NEW FAST INVERSE QR LEAST SQUARES ADAPTIVE ALGORITHMS

A.A. Rontogiannis[1] *and S. Theodoridis*[2]

[1]Dept. of Informatics
Division of Communications and Signal Processing
University of Athens
[2]Computer Technology Institute
Patras, Greece

ABSTRACT

This paper presents two new, closely related adaptive algorithms for LS system identification. The starting point for the derivation of the algorithms is the inverse Cholesky factor of the data correlation matrix, obtained via a QR decomposition (QRD). Both are of $O(p)$ computational complexity with p being the order of the system. The first algorithm is a fixed order QRD scheme with enhanced parallelism. The second is a lattice type algorithm based on Givens rotations, with lower complexity compared to previously derived ones.

1. INTRODUCTION

Adaptive least squares algorithms for system identification [1], are popular due to their fast converging properties and are used in a variety of applications, such as channel equalization, echo cancellation, spectral analysis, control, to name but a few. Among the various efficiency issues characterizing the performance of an algorithm those of computational complexity, parallelism and numerical robustness are of particular importance, especially in applications where medium to long filter lengths are required. Sometimes it may be preferable to use an algorithm of higher complexity but with good numerical error robustness, since this may allow its implementation with shorter wordlenghts and fixed point arithmetic. This has led to the development of a class of adaptive algorithms based on the numerically robust QR factorization of the input data matrix via the Givens rotation approach.

The development of Givens rotation based QRD algorithms has evolved along three basic directions. Schemes of $O(p^2)$ complexity per time iteration were the first to be derived, with p being the order of the system [2],[3]. These schemes update the Cholesky factor of the input data correlation matrix and can efficiently be implemented on two dimensional systolic arrays. Furthermore, they can provide the modeling error directly, without it being necessary to compute explicitly the estimates of the transversal parameters of the unknown FIR system [3]. An alternative $O(p^2)$ RLS scheme was recently described and it is based on the update of the inverse Cholesky factor of the data correlation matrix [8].

The other category of Givens rotation based algorithms is of the lattice, order recursive type, exhibiting $O(p)$ complexity per time iteration [1],[4]. As with all LS lattice structures, these algorithms compute the modeling LS error for all intermediate orders in a pipelined fashion. The algorithms of the third class also compute the modeling error directly but they lack the pipelined property of the lattice type algorithms [5],[6]. On the other hand, they have lower complexity, compared to their lattice counterparts and they are appropriate for fixed order modeling. This is basically due to the fact that a set of rotation parameters (corresponding to the reflection coefficients) are generated backwards in order, starting from that with the maximum order [1].

In this paper two new, numerically robust, QRD algorithms based on Givens rotations are introduced. The first is of the latter type, i.e., fixed order, direct error computing algorithm. It has similar complexity but it offers enhanced parallelism compared to previously derived ones of the same category. Thus, if two processors are used the computation time is almost halved. The other algorithm is of the lattice type with the same complexity as that of the fixed order one. Therefore, a substantial savings is accomplished compared to already known QR lattice schemes.

The paper is organized as follows. Section 2 briefly reviews the application of the QR decomposition method to the RLS problem. The new fixed order direct error computing algorithm is then derived in section 3. A modification of this algorithm leads to a lattice type scheme. Simulation results are provided in section 4 while section 5 concludes this work. For clarity of presentation real signals are considered throughout this paper. We mostly adopt the notation that appears in [1].

