A NEW QRD-BASED BLOCK ADAPTIVE ALGORITHM

M. Bhouri, M. Bonnet, M. Mboup

UFR Mathématiques et Informatique - Université René Descartes-Paris 5
45, rue des Saints-Pères, 75270 Paris cedex 06, France
email: bhom@math-info.univ-paris5.fr

ABSTRACT

In this paper we present a new robust adaptive algorithm. It is derived from the standard QR Decomposition based RLS (QRD-RLS) algorithm by introducing a non-orthogonal transform into the update recursion. Instead of updating an upper triangular matrix, as it is the case for the QRD-RLS, we adapt an upper triangular block diagonal matrix. The complexity of the algorithm, thus obtained, varies from $O\left(N^2\right)$ to $O\left(N\right)$ when the size of the diagonal blocks decreases. Simulations of the new algorithm have shown a better robustness than the standard QRD-based algorithm in the context of multichannel adaptive filtering with highly intercorrelated channels.

1. INTRODUCTION

Recursive least squares (RLS) adaptive algorithm is well known for its invariably fast convergence compared to the LMS class algorithms which convergence properties are very sensitive to input correlation. However, this advantage of RLS over LMS is obtained at the expense of a lower robustness and a higher complexity. Indeed, the classical RLS algorithm (or the pseudo-inverse based algorithm) has poor numerical properties and a high complexity of $O\left(N^2\right)$. Fast versions of the RLS class (FTF, FAEST) are derived for adaptive filtering by exploiting the input vector shift invariance structure, resulting in a reduced complexity of order $O\left(N\right)$. Unfortunaly, these versions suffer from serious numerical problems due to error roundoff accumulation.

A more robust version of the RLS adaptive algorithm was the QRD-RLS algorithm. It was introduced first by Gentleman and Kung [5] then promoted by the works of McWhirter [8] and Cioffi [3][4]. Since then, it has received a particular attention over the last decade. This can be attributed essentially to two aspects of the algorithm:

- it has good numerical properties, due to the use of the robust QR decomposition involving the Givens or Householder transformations;
- it is suitable for systolic array implementation which makes it interesting for real time applications on VLSI circuits.

We start this paper by presenting the QRD-RLS algorithm for the general context of adaptive parameter estimation. Then we derive a new algorithm, called 2-block-diagonal algorithm, by reducing the transformed input data matrix to be 2-block-diagonal. This algorithm is then generalized to a p-block-diagonal one. Finally, simulations show the algorithm performances in the context of multichannel adaptive filtering.

2. QRD-RLS ADAPTIVE ALGORITHM

In this section we recall the QRD-RLS. Consider an adaptive parameter estimation setup where the signal reference $d\left(n\right)$ is described as

$$d\left(n\right) = \mathbf{x}^T\left(n\right)\mathbf{w}^{op} + v\left(n\right)$$

where $\mathbf{x}^T\left(n\right) = \left(x_1\left(n\right), x_2\left(n\right), \ldots, x_N\left(n\right)\right)$ is the input vector, $\mathbf{w}^{op}$ is the N-dimension unknown parameter vector and $v\left(n\right)$ is a white noise perturbation.

This is a general model, which includes the multichannel adaptive filtering.

The output error vector is given by

$$\mathbf{e}\left(n\right) = \mathbf{d}\left(n\right) - \mathbf{X}\left(n\right)\mathbf{w}\left(n\right)$$

where $\mathbf{w}\left(n\right)$ is the N-dimension adaptive parameter vector,

$$\mathbf{X}\left(n\right) = \left(\mathbf{x}_1\left(n\right), \mathbf{x}_2\left(n\right), \ldots, \mathbf{x}_N\left(n\right)\right) = \begin{pmatrix} \mathbf{x}^T\left(n\right) \\ \vdots \\ \mathbf{x}^T\left(0\right) \end{pmatrix}$$

with $\mathbf{x}_i^T\left(n\right) = \left(x_i\left(n\right), \ldots, x_i\left(0\right)\right) \quad i = 1, \ldots, N$ and $\mathbf{d}^T\left(n\right) = \left(d\left(n\right), \ldots, d\left(0\right)\right)$. Let $\mathbf{\Lambda}\left(n\right) = \mathrm{diag}\left(1, \lambda, \ldots, \lambda^n\right)$ and define

