# FAST QR BASED IIR ADAPTIVE FILTERING ALGORITHM

*Mounir Bhouri*

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 QR based algorithm for IIR adaptive filtering. This algorithm achieves a reduction of complexity with regard to the IIR-QR algorithm by using a block reduction transformation. Moreover, this new approach make it possible to directly transform fast FIR algorithm into fast $O(N)$ versions of the IIR algorithm. Therefore, we derive a fast version of the algorithm from the rotation-based lattice algorithm (QR-LSL). Simulations, have proven the fast convergence and the good numerical properties of both algorithms for systems satisfying the strictly positive real (SPR) condition.

## 1. INTRODUCTION

The finite impulse response FIR structures are widely used in adaptive filters, due to their simplicity. They benefit of an important class of adaptive filtering algorithms. However, the infinite impulse response IIR structures seems to be more suitable to some applications involving long impulse response such as the acoustic echo cancellation. Unfortunately, the complexity of these structures, due to some inherent problems (local minima for output error formulation and biased solution for the equation error), make the derivation of efficient adaptive filtering algorithms rather difficult. The IIR filters generally use iterative gradient based algorithms [8], which have slower convergence than recursive least squares algorithms. A QR least squares adaptive algorithm IIR-QR have been proposed for IIR filtering [4]. It minimizes a modified least squares criterion resulting from a pseudo-linear regression (PLR) approximation [6]. This algorithm converge rapidly and has a reduced complexity (still $O\left(N^2\right)$) compared to iterative algorithms. A fast $O(N)$ version of this algorithm exists, but it is inherently instable [3].

In the following, we present a new IIR adaptive filtering algorithm based on a block decomposition of the transformed data matrix. Then, our QR based IIR algorithm simplifies to two FIR-like adaptive algorithms. It becomes then easy to derive fast $O(N)$ versions of this algorithm. We derive a lattice structure QR-LSL like IIR adaptive filtering algorithm. In the last section, simulations of the algorithm and its fast version are given.

## 2. A QR BASED IIR ADAPTIVE FILTER

The adaptive algorithm tries to minimize the error between the output of the adaptive filter $y(n)$ and the desired signal $d(n)$

$$e(n) = d(n) - y(n)$$

with the output of the filter

$$y(n) = \sum_{i=0}^{N_X-1} a_i(n)\, x(n-i) + \sum_{i=1}^{N_Y} b_i(n)\, y(n-i)$$

where $x(n)$ is the input signal of the adaptive filter and $a_i(n)$ (respectively $b_i(n)$) are the adaptive feedforward (respectively feedback) weights.

We denote, $\mathbf{r}(n)$ the regression vector and $\mathbf{w}(n)$ the weight vector of dimension $N = N_X + N_Y$

$$\begin{aligned} \mathbf{r}^T(n) &= \begin{pmatrix} \mathbf{x}^T(n) & \mathbf{y}^T(n) \end{pmatrix} \\ \mathbf{w}^T(n) &= \begin{pmatrix} \mathbf{w}_X^T(n) & \mathbf{w}_Y^T(n) \end{pmatrix} \end{aligned} \tag{1}$$

with

$$\begin{aligned} \mathbf{x}^T(n) &= \begin{pmatrix} x(n) & \cdots & x(n-N_X+1) \end{pmatrix} \\ \mathbf{y}^T(n) &= \begin{pmatrix} y(n-1) & \cdots & y(n-N_Y) \end{pmatrix} \\ \mathbf{w}_X^T(n) &= \begin{pmatrix} a_0(n) & a_1(n) & \cdots & a_{N_X-1}(n) \end{pmatrix} \\ \mathbf{w}_Y^T(n) &= \begin{pmatrix} b_1(n) & b_2(n) & \cdots & b_{N_Y}(n) \end{pmatrix} \end{aligned}$$

