline{\mathbf{X}}(m-i,n)$, $i = 0, 1 \dots q-1$, carry the input data that lay on the $i-th$ row of the support region $\mathcal{R}_F$, i.e., $\underline{\mathbf{X}}(m-i,n) = [x(m-i,n)\, x(m-i,n-1) \dots x(m-i,n-p+1)]^T$. Vector $\mathbf{C}$ that carries the filter coefficients has a similar structure. Thus, we may write $\mathbf{C} = [\mathbf{C}_1^T \ \mathbf{C}_2^T \ \dots \ \mathbf{C}_q^T]^T$. $\mathbf{C}_i$, $i = 1, 2 \dots q$, is defined as: $\mathbf{C}_i = [c_{0,i} \ c_{1,i} \ \dots \ c_{p-1,i}]^T$. Both vectors $\mathbf{X}(m,n)$ and $\mathbf{C}$, have dimensions $(pq) \times 1$.

2.1 The 2D Adaptive Filtering

Least Squares 2D filtering aims to shape an input signal $x(m,n)$ so that the corresponding

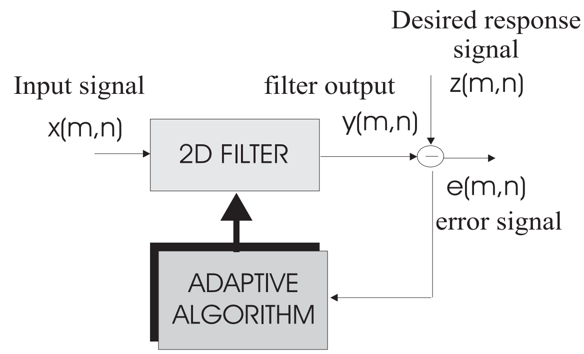

Figure 1: 2D adaptive filtering

output $y(m,n)$ matches a desired signal $z(m,n)$. In the LS formulation we assume that data records of input and the desired response $x(m,n)$ and $z(m,n)$, over the range $(m,n) \in [0,0] \times [M,N]$, provide our knowledge basis, and that we select the filter which minimizes a cost function over the available data record. The minimization procedure is carried out recursively, i.e., the optimum LS filter is re-estimated (or adapted) each time a new pair of measurements is collected, (see Figure 1).

Let us consider the filter output $\mathbf{y}(m,n)$, over a subset of the filter mask $\mathcal{R}_A \subseteq \mathcal{R}_F$. $\mathcal{R}_A$ is called the projection support region, and it is defined as: $(m-k, n-l)$, $0 \leq k \leq l-1$, $0 \leq l \leq k-1$, (see Figure 2). Thus $\mathbf{y}(m,n) = \boldsymbol{\mathcal{X}}^T(m,n)\mathbf{C}(m,n)$. The data matrix $\boldsymbol{\mathcal{X}}(m,n)$, of dimensions $(pq) \times (kl)$, carries the regressor vectors for all points that belong to the projection support region, $\mathcal{R}_A$, It is organized as

$$\begin{aligned}\boldsymbol{\mathcal{X}}(m,n) = [&\underline{\mathcal{X}}(m,n)\ \underline{\mathcal{X}}(m-1,n) \\ &\dots \underline{\mathcal{X}}(m-l+1,n)]\end{aligned} \tag{3}$$

Each entry, $\underline{\mathcal{X}}(m-i,n)$, $i = 0, 1 \dots l-1$, carries the regressor vector associated with samples $x(m-i,n)$, that lay on the $i-th$ row of $\mathcal{R}_A$, i.e., $\underline{\mathcal{X}}(m-i,n) = [\mathbf{X}(m-i,n) \ \mathbf{X}(m-i,n-1) \ \dots \ \mathbf{X}(m-i,n-k+1)]$. Let $\boldsymbol{\epsilon}(m,n)$ be the a