2. THE QR DECOMPOSITION METHOD

The QR decomposition approach to the least squares problem can be expressed as follows ([1])

$$Q_p(N)U_p(N) = \begin{bmatrix} \tilde{R}_p(N) \\ \\ \mathbf{0} \end{bmatrix}$$

where $Q_p(N)Q_p^T(N) = \mathbf{I}$ and $U_p(N)$ is the weighted $N \times p$ input data matrix. $\tilde{R}_p(N)$ is a $p \times p$ upper triangular factor. It is by now well known that the optimum least squares

0-7803-2431-5/95 $4.00 © 1995 IEEE

coefficients' vector $\mathbf{c}_p(N)$ can be obtained by solving the following system of equations

$$\tilde{R}_p(N)\mathbf{c}_p(N) = \mathbf{p}_p(N)$$

where $\mathbf{p}_p(N)$ is the upper $p \times 1$ part of the vector $Q_p(N)\mathbf{y}(N)$, $\mathbf{y}(N)$ being the weighted "desired" vector.
The efficient update of the factor $\tilde{R}_p(N)$ is at the heart of our problem. It has been shown that ([1])

$$Q_p(N)\begin{bmatrix} \lambda^{1/2}\tilde{R}_p(N-1) \\ \mathbf{u}_p^T(N) \end{bmatrix} = \begin{bmatrix} \tilde{R}_p(N) \\ \mathbf{0}^T \end{bmatrix}$$

$\hat{Q}_p(N)$ results from a sequence of basic Givens rotations which successively annihilate the elements of the new data vector $\mathbf{u}_p^T(N) = [u(N), u(N-1), \ldots, u(N-p+1)]$ against $\lambda^{1/2}\tilde{R}_p(N-1)$. λ is the usual forgetting factor. At the same time, it is most interesting that

$$\hat{Q}_p(N)\begin{bmatrix} \lambda^{1/2}\mathbf{p}_p(N-1) \\ y(N) \end{bmatrix} = \begin{bmatrix} \mathbf{p}_p(N) \\ \tilde{e}_p(N) \end{bmatrix} \quad (1)$$

In the last equation, $\tilde{e}_p(N)$ is the so-called angle normalized error which is related to the prior error $e_p(N)$ by

$$\tilde{e}_p(N) = e_p(N)\tilde{a}_p(N) \quad (2)$$

and $\tilde{a}_p(N)$ is the square root of the likelihood related variable $a_p(N)$ ([1]).
In the following section two fast algorithms are described. The first is of the fixed order type computing directly the error $e_p(N)$. A modification of this leads to an order recursive (lattice type) scheme for direct error computation.

3. DERIVATION OF THE NEW ALGORITHMS

In contrast to previously derived fast QRD algorithms ([1], [5], [6]) our starting point is the vector

$$\mathbf{g}_p(N) = \frac{\tilde{R}_p^{-T}(N-1)\mathbf{u}_p(N)}{\sqrt{\lambda}} \quad (3)$$

The essence behind any fast fixed order $O(p)$ scheme is to be able to circumvent the time update of a matrix by updating a vector quantity instead. As we shall see $\mathbf{g}_p(N)$ is such a vector, whose time update provides all necessary rotation angles.

3.1. Time update of $\mathbf{g}_p(N)$

It is easily shown that the factor $\tilde{R}_{p+1}(N)$ can be written in the following forms

$$\tilde{R}_{p+1}(N) = \begin{bmatrix} \tilde{R}_p(N) & \mathbf{p}_p^b(N) \\ \mathbf{0}^T & \tilde{a}_p^b(N) \end{bmatrix} \equiv \begin{bmatrix} \tilde{a}_0^f(N) & \mathbf{z}^T \\ \mathbf{0} & R \end{bmatrix} \quad (4)$$

where $\mathbf{p}_p^b(N)$ and $\tilde{a}_p^b(N)$ are quantities that stem from the backward prediction problem and R is the lower right triangular $p \times p$ part of $\tilde{R}_{p+1}(N)$. From (4) we have

$$\tilde{R}_{p+1}^{-1}(N) = \begin{bmatrix} \tilde{R}_p^{-1}(N) & -\frac{1}{\tilde{a}_p^b(N)}\tilde{R}_p^{-1}(N)\mathbf{p}_p^b(N) \\ \mathbf{0}^T & \frac{1}{\tilde{a}_p^b(N)} \end{bmatrix} \quad (5)$$

$$\tilde{R}_{p+1}^{-1}(N) = \begin{bmatrix} \frac{1}{\tilde{a}_0^b(N)} & -\frac{1}{\tilde{a}_0^b(N)}\mathbf{z}^T R^{-1} \\ \mathbf{0} & R^{-1} \end{bmatrix} \quad (6)$$