In the least square approach the adaptive filter minimize the following error criterion

$$J(n) = \sum_{k=0}^{n} \lambda^{2(n-k)} \left[ d(k) - \mathbf{r}|_{\mathbf{w}}^T(k)\, \mathbf{w}(n) \right]^2$$

with

$$\begin{aligned}\mathbf{r}|_{\mathbf{w}}^{T}(k) &= \begin{pmatrix} \mathbf{x}^{T}(k) & \mathbf{y}|_{\mathbf{w}}^{T}(k) \end{pmatrix}\\ \mathbf{y}|_{\mathbf{w}}^{T}(k) &= \begin{pmatrix} y|_{\mathbf{w}}(k-1) & \cdots & y|_{\mathbf{w}}(k-N_Y) \end{pmatrix}\end{aligned}$$

and where $y|_{\mathbf{w}}(k)$ is the output of the adaptive filter when the weights are frozen to $\mathbf{w}(n)$

$$y|_{\mathbf{w}}(k) = \sum_{i=0}^{N_X-1} a_i(n)\, x(k-i) + \sum_{i=1}^{N_Y} b_i(n)\, y(k-i)$$

The IIR-QR algorithm use the pseudo-linear regression (PLR) approximation [6]. Thus, it minimizes the following least square criterion

$$\begin{aligned}J'(n) &= \sum_{k=0}^{n} \lambda^{2(n-k)} \left[d(k) - \mathbf{r}^{T}(k)\,\mathbf{w}(n)\right]^{2}\\ &= \left\|\mathbf{d}(n) - \mathbf{X}(n)\,\mathbf{w}(n)\right\|^{2}\end{aligned}$$

with

$$\mathbf{X}(n) = \begin{pmatrix} \mathbf{r}^{T}(n)\\ \lambda\mathbf{r}^{T}(n-1)\\ \vdots\\ \lambda^{n}\mathbf{r}^{T}(0) \end{pmatrix},\ \mathbf{d}(n) = \begin{pmatrix} d(n)\\ \lambda d(n-1)\\ \vdots\\ \lambda^{n} d(0) \end{pmatrix}$$

This is achieved by applying a recursive QR decomposition algorithm to update the triangular factor of the matrix $\mathbf{X}(n)$ [5]

$$\begin{aligned}\begin{pmatrix} \mathbf{0}^{T}\\ \mathbf{R}(n) \end{pmatrix} &= \mathbf{Q}^{v}(n) \begin{pmatrix} \mathbf{x}^{T}(n)\\ \lambda\mathbf{R}(n-1) \end{pmatrix}\\ \begin{pmatrix} \bar{e}(n)\\ \bar{\mathbf{d}}(n) \end{pmatrix} &= \mathbf{Q}^{v}(n) \begin{pmatrix} d(n)\\ \lambda\bar{\mathbf{d}}(n-1) \end{pmatrix}\end{aligned}$$

It follows that, the optimal solution is given by solving the triangular system

$$\mathbf{R}(n)\,\mathbf{w}(n) = \bar{\mathbf{d}}(n) \tag{2}$$

or otherwise expressed by partitioning $\mathbf{R}(n)$ and $\bar{\mathbf{d}}(n)$ (see [1])

$$\begin{pmatrix} \mathbf{R}_X(n) & \mathbf{B}_{XY}(n)\\ \bigcirc & \mathbf{R}_Y(n) \end{pmatrix} \mathbf{w}(n) = \begin{pmatrix} \bar{\mathbf{d}}_X(n)\\ \bar{\mathbf{d}}_Y(n) \end{pmatrix} \tag{3}$$

where $\mathbf{R}_X(n)$ (respectively $\mathbf{R}_Y(n)$) is a $N_X \times N_X$ (respectively $N_Y \times N_Y$ ) upper triangular matrix , $\mathbf{B}_{XY}(n)$ is a $N_X \times N_Y$ matrix and $\bar{\mathbf{d}}_X(n)$ (respectively $\bar{\mathbf{d}}_Y(n)$) is an $N_X$ (respectively $N_Y$) vector. Then by passing $\mathbf{B}_{XY}(n)$ to the right hand side of (3), the system (2) is converted into two reduced complexity subsystems

