A New Fast QR Algorithm Based on *a Priori* Errors

José Antonio Apolinário Jr., *Student Member, IEEE,* and Paulo S. R. Diniz, *Senior Member, IEEE*

Abstract—This letter presents a new fast QR algorithm based on Givens rotations using *a priori* errors. The principles behind the triangularization of the weighted input data matrix via QR decomposition and the type of errors used in the updating process are exploited in order to investigate the relationships among different fast algorithms of the QR family. These algorithms are classified according to a general framework and a detailed description of the new algorithm is presented.

Index Terms—Adaptive filters, fast QR decomposition, recursive least squares, RLS algorithms.

I. INTRODUCTION

FAST recursive least squares (RLS) algorithms based on QR decomposition (using Givens rotations) are among those adaptive filtering algorithms with desired characteristics such as numerical robustness and possibility of efficient implementation.

From the conventional $(O[N^2])$ QR decomposition method [1], [2], a number of fast algorithms $(O[N])$ were derived [3]–[6]. These algorithms can be classified in terms of the type of triangularization applied to the input data matrix (upper or lower triangular) and type of errors (*a posteriori* or *a priori*) involved in the updating process. As will be clear later, an upper triangularization (in the notation of this work) involves the updating of forward prediction errors, while a lower triangularization involves the updating of backward predictions errors. The classification is summarized in Table I. This table also indicates how these algorithms will be designated hereafter.

The proposed algorithm, referred as FQR_PRI_F, is a fast QR that updates *a priori* forward prediction errors. The FQR_PRI_B algorithm was independently developed in [5] and [6] using different approaches. The approach that will be used here derives from concepts used in the inverse QR algorithm [5], [7] (where the inverse Cholesky factor is updated).

II. BASIC CONCEPTS OF QR DECOMPOSITION ALGORITHMS

This section reviews the basic concepts of the conventional and inverse QR algorithms in order to establish the notation of this letter. The RLS algorithms minimize the following cost

TABLE I
CLASSIFICATION OF THE FAST QR ALGORITHMS

Error	Prediction	
Type	Forward	Backward
A Posteriori	FQR_POS_F [3]	FQR_POS_B [4]
A Priori	FQR_PRI_F [new]	FQR_PRI_B [5], [6]

function:

$$\xi(k) = \sum_{i=0}^{k} \lambda^{k-i} e^2(i) = \boldsymbol{e}^T(k)\boldsymbol{e}(k) = \|\boldsymbol{e}(k)\|^2 \qquad (1)$$

where each component of the vector $\boldsymbol{e}(k)$ is the *a posteriori* error at instant i weighted by $\lambda^{(k-i)/2}$ (λ is the forgetting factor). The vector $\boldsymbol{e}(k)$ is given by

$$\boldsymbol{e}(k) = \boldsymbol{d}(k) - \boldsymbol{X}^{(N+1)}(k)\boldsymbol{w}(k). \qquad (2)$$

In (2), $\boldsymbol{d}(k)$ is the weighted desired signal vector, $\boldsymbol{X}^{(N+1)}(k)$ is the weighted input data matrix, N is the order (the number of coefficients is $N + 1$), and $\boldsymbol{w}(k)$ is the coefficient vector. The premultiplication of the above equation by the orthonormal matrix $\boldsymbol{Q}^{(N+1)}(k)$ triangularizes $\boldsymbol{X}^{(N+1)}(k)$ without affecting the cost function.

$$\boldsymbol{Q}^{(N+1)}(k)\boldsymbol{e}(k) = \begin{bmatrix} \boldsymbol{e}_{q_1}(k) \\ \boldsymbol{e}_{q_2}(k) \end{bmatrix}$$
$$= \begin{bmatrix} \boldsymbol{d}_{q_1}(k) \\ \boldsymbol{d}_{q_2}(k) \end{bmatrix} - \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{U}^{(N+1)}(k) \end{bmatrix} \boldsymbol{w}(k). \qquad (3)$$