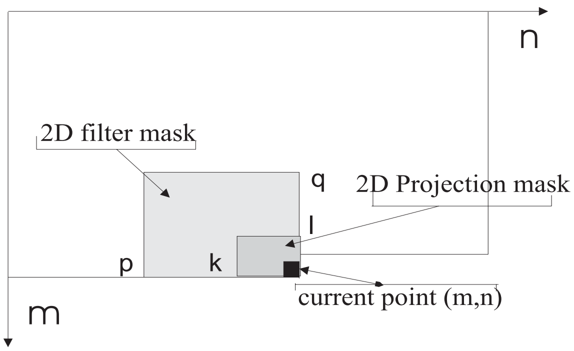

Figure 2: Support region of the 2D filter

posteriori filtering error, defined as

$$\boldsymbol{\epsilon}(m,n) = \mathbf{z}(m,n) - \mathbf{y}(m,n) \qquad (4)$$

$\mathbf{z}(m,n)$, is a vector of dimensions $(kl)\times 1$, which carries the samples of the desired response signal, that lay on the projection area, i.e., $\mathbf{z}(m,n) = [\underline{\mathbf{z}}^T(m,n)\ \underline{\mathbf{z}}^T(m-1,n)\ \ldots \underline{\mathbf{z}}^T(m-l+1,n)]^T$ Entries $\underline{\mathbf{z}}(m-i,n)$, $i = 0,1\ldots k-1$, are defined as: $\mathbf{z}(m-i,n) = [z(m-i,n)\ z(m-i,n-1)\ \ldots z(m-i,n-k+1)]^T$.

Following [5], let us consider adaptive filtering schemes that minimize the error function

$$\mathcal{V}(\mathbf{C}) = ||\boldsymbol{\epsilon}(m,n)||^2_{\mathbf{S}^{-1}(m,n)} + ||\boldsymbol{\Delta}(m,n)||^2_{\mathbf{T}^{-1}(m,n)}$$

where $\boldsymbol{\epsilon}(m,n)$ is defined by eq. (4), and

$$\boldsymbol{\Delta}(m,n) = \mathbf{C}(m,n) - \mathbf{C}(m,n-1)$$

The norm is defined as: $||v||^2_{\mathbf{A}} = v^T\mathbf{A}v$. Matrices $\mathbf{S}^{-1}(m,n)$, $\mathbf{T}^{-1}(m,n)$ are Hermitian positive definite matrices.

Minimizing $\mathcal{V}(\mathbf{C})$, with respect to the filter coefficients vector at space (m,n), i.e., $\mathbf{C}(m,n)$, we get the recursive equation

$$\boldsymbol{\mathcal{A}}(m,n)\mathbf{C}(m,n) = \mathbf{T}^{-1}(m,n)\mathbf{C}(m,n-1) + \boldsymbol{\mathcal{X}}(m,n)\mathbf{S}^{-1}(m,n)\mathbf{z}^T(m,n)$$

where: $\boldsymbol{\mathcal{A}}(m,n) = \boldsymbol{\mathcal{X}}^T(m,n)\mathbf{S}^{-1}(m,n)\boldsymbol{\mathcal{X}}(m,n) + \mathbf{T}^{-1}(m,n)$. Applying the matrix inversion lemma, we get

$$\mathbf{C}(m,n) = \mathbf{C}(m,n-1) + \mathbf{T}(m,n)\boldsymbol{\mathcal{X}}(m,n)\boldsymbol{\mathcal{B}}^{-1}(m,n)\mathbf{e}(m,n) \qquad (5)$$

where: $\boldsymbol{\mathcal{B}}(m,n) = \boldsymbol{\mathcal{X}}^T(m,n)\mathbf{T}(m,n)\boldsymbol{\mathcal{X}}(m,n) + \mathbf{S}(m,n)$, and
$\mathbf{e}(m,n) = \mathbf{z}(m,n) - \boldsymbol{\mathcal{X}}^T(m,n)\mathbf{C}(m,n-1)$
Eq. (5) defines a family of 2D adaptive algorithms, where individual algorithms are revealed by proper choice of the weighting matrices $\mathbf{S}^{-1}(m,n)$, and $\mathbf{T}^{-1}(m,n)$. Using the above updating scheme at our disposal, two 2D underdetermined LS adaptive algorithms are derived, namely, the '*2D Underdetermined Sliding Window LS* adaptive algorithm, (2D USW LS), and the *2D Underdetermined Exponential Window* LS, (2D UEW LS) adaptive algorithm.