$$\begin{aligned}\mathbf{R}_X(n)\,\mathbf{w}_X(n) &= \bar{\mathbf{d}}'_X(n)\\ \mathbf{R}_Y(n)\,\mathbf{w}_Y(n) &= \bar{\mathbf{d}}_Y(n)\end{aligned} \tag{4}$$

with

$$\bar{\mathbf{d}}'_X(n) = \bar{\mathbf{d}}_X(n) - \mathbf{B}_{XY}(n)\,\mathbf{w}_Y(n) \tag{5}$$

In addition to this iterative transformation. We propose then, to apply the following updation scheme

$$\begin{aligned}&\mathbf{Q}_X^{v}(n) \begin{pmatrix} \mathbf{x}^{T}(n) & d(n) & \mathbf{y}^{T}(n)\\ \lambda\mathbf{R}_X(n-1) & \lambda\bar{\mathbf{d}}'_X(n-1) & \bigcirc \end{pmatrix} \qquad (6)\\ &= \begin{pmatrix} \mathbf{0}^{T} & \bar{e}_X(n) & \mathbf{y}'^{T}(n)\\ \mathbf{R}_X(n) & \bar{\mathbf{d}}_X(n) & \mathbf{B}_{XY}(n) \end{pmatrix}\end{aligned}$$

$$\begin{aligned}&\mathbf{Q}_Y^{v}(n) \begin{pmatrix} \mathbf{y}^{T}(n) & \bar{d}_Y(n)\\ \lambda\mathbf{R}_Y(n-1) & \lambda\bar{\mathbf{d}}_Y(n-1) \end{pmatrix} \qquad (7)\\ &= \begin{pmatrix} \mathbf{0}^{T} & \bar{e}_Y(n)\\ \mathbf{R}_Y(n) & \bar{\mathbf{d}}_Y(n) \end{pmatrix}\end{aligned}$$

Consequently, these subproblem becomes equivalent to two QR-RLS algorithms driven respectively by the input signals $x(n)$ and $y(n-1)$. The two algorithms are independent in the sense that the rotation based transformations $\mathbf{Q}_X^{v}(n)$ and $\mathbf{Q}_Y^{v}(n)$ are only identified by their corresponding input data $x(n)$ or $y(n-1)$ .

Moreover, we can verify from (6) that $\mathbf{B}_{XY}(n)$ is a rank one matrix, and the $\mathbf{B}_{XY}(n)$ need not to be computed explicitly. Since, the product $\mathbf{B}_{XY}(n)\,\mathbf{w}_Y(n)$ needed in (5) is given by

$$\mathbf{B}_{XY}(n)\,\mathbf{w}_Y(n) = y_Y(n)\,\mathbf{g}_X(n) \tag{8}$$

with

$$\begin{aligned}\begin{pmatrix} \gamma_X(n)\\ \mathbf{g}_X(n) \end{pmatrix} &= \mathbf{Q}_X^{v}(n) \begin{pmatrix} 1\\ \mathbf{0} \end{pmatrix}\\ y_Y(n) &= \mathbf{y}^{T}(n)\,\mathbf{w}_Y(n)\end{aligned}$$

Like the QR-RLS algorithm the output error of the algorithm can be easily computed without the explicit computation of $\mathbf{w}(n)$ as in [7]

$$e(n) = \gamma_Y(n)\,\bar{e}_Y(n)$$

Then, the filter output is computed by

$$y(n) = d(n) - e(n) \tag{9}$$

The desired input signal for the first QR-RLS algorithm (6) is the global desired response $d(n)$, whereas the desired input signal for the second algorithm (7) is $\bar{d}_Y(n) = \bar{e}_X(n)/\gamma_X(n)$. And the filter partial output is given by $y_Y(n) = \bar{d}_Y(n) - e(n)$, this achieves the computation of $\bar{\mathbf{d}}'_X(n)$ using (5) and (8).