The weighted-square error in (1) is minimized by choosing $\boldsymbol{w}(k)$ such that the term $\boldsymbol{d}_{q2}(k) - \boldsymbol{U}^{(N+1)}(k)\boldsymbol{w}(k)$ is zero. Equation (3) can be written in a recursive form, as follows, while avoiding ever increasing order for the vectors and matrices involved [1]:

$$\begin{bmatrix} e_{q_1}(k) \\ \boldsymbol{d}_{q_2}(k) \end{bmatrix} = \boldsymbol{Q}_\theta^{(N+1)}(k) \begin{bmatrix} d(k) \\ \lambda^{1/2}\boldsymbol{d}_{q_2}(k-1) \end{bmatrix} \qquad (4)$$

where $\boldsymbol{Q}_\theta^{(N+1)}(k)$ is a sequence of Givens rotations that annihilates the elements of the input vector $\boldsymbol{x}^{(N+1)}(k) = [x(k)x(k-1)\cdots x(k-N)]^T$ in the equation

$$\begin{bmatrix} \boldsymbol{0}^T \\ \boldsymbol{U}^{(N+1)}(k) \end{bmatrix} = \boldsymbol{Q}_\theta^{(N+1)}(k) \begin{bmatrix} \boldsymbol{x}^{(N+1)^T}(k) \\ \lambda^{1/2}\boldsymbol{U}^{(N+1)}(k-1) \end{bmatrix}. \qquad (5)$$

The following relation also used in the conventional QR algorithm is obtained by postmultiplying $\boldsymbol{e}_q^T(k)\boldsymbol{Q}^{(N+1)}(k)$ by the pinning vector $[1 \quad 0 \quad \cdots \quad 0]^T$.

$$e(k) = e_{q_1}(k) \prod_{i=0}^{N} \cos\theta_i(k) = e_{q_1}(k)\gamma(k) \qquad (6)$$

where $\gamma(k)$ is the first element of the first row of $\boldsymbol{Q}_\theta^{(N+1)}(k)$.

III. THE NEW FAST QR-RLS ALGORITHM

The key difference in the development of the fast QR algorithms is the way that the matrix $U(k)$ is triangularized, as follows:

$$U^{(N+1)}(k) = \begin{bmatrix} & & & & 0 \\ & & & \cdot & \cdot \\ & & \cdot & & \cdot \\ & \cdot & & & \cdot \\ \cdot & & & & \cdot \\ 0 & \cdot & \cdot & \cdot & 0 \end{bmatrix} N+1 \quad \text{or}$$

$$N+1$$

$$U^{(N+1)}(k) = \begin{bmatrix} 0 & \cdot & \cdot & \cdot & \cdot & 0 \\ \cdot & & & & \cdot \\ \cdot & & & \cdot & \\ \cdot & & \cdot & & \\ \cdot & \cdot & & & \\ \cdot & \cdot & & & \\ 0 & & & & \end{bmatrix} N+1. \quad (7)$$

$$N+1$$

In both types of triangularization, the matrix $Q_\theta^{(N+1)}(k)$ can be partitioned as

$$Q_\theta^{(N+1)}(k) = \begin{bmatrix} \gamma(k) & -\gamma(k)a^{(N+1)^T}(k) \\ f^{(N+1)}(k) & E(k) \end{bmatrix} \quad (8)$$

where, using (8) in (5) and recalling that $Q_\theta^{(N+1)}(k)$ is orthonormal, it is possible to prove that $f^{(N+1)}(k) = [U^{(N+1)}(k)]^{-T}x^{(N+1)}(k)$ is the normalized *a posteriori* forward (upper triangularization)/backward (lower triangularization) prediction error vector [6], $a^{(N+1)}(k) = U^{-T}(k-1)x(k)/\sqrt{\lambda}$ is the normalized *a priori* forward (upper triangularization)/backward (lower triangularization) prediction error vector [6], and $E(k) = \lambda^{1/2}[U(k)]^{-T}[U(k-1)]^T$.