By exploiting the relation between R and $\tilde{R}_p(N-1)$ we shall end up with a step up step down update procedure for $\mathbf{g}_p(N)$. Indeed, if we consider the forward prediction problem then, after some manipulations, we can write the equation

$$\hat{Q}_p^f(N)\begin{bmatrix} \tilde{a}_p^f(N) & \mathbf{0}^T \\ \mathbf{p}_p^f(N) & \tilde{R}_p(N-1) \end{bmatrix} = \tilde{R}_{p+1}(N) \quad (7)$$

where $\hat{Q}_p^f(N)$ is a sequence of Givens rotations that successively annihilate the elements of $\mathbf{p}_p^f(N)$. The procedure starts from the last element of $\mathbf{p}_p^f(N)$ and proceeds upwards. The required relation is now available. If $\hat{Q}_p^f(N)$ is partitioned as

$$\hat{Q}_p^f(N) = \begin{bmatrix} q_1 & \mathbf{q}_2^T \\ \mathbf{q}_3 & Q \end{bmatrix} \quad (8)$$

then combination of (4),(7) and (8) results in

$$R = Q\tilde{R}_p(N-1) \quad (9)$$

$$\mathbf{z}^T = \mathbf{q}_2^T\tilde{R}_p(N-1) \quad (10)$$

Joining together equations (3),(6),(9),(10) and the input vector partition

$$\mathbf{u}_{p+1}(N+1) = [u(N+1), \mathbf{u}_p^T(N)]^T$$

we obtain the following time and order update formula for $\mathbf{g}_p(N)$

$$\begin{bmatrix} r_p \\ \mathbf{g}_p(N) \end{bmatrix} = (\hat{Q}_p^f(N))^T\mathbf{g}_{p+1}(N+1) \quad (11)$$

Note, that the first element of $\mathbf{g}_{p+1}(N+1)$, $r_0 = \frac{u(N+1)}{\tilde{a}_0^b(N)\sqrt{\lambda}}$, is known at time $N+1$. Furthermore if we combine equations (3) and (5) and the input partition

$$\mathbf{u}_{p+1}(N+1) = [\mathbf{u}_p^T(N+1), u(N-p+1)]^T$$

we get

$$\mathbf{g}_{p+1}(N+1) = \begin{bmatrix} \mathbf{g}_p(N+1) \\ k_{p+1} \end{bmatrix} \quad (12)$$

where k_{p+1} is just the last element of $\mathbf{g}_{p+1}(N+1)$. The update of $\mathbf{g}_p(N)$ in $O(p)$ is now straightforward from (11) and (12).

Algorithm	Add's	Mult's	Div's/SQRT's
QR-Lattice	$8p$	$27p+1$	$6p$
Fast QRD	$8p+2$	$20p+3$	$6p+4$
New	$8p+1$	$19p+3$	$7p+3$

Table 1: Comparison of complexities of fast rotation-based algorithms

3.2. Rotation angles update

Having completed the time update of $\mathbf{g}_p(N)$ in $O(p)$ we have all information available to obtain the rotation angle parameters providing the prior error. Indeed, by considering the effect of $\hat{Q}_p^f(N+1)$ on the first column of the matrix in (7) we have (for the next time instant $N+1$)

$$\hat{Q}_p^f(N+1)\begin{bmatrix} \tilde{a}_p^f(N+1) \\ \mathbf{p}_p^f(N+1) \end{bmatrix} = \begin{bmatrix} \tilde{a}_0^f(N+1) \\ \mathbf{0} \end{bmatrix} \tag{13}$$