$$\mathbf{d}_\lambda\left(n\right) \triangleq \mathbf{\Lambda}\left(n\right)\mathbf{d}\left(n\right) = \begin{pmatrix} d\left(n\right) \\ \lambda\mathbf{d}_\lambda\left(n-1\right); \end{pmatrix}$$

$$\mathbf{X}_\lambda\left(n\right) \triangleq \mathbf{\Lambda}\left(n\right)\mathbf{X}\left(n\right) = \begin{pmatrix} \mathbf{x}^T\left(n\right) \\ \lambda\mathbf{X}_\lambda\left(n-1\right) \end{pmatrix} \tag{1}$$

and $\mathbf{e}_\lambda\left(n\right) \triangleq \mathbf{\Lambda}\left(n\right)\mathbf{e}\left(n\right) = \mathbf{d}_\lambda\left(n\right) - \mathbf{X}_\lambda\left(n\right)\mathbf{w}\left(n\right)$.
Then, the least squares parameter estimation problem is the minimization problem:

$$\min_{\mathbf{w}\left(n\right)}\left[\zeta\left(n\right)\right] \tag{2}$$

where

$$\zeta\left(n\right) = \left\|\mathbf{e}_\lambda\left(n\right)\right\|^2 = \left\|\mathbf{Q}^T\left(n\right)\mathbf{e}_\lambda\left(n\right)\right\|^2 \tag{3}$$

for any $\left(\left(n+1\right) \times \left(n+1\right)\right)$ orthogonal matrix $\mathbf{Q}\left(n\right)$. Let choose $\mathbf{Q}\left(n\right)$ such that the transformed vector $\bar{\mathbf{X}}\left(n\right)$ of $\mathbf{X}_\lambda\left(n\right)$ is as follow

$$\bar{\mathbf{X}}\left(n\right) = \mathbf{Q}^T\left(n\right)\mathbf{X}_\lambda\left(n\right) = \begin{pmatrix} \bigcirc \\ \mathbf{R}\left(n\right) \end{pmatrix} \qquad n \geq N-1$$

where $\mathbf{R}\left(n\right)$ is an $\left(N \times N\right)$ upper triangular matrix. Accordingly, the vector $\mathbf{d}_\lambda\left(n\right)$ is transformed by $\mathbf{Q}^T\left(n\right)$ into

$$\bar{\mathbf{d}}\left(n\right) = \mathbf{Q}^T\left(n\right)\mathbf{d}_\lambda\left(n\right) = \begin{pmatrix} \bar{\mathbf{d}}^e\left(n\right) \\ \bar{\mathbf{d}}^w\left(n\right) \end{pmatrix} \qquad n \geq N-1$$

where $\bar{\mathbf{d}}^w(n)$ is a N-dimension vector. The error norm in equation (3) then reads as

$$\zeta(n) = \left\| \begin{pmatrix} \bar{\mathbf{d}}^e(n) \\ \bar{\mathbf{d}}^w(n) - \mathbf{R}(n)\,\mathbf{w}(n) \end{pmatrix} \right\|^2 \qquad n \geq N-1$$

Since $\bar{\mathbf{d}}^e(n)$ is independent of $\mathbf{w}(n)$, the solution of the least squares problem is obtained by annulling the second block:

$$\mathbf{R}(n)\,\mathbf{w}(n) = \bar{\mathbf{d}}^w(n) \qquad n \geq N-1 \tag{4}$$

The inversion of the matrix $\mathbf{R}(n)$ at time n has a complexity of $O\left(N^2\right)$ while its computation may be achieved recursively, from $\mathbf{R}(n-1)$ with $O\left(N^2\right)$ complexity.