In the derivation of fast QR algorithms, we start by applying the QR decomposition to the forward and backward prediction problems whose prediction errors are, respectively, defined as

$$e_f(k) = \begin{bmatrix} d_f(k) & \begin{matrix} X(k-1) \\ 0^T \end{matrix} \end{bmatrix} \begin{bmatrix} 1 \\ -w_f(k) \end{bmatrix} \quad (9)$$

$$e_b(k) = [X(k) \quad d_b(k)] \begin{bmatrix} -w_b(k) \\ 1 \end{bmatrix}. \quad (10)$$

It is fundamental to note that the partitioned matrices in the last two equations correspond to $X^{(N+2)}(k+1)$ (weighted input data matrix of order $N+1$). Our aim is to triangularize $X^{(N+2)}(k)$ such that $Q^{(N+2)}(k)X^{(N+2)}(k) = \begin{bmatrix} 0 \\ U^{(N+2)}(k) \end{bmatrix}$. The upper triangularization of $U^{(N+2)}(k)$ is implemented by premultiplying $e_f(k)$ by the product $Q_f(k)\begin{bmatrix} Q^{(N+1)}(k-1) & 0 \\ 0^T & 1 \end{bmatrix}$, where $Q_f(k)$ is a set of Givens rotations generating $\|e_f(k)\|$ by eliminating the first $k-N$

elements of the rotated desired vector of the forward predictor. The result is

$$U^{(N+2)}(k) = \begin{bmatrix} d_{fq_2}(k) & U^{(N+1)}(k-1) \\ \|e_f(k)\| & 0^T \end{bmatrix} \quad (11)$$

where $\|e_f(k)\|^2 = \lambda\|e_f(k-1)\|^2 + e_{fq_1}^2(k)$.

By working with nonincreasing dimensions, it is easy to show that [1]

$$Q_\theta^{(N+2)}(k) = Q_{\theta f}(k)\begin{bmatrix} Q_\theta^{(N+1)}(k) & 0 \\ 0^T & 1 \end{bmatrix} \quad (12)$$

$$\begin{bmatrix} e_{fq_1}(k) \\ d_{fq_2}(k) \end{bmatrix} = Q_\theta^{(N+1)}(k-1)\begin{bmatrix} x(k) \\ \lambda^{1/2}d_{fq_2}(k-1) \end{bmatrix}. \quad (13)$$

In the backward prediction problem, the triangularization is achieved using three matrices, $Q^{(N+2)}(k) = Q_b'(k)Q_b(k)Q^{(N+1)}(k)$, where $Q_b(k)$ and $Q_b'(k)$ are two sets of Givens rotations applied to generate, respectively, $\|e_b(k)\|$ and $\|e_b^{(0)}(k)\|$. As a result, we have

$$U^{(N+2)}(k) = Q_{\theta b}'(k)\begin{bmatrix} 0^T & \|e_b(k)\| \\ U^{(N+1)}(k) & d_{bq_2}(k) \end{bmatrix}$$

$$= \begin{bmatrix} z^T & \|e_b^{(0)}(k)\| \\ R & 0 \end{bmatrix} \quad (14)$$

where $Q_{\theta b}'(k)$ is a submatrix of $Q_b'(k)$ and $\|e_b^{(0)}(k)\|$ is the norm of the backward error of a zero-coefficient predictor.

If we take the inverse of (11) and (14), the following relations result:

$$[U^{(N+2)}(k)]^{-1}$$

$$= \begin{bmatrix} 0^T & \dfrac{1}{\|e_f(k)\|} \\ U^{(N+1)^{-1}}(k-1) & \dfrac{-U^{(N+1)^{-1}}(k-1)d_{fq_2}(k)}{\|e_f(k)\|} \end{bmatrix}$$

$$= \begin{bmatrix} 0 & R^{-1} \\ \dfrac{1}{\|e_b^{(0)}(k)\|} & \dfrac{-z^T R^{-1}}{\|e_b^{(0)}(k)\|} \end{bmatrix}. \quad (15)$$

The expressions of $[U^{(N+2)}(k)]^{-1}$ given in (15) can be used to obtain the vectors $f^{(N+2)}(k+1)$ and $a^{(N+2)}(k+1)$. The choice of one of these vectors will determine the algorithm: updating $f^{(N+1)}(k)$ (*a posteriori* errors) will lead to the FQR_POS_F algorithm [3] and updating $a^{(N+1)}(k)$ (*a priori* errors) will lead to the new FQR_PRI_F algorithm.