Vector $\mathbf{p}_p^f(N+1)$ results by applying (1) to the forward prediction problem at time N. Indeed,

$$\hat{Q}_p(N)\begin{bmatrix} \lambda^{1/2}\mathbf{p}_p^f(N) \\ u(N+1) \end{bmatrix} = \begin{bmatrix} \mathbf{p}_p^f(N+1) \\ \tilde{e}_p^f(N+1) \end{bmatrix} \tag{14}$$

where $\tilde{e}_p^f(N+1)$ is the angle-normalized forward error. The square root of the forward energy $\tilde{a}_p^f(N+1)$ can then be computed from the well known formula ([1])

$$\tilde{a}_p^f(N+1) = \sqrt{\lambda a_p^f(N) + [\tilde{e}_p^f(N+1)]^2} \tag{15}$$

Finally, the rotation angles of $\hat{Q}_p(N+1)$ can be calculated by using the equation ([7])

$$\hat{Q}_p(N+1)\begin{bmatrix} -\mathbf{g}_p(N+1) \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ \delta_p(N+1) \end{bmatrix} \tag{16}$$

and the scalar $\delta_p(N+1)$ is the inverse of $\tilde{a}_p(N+1)$, that is

$$\delta_p(N+1) = \frac{1}{\tilde{a}_p(N+1)} \tag{17}$$

Equations (11),(12),(14),(15),(13) and (16) compose the prediction part of our algorithm. However, the prior error at time $N+1$ needs to be calculated. This is accomplished in the filtering part of the algorithm. More specifically, equation (1) at time $N+1$ takes the form

$$\hat{Q}_p(N+1)\begin{bmatrix} \lambda^{1/2}\mathbf{p}_p(N) \\ y(N+1) \end{bmatrix} = \begin{bmatrix} \mathbf{p}_p(N+1) \\ \tilde{e}_p(N+1) \end{bmatrix} \tag{18}$$

The prior error is then given according to (2) and (17) as

$$e_p(N+1) = \tilde{e}_p(N+1)\delta_p(N+1) \tag{19}$$

The algorithm described so far is summarized in figure 1. The complexity of the algorithm is shown in table 1. Its

(Step 1)
$r_0 = \mathbf{g}_{p+1}^{(1)}(N+1) = \frac{u(N+1)}{\sqrt{\lambda}\tilde{a}_0^b(N)} = \frac{u(N+1)}{\sqrt{\lambda}\tilde{a}_0^f(N)}$
for $i = 1:p$,
$\mathbf{g}_p^{(i)}(N+1) = \mathbf{g}_{p+1}^{(i)}(N+1)$;
$\mathbf{g}_{p+1}^{(i+1)}(N+1) = \frac{\mathbf{g}_p^{(i)}(N) - s\phi_i(N) r_{i-1}}{c\phi_i(N)}$;
$r_i = c\phi_i(N) r_{i-1} - s\phi_i(N)\mathbf{g}_{p+1}^{(i+1)}(N+1)$;
end;

(Step 2)
$\tilde{e}_0^f(N+1) = u(N+1)$;
for $i = 1:p$,
$\mathbf{p}_p^{f(i)}(N+1) = \lambda^{1/2} c\theta_i(N)\mathbf{p}_p^{f(i)}(N) + s\theta_i(N)\tilde{e}_{i-1}^f(N+1)$;
$\tilde{e}_i^f(N+1) = c\theta_i(N)\tilde{e}_{i-1}^f(N+1) - \lambda^{1/2} s\theta_i(N)\mathbf{p}_p^{f(i)}(N)$;
end;

(Step 3)
$\tilde{a}_p^f(N+1) = \sqrt{\lambda a_p^f(N) + [\tilde{e}_p^f(N+1)]^2}$;
for $i = p:1$,
$\tilde{a}_{i-1}^f(N+1) = \sqrt{[\tilde{a}_i^f(N+1)]^2 + [\mathbf{p}_p^{f(i)}(N+1)]^2}$;
$c\phi_i(N+1) = \frac{\tilde{a}_i^f(N+1)}{\tilde{a}_{i-1}^f(N+1)}$;
$s\phi_i(N+1) = \frac{\mathbf{p}_p^{f(i)}(N+1)}{\tilde{a}_{i-1}^f(N+1)}$;
end;