$$\mathbf{e}(m,n) = \mathbf{z}(m,n) - \boldsymbol{\mathcal{X}}^T(m,n)\mathbf{C}(m,n-1)$$

$$\mathbf{R}(m,n) = \mathbf{X}^T(m,n)\boldsymbol{\mathcal{X}}(m,n) + \delta\mathbf{I}$$

$$\mathbf{R}(m,n)\mathbf{G}(m,n) = \mathbf{e}(m,n)$$

$$\mathbf{C}(m,n) = \mathbf{C}(m,n-1) + \alpha(m,n)\mathbf{X}(m,n)\mathbf{G}(m,n)$$

Table 1: The 2D USW LS adaptive algorithm

3 2D USW LS adaptive filtering

A 2D USW LS adaptive algorithm can readily be derived from eqs. (5), setting $\mathbf{T}(m,n) = \mathbf{I}$ and $\mathbf{S}(m,n) = \mu(m,n)\boldsymbol{\mathcal{X}}^T(m,n)\boldsymbol{\mathcal{X}}(m,n)$. The resulting algorithm is tabulated in Table 1. The trimming factor, $\alpha(m,n) = 1/(1+\mu(m,n))$, controls the speed of convergence of the algorithm. When $\alpha(m,n)$ is set equal to a constant value, then, the 2D USW LS adaptive algorithm reduces to the 2D Affine Projection Algorithm, proposed in [7]. Matrix $\delta\mathbf{I}$ that appears in Table 1, serves as a regularization factor, that prevents the covariance matrix $\mathbf{R}(m,n)$ of being singular. δ is usually given a small positive value.

Direct implementation of the 2D USW LS adaptive scheme, of Table 1, requires a) the computation of the covariance matrix $\mathbf{R}(m,n)$ and b) a linear system solver, like the Clolesky's scheme, for the solution of the linear system involved into the computation of the gain vector $\mathbf{G}(m,n)$. Thus, the computational complexity of the original 2D USW LS scheme is $C_{USW} = 2pqkl + k^2l^2pq + k^3l^3/6$. This figure can be reduced by taken into account spatial shift invariance properties that the 2D regressor vector possess.

3.1 Fast inverse covariance estimation

Let us consider the data vector $\mathbf{x}(m,n)$ that carries the space samples that correspond to the projection support region $\mathcal{R}_A$, $\mathbf{x}(m,n) = [\underline{\mathbf{x}}^T(m,n)\ \underline{\mathbf{x}}^T(m-1,n)\ \ldots\ \underline{\mathbf{x}}^T(m-l+1,n)]$, where, $\underline{\mathbf{x}}_k(m-i,n) = [x(m-i,n)\ x(m-i,n-1)\ \ldots\ x(m-i,n-k+1)]^T$, $i = 0,1\ldots k-1$. Then, the data matrix $\boldsymbol{\mathcal{X}}(m,n)$, described by eq. (3), can alternatively be organized in terms of $\mathbf{x}(m,n)$, as $\boldsymbol{\mathcal{X}}(m,n) = [\underline{\chi}^T(m,n)\ \underline{\chi}^T(m-1,n)\ \ldots \underline{\chi}^T(m-q+1,n)]^T$. Entries $\underline{\chi}(m-i,n)$, $i = 0,1\ldots q-1$, are defined as $\underline{\chi}(m-i,n) = [\mathbf{x}^T(m-i,n)\ \mathbf{x}^T(m-i,n-1)\ldots \mathbf{x}^T(m-i,n-p+1)\ ]^T$. Using the above, $\mathbf{R}(m,n)$, is equivalently expressed as $\mathbf{R}(m,n) = \delta\mathbf{I} + \sum_{j=0}^{p-1}\sum_{i=0}^{q-1}\mathbf{x}(m-i,n-j)\mathbf{x}^T(m-i,n-j)$. It can be readily shown that $\mathbf{R}(m,n)$ is recursively estimated according to the following scheme

$$\mathbf{R}(m,n) = \mathbf{R}(m,n-1) - \sum_{i=0}^{q-1}\mathbf{x}(m-i,n-p)\mathbf{x}^T(m-i,n-p) + \sum_{i=0}^{q-1}\mathbf{x}(m-i,n)\mathbf{x}^T(m-i,n)$$