Due to the input matrix structure given by (1), we can write

$$\begin{pmatrix} \bigcirc \\ \mathbf{R}(n) \end{pmatrix} = \mathbf{Q}^u(n) \begin{pmatrix} \mathbf{x}^T(n) \\ \bigcirc \\ \lambda \mathbf{R}(n-1) \end{pmatrix} \qquad n \geq N \tag{5}$$

with $\mathbf{Q}^u(n) = \mathbf{Q}^T(n) \begin{pmatrix} 1 & \mathbf{0}^T \\ \mathbf{0} & \mathbf{Q}(n-1) \end{pmatrix}$

It comes that the updation of $\mathbf{R}(n-1)$ begins by adding a new row to the transformed input matrix, then zeroing this row by mean of the orthogonal transform $\mathbf{Q}^u(n)$ to obtain the updated $\mathbf{R}(n)$ matrix. The matrix $\mathbf{Q}^u(n)$ is composed of N rotations (Givens) or reflections (Householder) [6]

The updation matrix $\mathbf{Q}^u(n)$ is also used to compute $\bar{\mathbf{d}}(n)$

$$\bar{\mathbf{d}}(n) = \begin{pmatrix} \bar{\mathbf{d}}^e(n) \\ \bar{\mathbf{d}}^w(n) \end{pmatrix} = \mathbf{Q}^u(n) \begin{pmatrix} d(n) \\ \lambda \bar{\mathbf{d}}^e(n-1) \\ \lambda \bar{\mathbf{d}}^w(n-1) \end{pmatrix} \qquad n \geq N \tag{6}$$

The overall updation step has a complexity of $O\left(N^2\right)$. Since the $n-N$ rows (from second to $(n-N+1)^{th}$) are not affected by the $\mathbf{Q}^u(n)$ matrix, one may use a reduced $((N+1) \times (N+1))$ orthogonal matrix $\mathbf{Q}^v(n)$ to update the following quantities :

$$\begin{aligned} \begin{pmatrix} \mathbf{0}^T \\ \mathbf{R}(n) \end{pmatrix} &= \mathbf{Q}^v(n) \begin{pmatrix} \mathbf{x}^T(n) \\ \lambda \mathbf{R}(n-1) \end{pmatrix} \\ \begin{pmatrix} \bar{d}_1^e(n) \\ \bar{\mathbf{d}}^w(n) \end{pmatrix} &= \mathbf{Q}^v(n) \begin{pmatrix} d(n) \\ \lambda \bar{\mathbf{d}}^w(n-1) \end{pmatrix}; \qquad n \geq N \end{aligned} \tag{7}$$

Finally, the QRD-RLS algorithm with a *soft-constrained* initialization scheme is summarized by

> 1- Initialization
> $\mathbf{R}(-1) = \mu \mathbf{I}_N$
> $\bar{\mathbf{d}}^w(-1) = \mu \mathbf{w}(-1)$
>
> for $n = 0 : L$
>
> 2- $\begin{pmatrix} \mathbf{0}^T \\ \mathbf{R}(n) \end{pmatrix} = \mathbf{Q}^v(n) \begin{pmatrix} \mathbf{x}^T(n) \\ \lambda \mathbf{R}(n-1) \end{pmatrix}$
> $\begin{pmatrix} \bar{d}_1^e(n) \\ \bar{\mathbf{d}}^w(n) \end{pmatrix} = \mathbf{Q}^v(n) \begin{pmatrix} d(n) \\ \lambda \bar{\mathbf{d}}^w(n-1) \end{pmatrix}$
>
> 3- $\mathbf{w}(n) = \mathbf{R}^{-1}(n)\,\bar{\mathbf{d}}^w(n)$
>
> 4- $e(n) = d(n) - \mathbf{x}^T(n)\,\mathbf{w}(n)$

3. A NEW QRD-BASED ADAPTIVE ALGORITHM

In this section we propose a new 2-block-diagonal algorithm, which will be extended later to a general p-block-diagonal algorithm $(1 \leq p \leq N)$. In order to make easier the presentation of the new algorithm, we maintain the same names for the variables involved in the QRD-RLS.