(Step 4)
$\delta_0(N+1) = 1$;
for $i = 1:p$,
$\delta_i(N+1) = \sqrt{\delta_{i-1}^2(N+1) + [\mathbf{g}_p^{(i)}(N+1)]^2}$;
$c\theta_i(N+1) = \frac{\delta_{i-1}(N+1)}{\delta_i(N+1)}$;
$s\theta_i(N+1) = \frac{\mathbf{g}_p^{(i)}(N+1)}{\delta_i(N+1)}$;
end;

(Step 5)
$\tilde{e}_0(N+1) = y(N+1)$;
for $i = 1:p$,
$\mathbf{p}_p^{(i)}(N+1) = \lambda^{1/2} c\theta_i(N+1)\mathbf{p}_p^{(i)}(N) + s\theta_i(N+1)$
$\tilde{e}_{i-1}(N+1)$;
$\tilde{e}_i(N+1) = c\theta_i(N+1)\tilde{e}_{i-1}(N+1) - \lambda^{1/2} s\theta_i(N+1)$
$\mathbf{p}_p^{(i)}(N)$;
end;
$e_p(N+1) = \tilde{e}_p(N+1)\delta_p(N+1)$;

Initialisation (Soft-constrained)
$\tilde{a}_0^b(0) = \sqrt{\lambda^p \mu}$, $a_p^f(0) = \lambda^p \mu$, $\mathbf{g}_p(0) = \mathbf{0}^T$, $\mathbf{p}_p^f(0) = \mathbf{0}^T$
$c\theta_i(0) = 1$, $c\phi_i(0) = 1$, $i = 1, 2, \ldots, p$

Figure 1: The new fixed order fast QRD algorithm

$\tilde{e}_0^f(N+1) = u(N+1);$

$\tilde{a}_0^f(N+1) = \sqrt{\lambda a_0^f(N) + [\tilde{e}_0^f(N+1)]^2};$

for $i = 1 : p$,
$\mathbf{p}_p^{f(i)}(N+1) = \lambda^{1/2} c\theta_i(N)\mathbf{p}_p^{f(i)}(N) + s\theta_i(N)\tilde{e}_{i-1}^f(N+1);$
$\tilde{e}_i^f(N+1) = c\theta_i(N)\tilde{e}_{i-1}^f(N+1) - \lambda^{1/2} s\theta_i(N)\mathbf{p}_p^{f(i)}(N);$
$\tilde{a}_i^f(N+1) = \sqrt{\lambda a_i^f(N) + [\tilde{e}_i^f(N+1)]^2};$
$c\phi_i(N+1) = \frac{\tilde{a}_i^f(N+1)}{\tilde{a}_{i-1}^f(N+1)};$
$s\phi_i(N+1) = \frac{\mathbf{p}_p^{f(i)}(N+1)}{\tilde{a}_{i-1}^f(N+1)};$
end;

Figure 2: Combination of steps 2 and 3 of Figure 1

complexity is similar to that of the fast QRD algorithm of [6],[1] (see table 1). However, there is a distinct advantage. Note that steps 1 and 2 as well as steps 3 and 4 of the new algorithm can be performed concurrently. Thus, by using two sets of DSP's the execution time is almost halved. This is not possible with the algorithms of [1], [6] which are sequential for each time iteration.
The algorithm of figure 1 is a fixed order algorithm. As we can easily observe from figure 1 the execution of step 3 starts after step 2 has been completed and $\tilde{a}_p^f(N+1)$ has been calculated. Then, the loop of step 3 goes backwards in order, something that does not comply with the basic "pipeline" concept of a lattice structure. However, steps 2 and 3 of the algorithm can be combined if equation (15) is adopted for the calculation of the forward energies of all orders. Such a modification is presented in figure 2. This leads to a new lattice algorithm with the same complexity as our fixed order scheme. Compared to its previously derived counterparts ([1],[4]) the new lattice algorithm has a substantially lower complexity (table 1).