Using the above recursions, an efficient algorithm for the computation of $\mathbf{R}^{-1}(m,n)$, can be developed, [11]. Indeed, let us consider the sequence of matrices $\mathbf{R}_i(m,n)$, $i = 1,2\ldots q$, defined in a recursive way, as

$$\hat{\mathbf{R}}_i(m,n) = \mathbf{R}_{i-1}(m,n) - \mathbf{x}(m-i,n-p)\mathbf{x}^T(m-i,n-p)$$

$$\mathbf{R}_i(m,n) = \hat{\mathbf{R}}_i(m,n) + \mathbf{x}(m-i,n)\mathbf{x}^T(m-i,n)$$

The initial conditions of the above recursive scheme, are given by : $\mathbf{R}_0(m,n) = \mathbf{R}(m,n-1)$, and $\mathbf{R}_q(m,n) = \mathbf{R}(m,n)$. Then, successive application of the matrix inversion lemma leads to a recursive estimation of inverse autocorrelation matrix $\mathbf{R}^{-1}(m,n)$. The resulting algorithm is summarized on Table 2. Variables $\mathbf{w}_i(m,n)$ and $\mathbf{v}_i(m,n)$ that appears in Table 2, are defined as follows

$$\hat{\mathbf{R}}_i(m,n)\mathbf{w}_i(m,n) = \mathbf{x}(m-i,n)$$
$$\mathbf{R}_{i-1}(m,n)\mathbf{v}_i(m,n) = \mathbf{x}(m-i,n-p)$$

When the recursive algorithm of Table 2 is utilized for the computation of the inverse matrix $\mathbf{R}^{-1}(m,n)$, the computational complexity of the 2D USW LS adaptive algorithm is reduced to $C_{USW-I} = 2pqkl + (q+1)k^2l^2$. This figure is an improvement over the original cost, C_{USW}. The memory requirements of the method is $O((p+k)\times(q+l))$.

LET $\mathbf{R}_0^{-1}(m,n) = \mathbf{R}^{-1}(m,n-1)$
FOR $i = 1$ TO q, DO

$$\mathbf{v}_i(m,n) = \mathbf{R}_{i-1}^{-1}(m,n)\mathbf{x}(m-i,n-p)$$
$$\alpha_i^v(m,n) = 1 - \mathbf{x}^T(m-i,n-p)\mathbf{v}_i(m,n)$$
$$\hat{\mathbf{R}}_i^{-1}(m,n) = \mathbf{R}_{i-1}^{-1}(m,n) + \frac{\mathbf{v}_i(m,n)\mathbf{v}_i^T(m,n)}{\alpha_i^v(m,n)}$$
$$\mathbf{w}_i(m,n) = \hat{\mathbf{R}}_i^{-1}(m,n)\mathbf{x}(m-i,n)$$
$$\alpha_i^w(m,n) = 1 + \mathbf{x}^T(m-i,n)\mathbf{w}_i(m,n)$$
$$\mathbf{R}_i^{-1}(m,n) = \hat{\mathbf{R}}_i^{-1}(m,n) - \frac{\mathbf{w}_i(m,n)\mathbf{w}_i^T(m,n)}{\alpha_i^w(m,n)}$$

END DO
LET $\mathbf{R}^{-1}(m,n) = \mathbf{R}_q^{-1}(m,n)$

Table 2: Fast covariance inverse estimation

4 2D UEW LS adaptive filtering

A 2D UEW LS adaptive algorithm is derived from eq.(5) setting $\mathbf{T}(m,n) = \mathbf{I}$ and $\mathbf{S}(m,n) = \mathbf{Q}(m,n) - \boldsymbol{\mathcal{X}}^T(m,n)\boldsymbol{\mathcal{X}}(m,n)$. $\mathbf{Q}(m,n)$ is an exponentially fading memory data matrix that is defined recursively as $\mathbf{Q}(m,n) = \lambda\mathbf{Q}(m,n-1) + \sum_{i=0}^{q-1}\lambda^i\mathbf{x}(m-i,n)\mathbf{x}^T(m-i,n)$. This recursion can be used to develop an efficient algorithm for the computation of the inverse matrix $\mathbf{Q}^{-1}(m,n)$, required by the algorithm. Indeed, let us consider the sequence of matrices $\mathbf{Q}_i(m,n)$, $i = 1,2\ldots q$, defined in a recursive way, as

$$\mathbf{Q}_i(m,n) = \mathbf{Q}_{i-1}(m,n) + \lambda^{i-1}\mathbf{x}(m-i,n)\mathbf{x}^T(m-i,n)$$