Expressing $a^{(N+2)}(k+1) = [U^{(N+2)}(k)]^{-T}x^{(N+2)}(k+1)/\sqrt{\lambda}$ in terms of the matrices in (15) and premultiplying the one that comes from the backward prediction problem by $Q_{\theta b}'(k)Q_{\theta b}'^T(k)$ yields

$$\begin{bmatrix} \dfrac{e_b'(k+1)}{\sqrt{\lambda}\|e_b(k)\|} \\ a^{(N+1)}(k+1) \end{bmatrix} = Q_{\theta b}'^T(k)\begin{bmatrix} a^{(N+1)}(k) \\ \dfrac{e_f'(k)}{\sqrt{\lambda}\|e_f(k-1)\|} \end{bmatrix}. \quad (16)$$

Once we have $a^{(N+1)}(k+1)$, the angles of $Q_\theta^{(N+1)}(k+1)$ are found through the following relation obtained by postmultiplying $[Q_\theta(k+1)]^T$ [see (8)] by the pinning vector.

$$\begin{bmatrix} 1/\gamma(k+1) \\ 0 \end{bmatrix} = Q_\theta^{(N+1)}(k+1)\begin{bmatrix} 1 \\ -a^{(N+1)}(k+1) \end{bmatrix}. \quad (17)$$

TABLE II
FQR_PRI_F ALGORITHM

FQR_PRI_F

for each k

$\{$ Obtaining $e'_f(k+1)$:

$$\begin{bmatrix} e_{fq_1}(k+1) \\ d_{fq_2}(k+1) \end{bmatrix} = Q_\theta^{(N+1)}(k) \begin{bmatrix} x(k+1) \\ \lambda^{1/2} d_{fq_2}(k) \end{bmatrix}$$

$e'_f(k+1) = e_{fq_1}(k+1)/\gamma(k)$

Obtaining $a^{(N+1)}(k+1)$:

$$\begin{bmatrix} \frac{e'_b(k+1)}{\sqrt{\lambda}\|e_b(k)\|} \\ a^{(N+1)}(k+1) \end{bmatrix} = Q'^T_{\theta b}(k) \begin{bmatrix} a^{(N+1)}(k) \\ \frac{e'_f(k+1)}{\sqrt{\lambda}\|e_f(k)\|} \end{bmatrix}$$

Obtaining $Q_{\theta f}(k+1)$:

$\| e_f(k+1) \| = \sqrt{e_{fq_1}^2(k+1) + \lambda \| e_f(k) \|^2}$

$cos\theta_f(k+1) = \lambda^{1/2} \| e_f(k) \| / \| e_f(k+1) \|$

$sin\theta_f(k+1) = e_{fq_1}(k+1) / \| e_f(k+1) \|$

Obtaining $c(k+1)$:

$$Q_\theta^{(N+2)}(k+1) = Q_{\theta f}(k+1) \begin{bmatrix} Q_\theta^{(N+1)}(k) & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix}$$

$\hat{Q}_\theta^{(N+2)}(k+1) = $ last $(N+2) \times (N+2)$
  elements of $Q_\theta^{(N+2)}(k+1)$

$$c(k+1) = \hat{Q}_\theta^{(N+2)}(k+1)Q'_{\theta b}(k) \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Obtaining $Q'_{\theta b}(k+1)$:

$$\begin{bmatrix} b \\ 0 \end{bmatrix} = Q'^T_{\theta b}(k+1)c(k+1)$$

Obtaining $Q_\theta^{(N+1)}(k+1)$:

$$\begin{bmatrix} 1/\gamma(k+1) \\ 0 \end{bmatrix} = Q_\theta^{(N+1)}(k+1) \begin{bmatrix} 1 \\ -a^{(N+1)}(k+1) \end{bmatrix}$$

Joint Process Estimation:

$$\begin{bmatrix} e_{q_1}(k+1) \\ d_{q_2}(k+1) \end{bmatrix} = Q_\theta^{(N+1)}(k+1) \begin{bmatrix} d(k+1) \\ \lambda^{1/2} d_{q_2}(k) \end{bmatrix}$$

$e(k+1) = e_{q_1}(k+1)\gamma(k+1)$

$\}$

The quantities required to compute the angles of $Q'_{\theta_b}(k+1)$ are not available at instant k, and a special strategy is required. The updated $Q'_{\theta_b}(k+1)$ is obtained [1] with the use of the vector $c(k+1)$ defined as

$$c(k+1) = \hat{Q}_\theta^{(N+2)}(k+1)Q'_{\theta_b}(k)\begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$= Q'_{\theta_b}(k+1)\begin{bmatrix} b \\ 0 \end{bmatrix}. \tag{18}$$

The submatrix $\hat{Q}_\theta^{(N+2)}(k+1)$ consisting of the last $(N+2) \times (N+2)$ elements of $Q_\theta^{(N+2)}(k+1)$ was already obtained in the forward prediction [see (12)]. Finally, the joint process estimation is calculated with (4) and (6).