4. SIMULATIONS

In order to verify the correctness of the derived algorithms a system identification problem was considered. The unknown FIR system was of order 10, the SNR = 30dB, the forgetting factor $\lambda = 0.98$ and the initialization parameter $\mu = 0.01$. Figure 3 shows the obtained error convergence curves. Three curves are overlaid although they are not distinguished. Two correspond to the novel algorithms developed in section 3 and the third to the fast QRD algorithm of [6]. The curves are the average of 200 realizations. Note that experiments with up to 500000 iterations were run with no indication of numerical stability problems.

5. CONCLUSIONS

Two fast inverse QR decomposition based algorithms were developed in this paper. The first is a fixed order QRD scheme for direct error computation with enhanced parallelism. A modification of the scheme leads to a lattice type algorithm with low complexity compared to already existing QRD lattice algorithms. Computer simulations

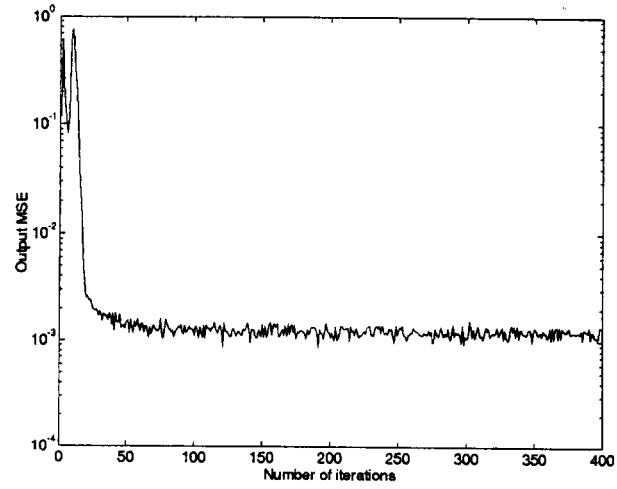

Figure 3: Initial convergence curves

have demonstrated the numerical stability of the proposed algorithms while a comparative numerical accuracy study among the various QRD as well as lattice schemes is currently under investigation.

6. REFERENCES

[1] N. Kalouptsidis and S. Theodoridis, *Adaptive system identification and signal processing algorithms*, Engelwood Cliffs, NJ : Prentice Hall, 1993.

[2] W.M. Gentleman, H.T. Kung, "Matrix triangularization by systolic arrays", *Proc. SPIE, Int. Soc. Opt. Eng.*, vol. 298, 1981.

[3] J.G. McWhirter, "Recursive least squares minimization using a systolic array", *Proc. SPIE, Int. Soc. Opt. Eng.*, vol. 431, pp. 105-112, 1983.

[4] F. Ling, "Givens rotation based least squares lattice and related algorithms", *IEEE Trans. S*, vol. SP-39, no. 7, pp. 1541-1551, July 1991.

[5] J.M. Cioffi, "The fast adaptive ROTOR's RLS algorithm", *IEEE Trans. ASSP*, vol. ASSP-38, no. 4, pp. 631-653, Apr. 1990.

[6] P.A. Regalia, M.G. Bellanger, "On the duality between fast QR methods and lattice methods in least squares adaptive filtering", *IEEE Trans. SP*, vol. SP-39, no. 4, pp. 879-891, Apr. 1991.

[7] C.T. Pan, R.J. Plemmons, "Least squares modifications with inverse factorizations : parallel implications", *J. Compt. Appl. Math.* vol. 27, pp. 109-127, 1989.

[8] S.T. Alexander, A.L. Ghirnikar, "A method for recursive least squares adaptive filtering based upon an inverse QR decomposition", *IEEE Trans. SP*, vol. SP-41, no. 1, pp. 20-30, Jan. 1993.