Infinite Precision Analysis of the Fast QR Algorithm Based on *a Posteriori* Backward Prediction Errors

César A. Medina S.[†], José A. Apolinário Jr.[†], and Paulo S. R. Diniz[‡]
[†]IME—DE/3 and [‡]COPPE/UFRJ—PEE, Rio de Janeiro—RJ, Brazil

Abstract— **The conventional QR Decomposition (QRD) method requires order N^2—O[N^2]—multiplications per output samples. However a number of fast QRD algorithms have been proposed with O[N] of complexity. Particularly the Fast QRD algorithm based on *a posteriori* backward prediction errors is well known for its good numerical behavior and low complexity. This algorithm has two distinct versions and, in order to decide which one to choose for a given implementation, its infinite precision analysis of the mean square values of the internal variables would be required. In addition to this implementation issue, the finite-precision analysis requires the estimates of these mean square values. In this work, we first present an overview of the Fast QRD algorithms based on *a posteriori* backward prediction errors, followed by an infinite precision analysis of the steady state mean square values of the internal variables. Finally, the validity of the analytical results are verified by computer simulations carried out in a system identification setup. In the appendices, the detailed description of each implementation is listed.**

I. Introduction

SINCE the first QR Decomposition based Fast RLS algorithm introduced by John Cioffi in 1990 [1], many other Fast QRD-based RLS algorithms were developed [2], [3], [4], [5], [6]. It is known that the fast QR algorithms based on backward prediction errors updating are minimal in system theory sense and backward stable [2], [7]. This work is focused on the study of the Fast QRD algorithm based on *a posteriori* backward prediction errors, which is well known for its good numerical behavior and low computational complexity. This algorithm has two know implementations and the choice of the best one is not an easy task, specially in fixed-point arithmetic implementations using a reasonably large number of bits of wordlength. The proper choice requires an infinite precision analysis of the mean square values of the internal variables in order to address the implementation complexity of the fixed-point arithmetic implementation. In addition, the finite-precision analysis requires the estimates of these mean square values.

César A. Medina and José A. Apolinário Jr. are with the Department of Electrical Engineering, Instituto Militar de Engenharia, Praça General Tibúrcio, 80, Rio de Janeiro—RJ, Brazil, 22.290-270, Phone: +55 21 2546 7030 Fax: +55 21 25467039. E-mails: csmedina@epq.ime.eb.br and apolin@ieee.org. Paulo S. R. Diniz is with the Program of Electrical Engineering, COPPE/Universidade Federal do Rio de Janeiro, P. O. Box 68504, Rio de Janeiro—RJ, Brazil, 21.945-970. E-mail: diniz@lps.ufrj.br.
César A. Medina S. thanks CAPES for partially supporting this work.

It can be seen on [5] that fast QRD-RLS algorithms can be classified in terms of the type of triangularization applied to the input data matrix (upper or lower triangular) and type of error vector (*a posteriori* or *a priori*) involved in the updating process. It can be seen from the Gram-Schmidt orthogonalization procedure that an upper triangularization (notation being the same as in [5]) involves the updating of forward prediction errors while a lower triangularization involves the updating of backward prediction errors. Table I presents this classification as well as points out how these algorithms will be designated hereafter.

TABLE I
CLASSIFICATION OF THE FAST QR ALGORITHMS.

Error Type	Prediction: Forward	Prediction: Backward
A Posteriori	FQR_POS_F [1]	FQR_POS_B [2], [6]
A Priori	FQR_PRI_F [5]	FQR_PRI_B [3], [4]

This paper is organized as follows. In Section 2, we present an overview of these fast algorithms. Then, in Section 3, the infinite precision analysis concerning the steady state mean square values of each internal variable is presented. In Section 4, the validation of the analytical results obtained is carried out through computer simulations. Finally, some conclusions are summarized.

II. The FQR Algorithms Based on Backward Prediction Errors

The RLS algorithms minimize the following cost function

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

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

$$\boldsymbol{e}(k) = \boldsymbol{d}(k) - \boldsymbol{X}(k)\boldsymbol{w}(k) \qquad (2)$$

In (2), $\boldsymbol{d}(k)$ is the weighted desired signal vector, $\boldsymbol{X}(k)$ is the weighted input data matrix of order N (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 $\boldsymbol{Q}(k)$ triangularizes $\boldsymbol{X}(k)$ without affecting the cost function.