With the equations presented in this section, we are able to describe the new fast QR algorithm based on *a priori* forward prediction errors. Table II describes the new algorithm.

In the case of the FQR_POS_F algorithm, the vector $f^{(N+2)}(k+1) = [U^{(N+2)}(k+1)]^{-T}x^{(N+2)}(k+1)$ is expressed in terms of the matrices in (15). The same mentioned strategy
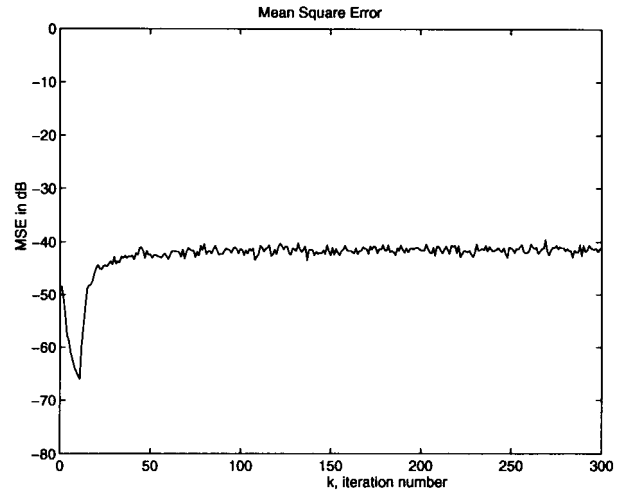

Fig. 1. Learning curve of the new algorithm.

is used to obtain $Q'_{\theta_b}(k+1)$ and the angles of $Q_\theta^{(N+1)}(k+1)$ can be calculated if we postmultiply $Q_\theta^{(N+1)}(k+1)$ by the pinning vector.

It is worth mentioning that, following similar steps as in the upper triangularization, it is possible to obtain the lower triangular matrix $U^{(N+2)}(k)$ from the forward and backward prediction problems and, after obtaining the inverse $[U^{(N+2)}(k)]^{-1}$, we can update $a^{(N+1)}(k)$ (FQR_PRI_B) or $f^{(N+1)}(k)$ (FQR_POS_B).

IV. SIMULATION RESULTS

In order to test the new algorithm, simulations were carried out in a system identification problem. The system order was $N = 10$, the input signal was a colored noise with a conditioning number around 55, a forgetting factor $\lambda = 0.98$, the SNR = 40 dB, and the initialization factor $\mu = 0.1$.

The learning curve (MSE in dB) is depicted in Fig. 1, corresponding to an average of 100 realizations.

Although finite precision analysis is under investigation, the new algorithm showed no sign of instability when simulated in fixed-point arithmetic—all variables represented with 16 b and 12 b in the fractional part.

REFERENCES

[1] P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*.   Boston, MA: Kluwer, 1997.
[2] S. Haykin, *Adaptive Filter Theory*, 2nd ed.   Englewood Cliffs, NJ: Prentice-Hall, 1991.
[3] J. M. Cioffi, "The fast adaptive ROTOR's RLS algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, pp. 631–653, Apr. 1990.
[4] P. A. Regalia and M. G. Bellanger, "On the duality between fast QR methods and lattice methods in least squares adaptive filtering," *IEEE Trans. Signal Processing*, vol. 39, pp. 879–891, Apr. 1991.
[5] A. A. Rontogiannis and S. Theodoridis, "New fast inverse QR least squares adaptive algorithms," in *Proc. ICASSP'95*, Detroit, MI, 1995.
[6] M. D. Miranda and M. Gerken, "A hybrid QR-lattice least squares algorithm using *a priori* errors," in *Proc. MWSCAS'95*, Rio de Janeiro, Brazil, 1995.
[7] S. T. Alexander and A. L. Ghirnikar, "A method for recursive least squares adaptive filtering based upon an inverse QR decomposition," *IEEE Trans. Signal Processing*, vol. 41, pp. 20–30, Jan. 1993.