The IIR adaptive algorithm, thus obtained is called IIR-BQR as it involves two Block' QR decompositions. It reduces the complexity of the IIR-QR algorithm [4] by approximately a factor of 2. It also reduces the hardware complexity of the parallel architecture required for its implementation by the same factor. Moreover, the major complexity of the new algorithm is retort between the two updation steps of $\mathbf{R}_X(n)$ and $\mathbf{R}_Y(n)$. These steps can be computed independently, and so they can be, efficiently, implemented on a biprocessor architecture.

Furthermore, the IIR-BQR is suited to fast implementation. In fact, the fast QR-RLS algorithms for FIR adaptive filtering can be used to derive fast $O(N)$ IIR adaptive algorithms when using this IIR filtering formulation. A fast algorithm based on the fast QR-RLS algorithm is derived in [2] for the FIR multichannel adaptive filtering, it can be easily modified for our IIR context. In the following, we present a fast lattice IIR algorithm based on the QR-LSL algorithm [9].

## 3. A FAST LATTICE IIR ALGORITHM

The QR-LSL algorithm (Table IV of [9] ) is presented here with a different steps' order. This algorithm will be applied to both subproblems as in equations(6) and (7), therefore, the subscript 'Z' will refer to letters 'X' or 'Y' to differentiate variables relatives to first (X) or second (Y) subproblems. The subproblem algorithms X and Y are first presented in Table-1. Then, we present the necessary additional steps to these two QR-LSL algorithms.

for $n = 1 : L$

$f_{0Z}(n) = b_{0Z}(n) = x_Z(n)$

$e_{0Z}(n) = d_Z(n)$

for $p = 1 : N_Z$

1- $$\mathbf{Q}_2\left(\theta^b_{Z,p}(n)\right)\begin{pmatrix} b_{Z,p-1}(n-1) & e_{Z,p-1}(n-1) \\ \lambda R^b_{Z,p}(n-1) & \lambda\Gamma^e_{Z,p}(n-1)\end{pmatrix} = \begin{pmatrix} 0 & e_{Z,p}(n-1) \\ R^b_{Z,p}(n) & \Gamma^e_{Z,p}(n)\end{pmatrix}$$

2- $\gamma_{Z,p}(n-1) = \gamma_{Z,p-1}(n-1)\cos\theta^b_{Z,p}(n)$

3- $$\mathbf{Q}_2\left(\theta^b_{Z,p}(n)\right)\begin{pmatrix} f_{Z,p-1}(n) \\ \lambda\Gamma^b_{Z,p}(n-1)\end{pmatrix} = \begin{pmatrix} f_{Z,p}(n) \\ \Gamma^b_{Z,p}(n)\end{pmatrix}$$

4- $$\mathbf{Q}_2\left(\theta^f_{Z,p}(n)\right)\begin{pmatrix} f_{Z,p-1}(n) & b_{Z,p-1}(n-1) \\ \lambda R^f_{Z,p}(n-1) & \lambda\Gamma^f_{Z,p}(n-1)\end{pmatrix} = \begin{pmatrix} 0 & b_{Z,p}(n) \\ R^f_{Z,p}(n) & \Gamma^f_{Z,p}(n)\end{pmatrix}$$

*Table-1 QR-LSL algorithm applied to FIR-like part Z (Z≡X or Y)*

where $\mathbf{Q}_2(\theta)$ is the $2\times 2$ rotation matrix of angle $\theta$, and $p$ is the recursive order of the algorithm.

The QR-LSL algorithm [9] is a rotation based LSL algorithm, it is mathematically equivalent to the LSL algorithm [5]. It differs from the standard QR-RLS by its lattice parametrization, so our previous derivation of the algorithm quantities is still valid, except for $\mathbf{r}(n)$ and $\mathbf{w}(n)$. The regression vector is no longer formed of $x(k)$ and $y(k)$ samples, it is denoted $\mathbf{r}'(n)$ and it contains the decoupled $b_{X,p-1}(n-1)$ and $b_{Y,p-1}(n-1)$ samples.