$$\begin{aligned} \boldsymbol{Q}(k)\boldsymbol{e}(k) &= \mathbf{e_q}(\mathbf{k}) = \begin{bmatrix} \boldsymbol{e}_{q_1}(k) \\ \boldsymbol{e}_{q_2}(k) \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{d}_{q_1}(k) \\ \boldsymbol{d}_{q_2}(k) \end{bmatrix} - \begin{bmatrix} \mathbf{0} \\ \boldsymbol{U}(k) \end{bmatrix} \boldsymbol{w}(k) \end{aligned} \qquad (3)$$

The weighted-square error in (1) is minimized by choosing $\boldsymbol{w}(k)$ such that the term $\boldsymbol{d}_{q2}(k) - \boldsymbol{U}(k)\boldsymbol{w}(k)$ is zero. Equation (3) can be written in a recursive form while avoiding ever increasing order for the vectors and matrices involved [8]:

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

where $\boldsymbol{Q}_\theta(k) = \prod_{i=N}^{0} \boldsymbol{Q}_{\theta_i}(k)$ is a sequence of Givens rotations that annihilates the elements of the input vector $\boldsymbol{x}(k) = [x(k)\, x(k-1)\, \cdots\, x(k-N)]^T$ in the equation

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

and,

$$\boldsymbol{Q}_{\theta_i}(k) = \begin{bmatrix} cos\theta_i(k) & \mathbf{0}^T & -sin\theta_i(k) & \mathbf{0}^T \\ \mathbf{0} & \boldsymbol{I}_{N-i} & \mathbf{0} & \mathbf{0}\cdots\mathbf{0} \\ sin\theta_i(k) & \mathbf{0}^T & cos\theta_i(k) & \mathbf{0}^T \\ \mathbf{0} & \mathbf{0}\cdots\mathbf{0} & \mathbf{0} & \boldsymbol{I}_i \end{bmatrix} \qquad (6)$$

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

$$e(k) = e_{q_1}(k) \prod_{i=0}^{N} \cos\theta_i(k) = e_{q_1}(k)\gamma(k) \qquad (7)$$

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

Matrix $\boldsymbol{Q}_\theta(k)$ in (4) can be partitioned as

$$\boldsymbol{Q}_\theta(k) = \begin{bmatrix} \gamma(k) & -\gamma(k)\boldsymbol{a}^T(k) \\ \boldsymbol{f}(k) & \boldsymbol{E}(k) \end{bmatrix} \qquad (8)$$

where, using (8) in (5) and recalling that $\boldsymbol{Q}_\theta(k)$ is orthonormal, it is possible to prove that, for the case of lower triangularization of $\mathbf{U}(\mathbf{k})$ (backward prediction errors update), $\boldsymbol{f}(k) = [\boldsymbol{U}(k)]^{-T}\boldsymbol{x}(k)$ is the normalized *a posteriori* backward prediction error vector [3], $\boldsymbol{a}(k) = \boldsymbol{U}^{-T}(k-1)\boldsymbol{X}(k)/\sqrt{\lambda}$ is the normalized *a priori* backward prediction error vector [3], and $\boldsymbol{E}(k) = \lambda^{1/2}[\boldsymbol{U}(k)]^{-T}[\boldsymbol{U}(k-1)]^T$.

The update of the *a posteriori* backward prediction error vector, $\mathbf{f}(\mathbf{k})$, leads to the so-called FQR_POS_B algorithm. The update equation of this vector is given by

$$\begin{bmatrix} \frac{e_b(k+1)}{\|\boldsymbol{e}_b(k+1)\|} \\ \boldsymbol{f}(k+1) \end{bmatrix} = \boldsymbol{Q}'_{\theta_f}(k+1) \begin{bmatrix} \boldsymbol{f}(k) \\ \frac{e_f(k+1)}{\|\boldsymbol{e}_f(k+1)\|} \end{bmatrix} \qquad (9)$$

where $\boldsymbol{Q}'_{\theta f}(k)$ is a sequence of Givens rotations that generates $\parallel \boldsymbol{e}_f(k) \parallel$ and it can be obtained from the following equation

$$\begin{bmatrix} \mathbf{0} \\ \parallel \boldsymbol{e}_f^{(0)}(k+1) \parallel \end{bmatrix} = \boldsymbol{Q}'_{\theta f}(k+1) \begin{bmatrix} \boldsymbol{d}_{fq_2}(k+1) \\ \parallel \boldsymbol{e}_f(k+1) \parallel \end{bmatrix} \qquad (10)$$

For this algorithm and from the derivation of (9), it can be observed that the last element of $\boldsymbol{f}(k+1)$, given by $\frac{x(k+1)}{\|\boldsymbol{e}_f^{(0)}(k+1)\|}$, was precalculated in the previous step. This fact leads to two slightly different implementations of the same algorithm. The first one, is based on this prior knowledge of the last element of $\boldsymbol{f}(k+1)$ and the second one is based on the straightforward computation of $\boldsymbol{f}(k+1)$, therefore, requiring the calculation of $\frac{e_f(k+1)}{\|\boldsymbol{e}_f(k+1)\|}$.