3.1. The 2-block-diagonal Algorithm

The main idea of our algorithm is to introduce a regularizing (non-orthogonal) transform to a QRD-RLS like updation scheme. The triangular matrix $\mathbf{R}(n)$ is partitioned into four blocks

$$\mathbf{R}(n) = \begin{pmatrix} \mathbf{R}_1(n) & \mathbf{B}_{12}(n) \\ \bigcirc & \mathbf{R}_2(n) \end{pmatrix}; \qquad N = N_1 + N_2$$

where $\mathbf{R}_1(n)$ (respectively $\mathbf{R}_2(n)$) is a $N_1 \times N_1$ (respectively $N_2 \times N_2$) upper triangular matrix , and $\mathbf{B}_{12}(n)$ is a $N_1 \times N_2$ matrix. The updation equations of the algorithm are very similar to those of the QRD-RLS. We start by the computation of the parameter vector $\mathbf{w}(n)$ solution of $\mathbf{R}(n)\,\mathbf{w}(n) = \bar{\mathbf{d}}^w(n)$. Then this equation is transformed via a non orthogonal transform into

$$\mathbf{R}'(n)\mathbf{w}(n) = \bar{\mathbf{d}}^{w\prime}(n). \tag{8}$$

$\mathbf{R}'(n)$ and $\bar{\mathbf{d}}^{w\prime}(n)$ are then used in the update equations similarly to the QRD-RLS update equation

$$\begin{aligned} \begin{pmatrix} \mathbf{0}^T \\ \mathbf{R}(n+1) \end{pmatrix} &= \mathbf{Q}^v(n) \begin{pmatrix} \mathbf{x}^T(n+1) \\ \lambda \mathbf{R}'(n) \end{pmatrix} \\ \begin{pmatrix} \bar{d}_1^e(n+1) \\ \bar{\mathbf{d}}^w(n+1) \end{pmatrix} &= \mathbf{Q}^v(n) \begin{pmatrix} d(n+1) \\ \lambda \bar{\mathbf{d}}^{w\prime}(n) \end{pmatrix} \end{aligned}$$

The transformation from (4) to (8) aims to reduce the matrix $\mathbf{R}(n)$ to its block-diagonal submatrix. The extra-diagonal block $\mathbf{B}_{12}(n)$ is moved to the right hand side of equation (4):

$$\begin{aligned} \mathbf{R}'(n) &= \begin{pmatrix} \mathbf{R}_1(n) & \bigcirc \\ \bigcirc & \mathbf{R}_2(n) \end{pmatrix} \\ \bar{\mathbf{d}}^{w\prime}(n) &= \bar{\mathbf{d}}^w(n) - \begin{pmatrix} \bigcirc & \mathbf{B}_{12}(n) \\ \bigcirc & \bigcirc \end{pmatrix} \mathbf{w}(n) \end{aligned}$$

The overall algorithm using a *soft-constrained* initialization scheme is summarized by

> 1- Initialization
> $\mathbf{R}'(-1) = \mu \mathbf{I}_N$
> $\bar{\mathbf{d}}^{w\prime}(-1) = \mu \mathbf{w}(-1)$
>
> for $n = 0 : L$
>
> 2- $\begin{pmatrix} \mathbf{0}^T \\ \mathbf{R}(n) \end{pmatrix} = \mathbf{Q}^v(n) \begin{pmatrix} \mathbf{x}^T(n) \\ \lambda \mathbf{R}'(n-1) \end{pmatrix}$
> $\mathbf{R}(n) = \begin{pmatrix} \mathbf{R}_1(n) & \mathbf{B}_{12}(n) \\ \bigcirc & \mathbf{R}_2(n) \end{pmatrix}$
> $\begin{pmatrix} \bar{d}_1^e(n) \\ \bar{\mathbf{d}}^w(n) \end{pmatrix} = \mathbf{Q}^v(n) \begin{pmatrix} d(n) \\ \lambda \bar{\mathbf{d}}^{w\prime}(n-1) \end{pmatrix}$
>
> 3- $\mathbf{w}(n) = \mathbf{R}^{-1}(n)\,\bar{\mathbf{d}}^w(n)$
>
> 4- $\mathbf{R}'(n) = \begin{pmatrix} \mathbf{R}_1(n) & \bigcirc \\ \bigcirc & \mathbf{R}_2(n) \end{pmatrix}$
> $\bar{\mathbf{d}}^{w\prime}(n) = \bar{\mathbf{d}}^w(n) - \begin{pmatrix} \bigcirc & \mathbf{B}_{12}(n) \\ \bigcirc & \bigcirc \end{pmatrix} \mathbf{w}(n)$
>
> 5- $e(n) = d(n) - \mathbf{x}^T(n)\,\mathbf{w}(n)$