Whereas, the corresponding parameter vector is $\kappa(n)$ such that $\mathbf{r}'^T(n)\,\kappa(n) = \mathbf{r}^T(n)\,\mathbf{w}(n)$ [5]. It follows, that the transformation $\mathbf{Q}^v_Z(n)$ corresponds to $N$ transformations $\mathbf{Q}_2\left(\theta^b_{Z,p}(n)\right), (p = 1,\ldots,N)$, and the vector $\bar{\mathbf{d}}_Z(n)$ corresponds to $\Gamma^e_{Z,p}(n), (p = 1,\ldots,N)$. This leads to the new iterative transformation (similar to equation (5))

$$\Gamma^{e\prime}_{X,p}(n) = \Gamma^e_{X,p}(n) - y_Y(n-1)\,G_{X,p}(n) \tag{10}$$

with

$$\begin{pmatrix} \gamma_{X,p}(n-1) \\ G_{X,p}(n)\end{pmatrix} = \mathbf{Q}_2\left(\theta^b_{Z,p}(n)\right)\begin{pmatrix} \gamma_{X,p-1}(n-1) \\ \mathbf{0}\end{pmatrix}$$

and $y_Y(n-1) = d_Y(n-1) - e(n-1)$

Consequently, the step 1-X[1] (Table-1) of the first algorithm (Z≡X), is transformed as in (6) to step 1'-X

$$\mathbf{Q}^v_X(n)\begin{pmatrix} b_{Z,p-1}(n-1) & e_{Z,p-1}(n-1) \\ \lambda R^b_{Z,p}(n-1) & \lambda\Gamma^{e\prime}_{Z,p}(n-1)\end{pmatrix} = \begin{pmatrix} 0 & e_{Z,p}(n-1) \\ R^b_{Z,p}(n) & \Gamma^e_{Z,p}(n)\end{pmatrix}$$

This first algorithm is initialized by $x_X(n) = x(n)$ and $d_X(n) = d(n)$, while the second algorithm is initialized by $x_Y(n) = y(n-1)$ and $d_Y(n) = e_{X,N_X}(n)/\gamma_{X,N_X}(n)$. This later initialization involves the filter output $y(n-1)$ which is computed as in (9) by $y(n-1) = d(n-1) - e(n-1)$. In order to initialize conveniently $f_{Y,0}(n) = y(n-1)$, we modify the Y-algorithm as follows :

- Computation of steps 1-Y and 2-Y (for $p = 1 : N_Y$) .
- Then, we are able to compute

$$\begin{aligned} e(n-1) &= \gamma_{Y,N_Y}(n-1)\,e_{Y,N_Y}(n-1) \\ f_{Y,0}(n) &= y(n-1) = d(n-1) - e(n-1) \end{aligned}$$

- Computation of steps 3-Y and 4-Y (for $p = 1 : N_Y$)

Finally, we complete the algorithm by the transformation of equation (10).

## 4. SIMULATION

The algorithms are simulated using the same system model as [4]. This model is often used to simulate acoustic transfer

[1] the letter X here indicate the algorithm X (Z≡X) to which the step belongs.

path in teleconference systems. It is defined by the following feedback polynomial

$$B\left(z^{-1}\right) = 1 - \alpha_i z^{-10}$$

and an arbitrary vector of feedforward coefficients

$$\begin{aligned} a_1 \quad &= \quad (1, 1.52, -0.92, -0.87, 0.55, 2.37, \\ &\quad\quad 1.44, -1.41, -0.28, -0.73, -0.48) \end{aligned}$$

We start the simulation for $\alpha_1 = 0.8$ , then we change abruptly the system at sample $t = 2500$ to the new value $\alpha_2 = 0.95$ and the new feedforward coefficients vector

$$\begin{aligned} a_2 \quad &= \quad (1, -0.66, 0.54, 0.53, -0.68, 0.38, \\ &\quad\quad 1.74, -0.20, 1.13, -0.54, -1.05) \end{aligned}$$

These two systems satisfy the strictly positive real (SPR) condition.