The first version of this algorithm was introduced in [6] and its detailed description is presented in Appendix A. The second version of this algorithm was introduced in [2] and its detailed description is given is Appendix B.

III. Mean Square Values of the Internal Variables of the FQR_POS_B Algorithm

The basic matrix equations of the fast QR algorithm based on *a posteriori* backward prediction errors are listed in Table II. In this section, we derive the mean square values for variables encountered in both implementations. For the infinite precision results of the FQR_POS_B algorithm, all variables have the same notation used in its detailed description (Appendices).

TABLE II
THE FQR_POS_B EQUATIONS.

FQR_POS_B
for each k
{ 1. Obtaining $\boldsymbol{d}_{fq_2}(k+1)$:
$\begin{bmatrix} e_{fq_1}(k+1) \\ \boldsymbol{d}_{fq_2}(k+1) \end{bmatrix} = \boldsymbol{Q}_\theta(k) \begin{bmatrix} x(k+1) \\ \lambda^{1/2}\boldsymbol{d}_{fq_2}(k) \end{bmatrix}$
2. Obtaining $\parallel \boldsymbol{e}_f(k+1) \parallel$:
$\parallel \boldsymbol{e}_f(k+1) \parallel = \sqrt{e^2_{fq_1}(k+1) + \lambda \parallel \boldsymbol{e}_f(k) \parallel^2}$
3. Obtaining $\boldsymbol{Q}'_{\theta f}(k+1)$:
$\begin{bmatrix} \boldsymbol{0} \\ \parallel \boldsymbol{e}_f^{(0)}(k+1) \parallel \end{bmatrix} = \boldsymbol{Q}'_{\theta f}(k+1) \begin{bmatrix} \boldsymbol{d}_{fq_2}(k+1) \\ \parallel \boldsymbol{e}_f(k+1) \parallel \end{bmatrix}$
4. Obtaining $\boldsymbol{f}(k+1)$:
$\begin{bmatrix} \frac{e_b(k+1)}{\parallel \boldsymbol{e}_b(k+1) \parallel} \\ \boldsymbol{f}(k+1) \end{bmatrix} = \boldsymbol{Q}'_{\theta f}(k+1) \begin{bmatrix} \boldsymbol{f}(k) \\ \frac{e_f(k+1)}{\parallel \boldsymbol{e}_f(k+1) \parallel} \end{bmatrix}$
5. Obtaining $\boldsymbol{Q}_\theta(k+1)$:
$\begin{bmatrix} 1 \\ \boldsymbol{0} \end{bmatrix} = \boldsymbol{Q}_\theta^T(k+1) \begin{bmatrix} \gamma(k+1) \\ \boldsymbol{f}(k+1) \end{bmatrix}$
6. Joint Process Estimation:
$\begin{bmatrix} e_{q_1}(k+1) \\ \boldsymbol{d}_{q_2}(k+1) \end{bmatrix} = \boldsymbol{Q}_\theta(k+1) \begin{bmatrix} d(k+1) \\ \lambda^{1/2}\boldsymbol{d}_{q_2}(k) \end{bmatrix}$
7. Updating the output error:
$e(k+1) = e_{q_1}(k+1)\gamma(k+1)$
}

A. *Mean Square Values of $cos\theta_i(k)$ and $sin\theta_i(k)$*

The following results can be found in [9].

$$E[cos^2\theta_i(k)] \approx \lambda \tag{11}$$

$$E[sin^2\theta_i(k)] \approx 1 - \lambda \tag{12}$$

B. *Mean Square Value of $e^{(i)}_{fq_1}(k)$*

The following result was first derived in [10].

$$E\left\{[e^{(i)}_{fq_1}(k)]^2\right\} \approx \sigma_x^2 \left(\frac{2\lambda}{1+\lambda}\right)^i \tag{13}$$

C. *Mean Square Value of $d_{fq2_i}(k)$*

The following result was obtained from [10].

$$E\left\{[d_{fq2_i}(k)]^2\right\} \approx \frac{\sigma_x^2}{1+\lambda}\left(\frac{2\lambda}{1+\lambda}\right)^{N+1-i} \tag{14}$$

D. *Mean Square Value of $\parallel \boldsymbol{e}_f^{(i)}(k) \parallel$*

The following result can be found in [10].