One can show that the submatrix $\mathbf{B}_{12}(n)$ is a rank one matrix , which does not need to be explicitly computed in step 2. Therefore, the computation of $\bar{\mathbf{d}}^{w\prime}(n)$ has a complexity of $O(N)$. Thus, step 3 can be computed in conjuction with step 4, with a lower complexity than the corresponding steps in the QRD-RLS algorithm which implies a gain for the global complexity as compared to the QRD-RLS. Thus, steps 3 and 4 are replaced by the equivalent step 3':

$$\begin{aligned}
\mathbf{w}_2(n) &= \mathbf{R}_2^{-1}(n)\bar{\mathbf{d}}_2^{w}(n) \\
\bar{\mathbf{d}}_1^{w\prime}(n) &= \bar{\mathbf{d}}_1^{w}(n) - \mathbf{B}_{12}(n)\mathbf{w}_2(n) \\
\mathbf{w}_1(n) &= \mathbf{R}_1^{-1}(n)\bar{\mathbf{d}}_1^{w\prime}(n) \\
\bar{\mathbf{d}}^{w\prime}(n) &= \begin{pmatrix} \bar{\mathbf{d}}_1^{w\prime}(n) \\ \bar{\mathbf{d}}_2^{w}(n) \end{pmatrix}, \qquad \mathbf{w}(n) = \begin{pmatrix} \mathbf{w}_1(n) \\ \mathbf{w}_2(n) \end{pmatrix}
\end{aligned}$$

Indeed, if we denote by $c_{QR}(N) = O(N^2)$ the complexity of the QRD-RLS, then the complexity [2] of our algorithm, denoted c_{ALG} is

$$c_{ALG}(N_1, N_2) = c_{QR}(N_1) + c_{QR}(N_2) + O(N)$$

It follows that

$$c_{ALG}(N_1, N_2) < c_{QR}(N_1 + N_2) \tag{9}$$

Moreover, it is shown in [2] that the transformation of the step 4 from (4) to (8) improves the conditioning of the system. Consequently, our algorithm can be seen as a preconditioned iterative algorithm.

3.2. The p-block-diagonal Algorithm

In the previous subsection we have presented the 2-block-diagonal algorithm, based on a 2×2-block partitionning of the triangular matrix $\mathbf{R}(n)$. The algorithm is then characterized by the couple (N_1, N_2) of the sizes of the 2 triangular submatrices $\mathbf{R}_1(n)$ and $\mathbf{R}_2(n)$. We now show that this can be extended to a general partition of the block diagonal of $\mathbf{R}(n)$ into p triangular $(N_k \times N_k)$ submatrices $\mathbf{R}_k(n)$ $(1 \leq k \leq p)$,

so that $\mathbf{R}(n)$ has the following form $(1 \leq p \leq N)$

$$\begin{pmatrix}
\boxed{\mathbf{R}_1(n)} & \boxed{\mathbf{B}_{1p}(n)} & & \\
 & \boxed{\mathbf{R}_2(n)} & \boxed{\mathbf{B}_{2p}(n)} & \\
 & & \ddots & \\
 & & \boxed{\mathbf{R}_{p-1}(n)} & \boxed{\mathbf{B}_{p-1,p}(n)} \\
\bigcirc & & & \boxed{\mathbf{R}_p(n)}
\end{pmatrix}$$

This defines a larger class of algorithms. The corresponding algorithm will be characterized by the p-uplet $(N_1, N_2, \ldots, N_p)$ of the sizes of the p diagonal blocks of $\mathbf{R}(n)$. We will refer to this algorithm as the $(N_1, N_2, \ldots, N_p)$-algorithm .