In figure-1 , we consider a noisy identification scheme with an SNR=15dB and a forgetting factor $\lambda = 0.97$. The mean square error MSE is averaged over 100 runs montecarlo simulations. This figure shows a good convergence properties of both the IIR-BQR and the fast Lattice algorithm, which are comparable to the performance of the IIR-QR [4]. However, we have never noticed during simulations any indication of numerical stability problems.

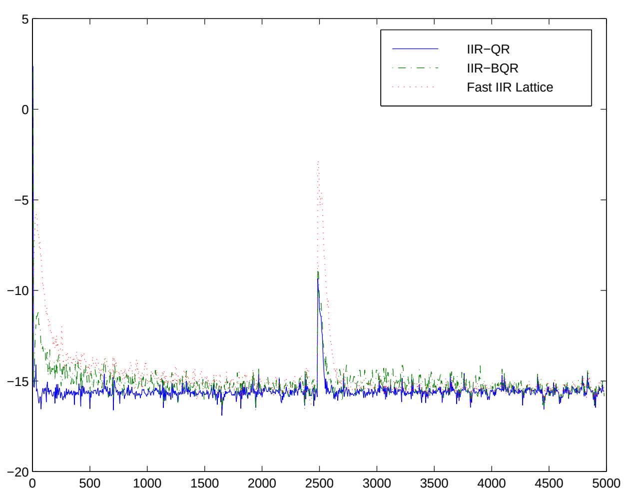


Figure 1: MSE vs. time for IIR-QR, IIR-BQR and fast Lattice algorithm

## 5. CONCLUSION

We presented in this paper a new QR based adaptive IIR filtering algorithm (IIR-BQR). Besides, its reduced complexity (about the half) compared to the IIR-QR, this algorithm is also highly parallelizable. Then, we derived a fast O(N) version of this algorithm which use a lattice parametrization, simulation showed that this fast algorithm is numerically stable.

## 6. REFERENCES


[1] M. Bhouri, M. Bonnet, and M. Mboup. A new QRD-based block adaptive algorithm. In *Proc. ICASSP*, pages 1497–1500, May 1998.

[2] M. Bhouri, M. Mboup, and M. Bonnet. A fast multichannel adaptive algorithm. In *Proc. EUSIPCO*, September 1998.

[3] M. Harteneck, R.W. Stewart, J. G. McWhirter, and I. K. Proudler. Algorithmic engineering applied to the QR-RLS adaptive algorithm. In *Proc. IMA Conf. Math Signal Processing*, December 1996.

[4] M. Harteneck, R.W. Stewart, J. G. McWhirter, and I. K. Proudler. An effective approach to adaptive IIR filtering. In *Proc. ICASSP*, pages 1634–1637, May 1996.

[5] S. Haykin. *Adaptive Filter Theory*. Prentice-Hall, Englewood Cliffs, NJ, second edition, 1991.

[6] L. Ljung and T. Söderström. *Theory and Practice of Recursive Identification*. MIT Press, Cambridge, Massachusetts, 1983.

[7] J.G. McWhirter. Recursive least squares minimization using a systolic array. In *Proc. SPIE Int. Soc. Opt. Eng.*, volume 431, pages 105–112, 1983.

[8] J.J. Shynk. Adaptive IIR filtering. *IEEE ASSP Mag.*, pages 4–21, November 1989.

[9] B. Yang and J.F. Böhme. Rotation-based RLS algorithms: Unified derivations, numerical properties and parallel implementations. *IEEE Trans. SP*, 40:1151–1167, 1992.