The initial conditions of the above recursive scheme, are: $\mathbf{Q}_0(m,n) = \lambda\mathbf{Q}(m,n-1)$. Then, sucessive application of the matrix inversion lemma leads to a recursive estimation of inverse autocorrelation matrix $\mathbf{Q}^{-1}(m,n)$. The resulting algorithm is summarized on Table 3. Variables $\mathbf{u}_i(m,n)$ that appears in Table 3, are defined as

$$\mathbf{Q}_{i-1}(m,n)\mathbf{u}_i(m,n) = \mathbf{x}(m-i,n)$$

The computational complexity of the proposed 2D UEW LS adaptive algorithm is $C_{UEW} = 2pqkl + (q+1)k^2l^2/2$. This figure is an improvement over C_{USW-I}. Moreover, the memory requirements is now reduced to $O(pq+kl)$.

$$\mathbf{e}(m,n) = \mathbf{z}(m,n) - \boldsymbol{\mathcal{X}}^T(m,n)\mathbf{C}(m,n-1)$$

$$\mathbf{F}(m,n) = \mathbf{Q}^{-1}(m,n)\mathbf{e}(m,n)$$

$$\mathbf{C}(m,n) = \mathbf{C}(m,n-1) + \alpha(m,n)\mathbf{X}(m,n)\mathbf{F}(m,n)$$

LET $\quad \mathbf{Q}_0^{-1}(m,n) = \mathbf{Q}^{-1}(m,n-1)$
FOR $i = 1$ TO q, DO

$$\mathbf{u}_i(m,n) = \mathbf{Q}_{i-1}^{-1}(m,n)\mathbf{x}(m-i,n)$$

$$\alpha_i^u(m,n) = \lambda^{-i+1} + \mathbf{x}^T(m-i,n)\mathbf{u}_i(m,n)$$

$$\mathbf{Q}_i^{-1}(m,n) = \mathbf{Q}_i^{-1}(m,n) - \frac{\mathbf{u}_i(m,n)\mathbf{u}_i^T(m,n)}{\alpha_i^u(m,n)}$$

END DO
LET $\quad \mathbf{Q}^{-1}(m,n) = \mathbf{Q}_q^{-1}(m,n)$

Table 3: The 2D UEW LS adaptive algorithm and fast covariance inverse update

5 Simulation Results

The performance of the proposed underdetermined 2D adaptive algorithms is investigated in the context of 2D system identification. We consider the following 2D FIR system

$$z(m,m) = \sum_{i=0}^{5}\sum_{j=0}^{5} c_{i,j}x(m-1,n-j) + \eta(m,n)$$

In this case, $p = q = 6$. $\eta(m,n)$ is a white noise, disturbance signal, which corresponds to an $SNR = 30db$. The input data signal, $x(m,n)$, is generated by the following model

$$A(z_1,z_2)x(m,n) = e(m,n)$$

where $A(z_1,z_2) = a(z_1)a(z_2)$, and $a(z) = 1 + a_1z^{-1} + a_2z^{-2}$. The eigenvalue spread of the 2D autocorrelation matrix of the input signal $x(m,n)$ is controlled by the model parameters a_1 and a_2, and it is set in the range of $O(10^3$. The 2D LMS, 2D RLS and the proposed 2D USW LS and 2D UEW LS (k=l=3) adaptive algorithms have been tested, over 256×256 available data, i.e., $M = N = 256$. The MSE plots, which are provided in Figure 3, indicate that the proposed 2D USW LS (curve 3) and 2D UEW LS (curve 4) behaves closely to the 2D-RLS (curve 1) , and outperforms the 2D LMS (curve 2), algorithm, which shows a very slow convergence rate. The tracking ability of the proposed algorithms is illustrated in Figure 4, for the case when the model parameters are arbuptly changed at the middle of the experiment.