The 2-block-diagonal algorithm is extended to the p-block-diagonal one by replacing the update equations (step 3), by

$$\begin{aligned}
\mathbf{R}'(n) &= \begin{pmatrix} \mathbf{R}_1(n) & & \bigcirc \\ & \ddots & \\ \bigcirc & & \mathbf{R}_p(n) \end{pmatrix} \qquad (10) \\
\bar{\mathbf{d}}^{w\prime}(n) &= \bar{\mathbf{d}}^{w}(n) - \mathbf{B}(n)\,\mathbf{w}(n)
\end{aligned}$$

with $\mathbf{B}(n) = \begin{pmatrix} \bigcirc & \boxed{\mathbf{B}_{1p}(n)} \\ & \vdots \\ & \boxed{\mathbf{B}_{p-1,p}(n)} \\ \bigcirc & \bigcirc \end{pmatrix}$

Under this general formulation, we notice that the 1-block-diagonal algorithm is the QRD-RLS algorithm and the N-block-diagonal algorithm is the fast QR algorithm presented by Liu [7]. It can be shown that the product $\mathbf{B}(n)\,\mathbf{w}(n)$ requires a complexity of $O(N)$. Then, for similar reasons as those for 2-block-diagonal algorithm (for more details see [2]), the complexity of the $(N_1, N_2, \ldots, N_p)$-algorithm is

$$c_{ALG}(N_1, \ldots, N_p) = c_{QR}(N_1) + \ldots + c_{QR}(N_p) + O(N)$$

The lowest complexity, equal to $O(N)$, is reached for $p = N$ (the (1,1,. . .,1)-algorithm corresponding to Liu's algorithm), whereas the highest complexity is for $p = N$ (QRD-RLS)

The orthogonal transform of step 2 can be realized by using either Givens or Householder transformations. In the following section, we have only simulated the Householder algorithm version.

Another algorithm class, using a similar matrix reduction approach and more suited to fast implementation , is presented in [1], it has a reduced complexity of $O(N)$.

4. SIMULATIONS

This section presents some examples of the behaviour of the proposed algorithm in a 2-channel adaptive filtering context, through 100 runs Monte Carlo simulations. The two channels are represented by two tap filters of length 6 and 4, respectively. Accordingly, the input vector is of the form

$$\mathbf{x}^T(n) = (x_1(n), \ldots, x_1(n-5), x_2(n), \ldots, x_2(n-3)),$$

where $x_1(n)$ and $x_2(n)$ are the channels input samples. The performance of our algorithm is compared below to that of the LMS, the QRD-RLS and Liu's algorithm. At time $n = 300$ the two channels initial parameters are switched to another set of parameters. This allows us to see the behaviour of the algorithms in a tracking situation.

Figure 1 compares our (6,4)-algorithm with Liu's algorithm, in the case of two input colored noise $x_1(n)$ and $x_2(n)$ highly intercorrelated with an output SNR of 40dB. We note that our algorithm converges faster than the Liu's algorithm.

In figures 2 and 3, we compare the ability of the LMS, the QRD-RLS and our algorithm to identify the two channels when the input covariance matrix is (nearly) rank deficient. For this, the channels input signals $x_1(n)$ and $x_2(n)$ are two coloured noises with a very high intercorrelation.

The output mean square error (MSE) is plotted in figure 2 v.s. the number of iterations (time). The overall output MSE decreases for each algorithm (with very slow convergence for the LMS). However, although the QRD-RLS exhibits the fastest decrease for the output MSE, the observation of figure 3 (which represents the normalized parameter vector deviation $\frac{\|\mathbf{w}(n)-\mathbf{w}^{op}\|}{\|\mathbf{w}^{op}\|}$), shows that the corresponding filter is not BIBO (Bounded Input Bounded Output) stable. This is due to the rank deficiency of the input vector covariance matrix. On the other hand, our algorithm is, as for the LMS, insensitive to such a rank deficiency case.