$$E[\parallel \boldsymbol{e}_f^{(i)}(k) \parallel^2] \approx \frac{\sigma_x^2}{1-\lambda}\left(\frac{2\lambda}{1+\lambda}\right)^i \tag{15}$$

E. *Mean Square Values of $cos\theta'_{f_i}(k)$ and $sin\theta'_{f_i}(k)$*

The following results were also derived in [10].

$$E[cos^2\theta'_{f_i}(k)] \approx \frac{2\lambda}{1+\lambda} \tag{16}$$

$$E[sin^2\theta'_{f_i}(k)] \approx \frac{1-\lambda}{1+\lambda} \tag{17}$$

F. *Mean Square Value of $\gamma^{(i)}(k)$*

It is known from the technical literature that $\gamma(k) = \prod_{i=0}^N cos\theta_i(k)$. If we use (11) and (12), and assume independence between $cos\theta_i(k)$ and $cos\theta_j(k)$, $i \neq j$, it is easy to find the next expression

$$E\left\{[\gamma^{(i)}(k)]^2\right\} \approx \lambda^i \tag{18}$$

The same expression was also obtained in [11] using a different approach.

G. *Mean Square Value of $d_{q2_i}(k)$*

The following results was first introduced in the literature in [9].

$$\begin{aligned} E[d^2_{q2_{N+1-i}}(k)] \approx \\ \left[\tfrac{2\lambda}{1+\lambda}\right]^i \quad \left[\tfrac{\sigma_x^2}{1-\lambda} w^2_{0,i} + \tfrac{\sigma_x^2}{1+\lambda}\textstyle\sum_{j=i+1}^N w^2_{0,j}\right] \end{aligned} \tag{19}$$

where $w^2_{0,i} = E[w_i^2(k)]$. Observe that although $w_{0,i}$ is not available, a rough estimate of $\sigma_x^2 w^2_{0,i}$ can be obtained based on the power of the reference signal [9].

H. *Mean Square Value of $e^{(i)}_{q_1}(k)$*

From the joint process estimation part of the FQR_POS_B algorithm we take the expressions of $e^{(i)}_{q_1}(k+1)$ and $d_{q2_{N+2-i}}(k+1)$, and use them to derive the expected value of $[e^{(i)}_{q_1}(k+1)]^2 + [d_{q2_{N+2-i}}(k+1)]^2$. By assuming stationarity, we find the following relation.

$$\begin{aligned} E\left\{[e^{(i)}_{q_1}(k)]^2\right\} &= E\left\{[e^{(i-1)}_{q_1}(k)]^2\right\} \\ &-(1-\lambda)E[d^2_{q2_{N+2-i}}(k)] \end{aligned} \tag{20}$$

where $E\left\{[e^{(0)}_{q_1}(k)]^2\right\} = \sigma_d^2 = \sigma_x^2\sum_{i=0}^N w^2_{o,i} + \sigma_n^2$ is the variance of the reference signal and σ_n^2 is the variance of the measurement noise (it is assumed here that the algorithm is applied in a sufficient-order identification problem, i.e., the unknown FIR system has the same order as the adaptive filter).

Finally, from the last equation of the algorithm, we have $E[e^2(k)] \approx \lambda^{N+1}E[e^2_{q_1}(k)]$. Since from (20) and (19) we have that $E[e^2_{q_1}(k)] = \sigma_n^2$, the following expression results:

$$E[e^2(k)] \approx \lambda^{N+1}\sigma_n^2 \tag{21}$$

I. Mean Square Value of $f_i(k)$

From the implementation of the step "Obtaining $\boldsymbol{Q}_\theta(k+1)$" (see Table II and Appendix A or B) we have

$$f_{N+2-i}(k+1) = \gamma^{(i-1)}(k+1)sin\theta_{i-1}(k+1) \tag{22}$$

By taking the expected value of (22) and using the approximations (12) and (18), we obtain

$$E[f_i^2(k)] \approx \lambda^{N+1-i}(1-\lambda) \tag{23}$$

J. Mean Square Value of aux_i

The implementation of the step "Obtaining $f(k+1)$" can be realized by two ways, as mentioned in the previous section. These implementations are found in Appendices A and B, respectively.