5 Conclusions

Two efficient 2D Underdetermined LS adaptive algorithms have been proposed, for 2D filtering system identification. The first method is based on the sliding data window estimation of the projection data covariance matrix. The

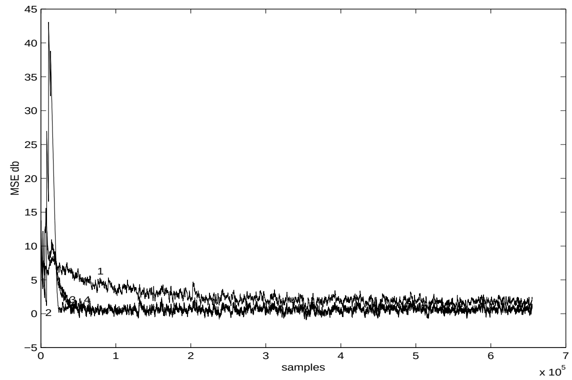

Figure 3: Stationaly model simulation

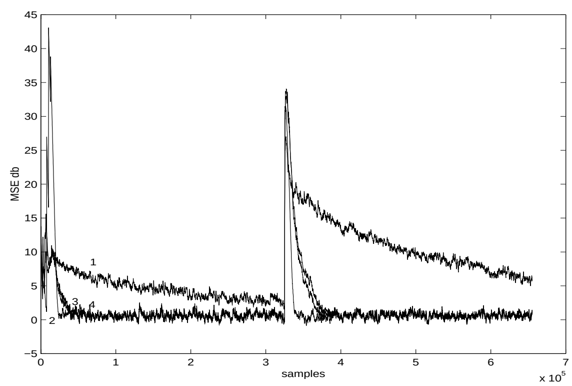

Figure 4: Time-varying model simulation

second method utilize an exponential data window of the corresponding data covariance matrix. The derivation of the proposed algorithm is based on the spatial shift invariant properties the 2D discrete time signals possess. The proposed algorithms have low computational complexity, comparable to that of the 2D LMS algorithm. Simulation results indicate that the convergence speed of the proposed scheme is comparable to the higher complexity 2D RLS algorithms.

Acknowledgment- This work has been partially supported by EPEAEK II, Measure 2.5, Action 2.5.1.

References:

[1] H. Kaufman and A. M. Tekalp, 'Survey of estimation techniques in image restoration,' IEEE Control Systems, pp. 16-23, Jan. 1991.

[2] M. Hadhoud, and D. Thomas, 'The Two-Dimensional adaptive LMS (TDLMS) algorithm,' IEEE Trans. on Circ. Systems, vol. 35, pp. 485-494, May 1988.

[3] Y. Boutalis, S. Kollias and G. Carayannis,' A fast multichannel approach to adaptive image estimation,' Trans. ASSP-37, pp. 1090-1098, 1989.

[4] G. Glentis and N. Kalouptsidis, 'Fast algorithms for 2-D least squares FIR filtering,' IEEE Int. Conf.-ASSP Toronto, 1991.

[5] D. Slock, 'Underdetermined growing and sliding window covariance fast transversal filter RLS algorithms,' Signal Processing VI: Theories and Applications, pp. 1169-1173, 1992.

[6] X. Liu, and M. Najim, 'A Two-Dimensional fast lattice recursive least squares algorithm,' IEEE Trans. on Signal Processing, vol. 44, pp. 2557-2567, Oct. 1996.

[7] M. Muneyasu, and T. Hinamoto, 'A realization of 2-D adaptive filters using affine projection algorithm,' J. Franklin Inst. vol. 335, No. 7, pp. 1185-1193, 1998.

[8] W. Mikhael, and S. Ghosh, 'Two-Dimensional block adaptive filtering algorithms with optimum convergence factors,' IEEE Trans. on Circuits and Systems II, vol. 42, pp. 505-515, May 1995.

[9] M. Azimi-Sadjadi, and H. Pan, 'Two-Dimensional block diagonal LMS adaptive filtering,' IEEE Trans. on Signal Processing, vol. 42, pp. 2420-2429, Sept. 1994.

[10] G.O. Glentis, C.H. Slump, and O.Herrmann, 'A true order recursive algorithm for two-dimensional mean squared error linear prediction and filtering, Singal Processing, vol. 80, pp. 1399-1418, 2000.

[11] A. Tekalp, H. Kaufman, and J. Woods, 'Fast recursive estimation of the parameters of a space-varying autoregressive image model,' IEEE Trans. on ASSP, vol. 33, pp. 469-472, April 1995