5. CONCLUSION

We have proposed a class of algorithms derived from the QRD-RLS algorithm, these algorithms inherit the fast convergence property of the RLS algorithms. This class is well suited for multichannel adaptive filtering, as in stereophonic acoustic echo cancellation, where degenerency situations (rank deficiency of the input covariance matrix due to highly cross-correlated input channels) are often encountered. This class extends the algorithm presented by Liu (see [7]). Not only, it preserves Liu's algorithm insensitiveness to the above mentioned degenerency, but it also converges faster. The complexity of this new class ranges between that of [7] and that of QRD-RLS algorithm.

6. REFERENCES

[1] M. Bhouri, M. Mboup, M. Bonnet A fast multichannel adaptive algorithm. submitted to the EUSIPCO-98, 1998.

[2] M. Bhouri. Bloc-QR iteratives algorithms for adaptive filtering. in preparation

[3] J. Cioffi. A fast QR frequency-domain RLS adaptive filter. In *Proc. ICASSP*, pages 407–410, Apr. 1987.

[4] J. Cioffi. The fast adaptive ROTOR's RLS algorithm. *IEEE Trans. ASSP*, 38:631–653, 1990.

[5] W. M. Gentleman and H. T. Kung. Matrix triangularization by systolic arrays. In *SPIE Proceedings*, volume 298, pages 19–26, 1982.

[6] G. H. Golub and C. F. V. Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, MD, second edition, 1989.

[7] Z.-S. Liu. QR methods of $O(n)$ complexity in adaptive parameter estimation. *IEEE Trans. SP*, SP-43:720–729, 1995.

[8] J. McWhirter. Recursive least squares minimization using a systolic array. In *Proc. SPIE*, volume 431, pages 105–112, 1983.

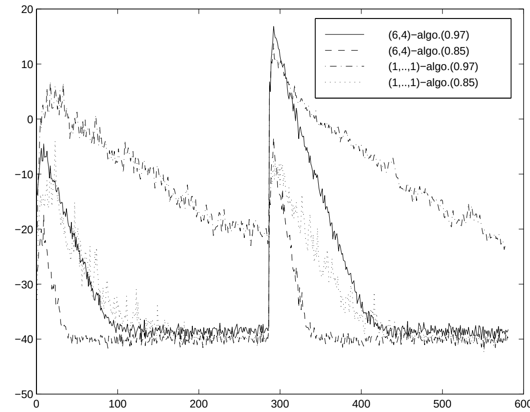

Figure 1: MSE vs. time for (6,4)- and (1,1,..,1)-algorithm

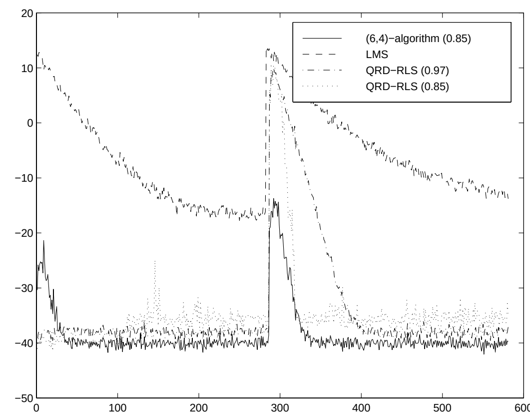

Figure 2: MSE vs. time for QRD-RLS, LMS and (6,4)-algorithm

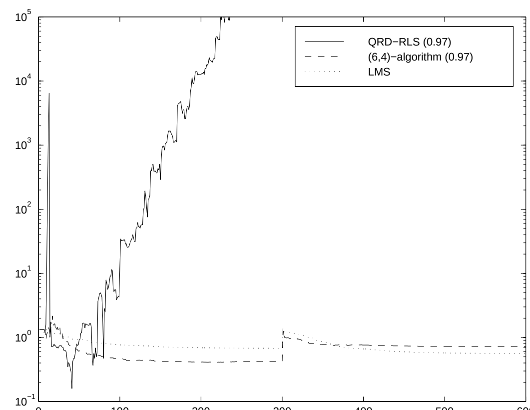

Figure 3: normalized parameter vector deviation vs. time for QRD-RLS, LMS and (6,4)-algorithm