In the first version of this algorithm since we take the expressions for $f_{N+2-i}(k+1)$ and aux_i, and use them to calculate $E[f^2_{N+2-i}(k)]+E[aux_i^2]$, it is straightforward to conclude that $E[aux_i^2]+E[f^2_{N+2-i}(k+1)] = E[aux^2_{i-1}]+E[f^2_{N+i-1}(k)]$. Since $f_{N+1}(k+1) = aux_0$, it is easy to figure out that $E[aux_i^2] = E[f^2_{N+1-i}(k)]$, then:

$$E[aux_i^2] \approx \lambda^i(1-\lambda) \tag{24}$$

For the second version of this algorithm since we use the expressions for $f_{i-1}(k+1)$ and aux_i to calculate $E[f^2_{i-1}(k+1)] + E[aux_i^2]$, it is straightforward to show that $E[aux^2_{i-1}] - E[f^2_{i-1}(k+1)] = E[aux_i^2] - E[f_i^2(k)]$. Since $f_{N+1}(k+1) = aux_{N+1}$, follows that $E[aux_i^2] = E[f_i^2(k)]$, therefore:

$$E[aux_i^2] \approx \lambda^{N+1-i}(1-\lambda) \tag{25}$$

IV. Simulation Results

In this section we consider an example where the input signal is a zero-mean random Gaussian process with variance $\sigma_x^2 = -30$ dB, the measurement noise is Gaussian with variance -70 dB, the desired signal is obtained through a fourth-order filter. In an ensemble of 1000 runs, each with 5000 samples, only the 4000 last output samples were used to calculate the mean square value.

The chosen λ was 0.95. The simulated and theoretical results are shown in Table III where we see that the predicted values are very close to the measured values.

TABLE III

Mean Square Value.

	Simulation	**Theoretical**
	$e^{(i)}_{fq_1}(k)$	$e^{(i)}_{fq_1}(k)$
i=0	$0.1001463\ 10^{-2}$	$0.1000000\ 10^{-2}$
i=1	$0.0977555\ 10^{-2}$	$0.0974359\ 10^{-2}$
i=2	$0.0954073\ 10^{-2}$	$0.0949375\ 10^{-2}$
i=3	$0.0931882\ 10^{-2}$	$0.0925032\ 10^{-2}$
i=4	$0.0909946\ 10^{-2}$	$0.0901314\ 10^{-2}$
i=5	$0.0888874\ 10^{-2}$	$0.0878203\ 10^{-2}$
	$\parallel \boldsymbol{e}^{(i)}_f(k) \parallel$	$\parallel \boldsymbol{e}^{(i)}_f(k) \parallel$
i=0	0.020026731	0.020000000
i=1	0.019549052	0.019487179
i=2	0.019079547	0.018987508
i=3	0.018635667	0.018500649
i=4	0.018196971	0.018026273
i=5	0.017775505	0.017564061
i=0	1.001001001	1.000000000
i=1	0.953151588	0.950000000
i=2	0.907438523	0.902500000
i=3	0.863791227	0.857375000
i=4	0.822138408	0.814506250
i=5	0.782402799	0.773780938
	$d_{fq2_i}(k)$	$d_{fq2_i}(k)$
	$\gamma^{(i)}(k)$	$\gamma^{(i)}(k)$
i=1	$0.421465504\ 10^{-3}$	$0.462212139\ 10^{-3}$
i=2	$0.438696191\ 10^{-3}$	$0.474375616\ 10^{-3}$
i=3	$0.443879956\ 10^{-3}$	$0.486859185\ 10^{-3}$
i=4	$0.469505010\ 10^{-3}$	$0.499671269\ 10^{-3}$
i=5	$0.477679154\ 10^{-3}$	$0.512820513\ 10^{-3}$
	$cos\theta'_{f_i}(k)$	$cos\theta'_{f_i}(k)$
i=0	0.977811931	0.974358974
i=1	0.977604023	0.974358974
i=2	0.978317517	0.974358974
i=3	0.978072023	0.974358974
i=4	0.978399190	0.974358974
	$sin\theta'_i(k)$	$sin\theta'_i(k)$
i=0	0.023189070	0.025641026
i=1	0.023396978	0.025641026
i=2	0.022683484	0.025641026
i=3	0.022928978	0.025641026
i=4	0.022601811	0.025641026

V. Conclusions

In this paper, the two implementations of the fast QR decomposition algorithm based on *a posteriori* backward prediction errors have been analyzed in finite precision environment. These implementations are generally good choices among the fast RLS algorithms due to their low computational complexity and proved stability when implemented with finite precision arithmetic.

Closed-form formulae for the estimation of the mean square values of the internal variables were obtained, and theoretical results were compared with computer simulations, confirming the accuracy of the analysis.

These expressions are keys for a proper implementation of this algorithm using fixed-point arithmetic processors, since the number of bits for each internal variable could be determined by its estimated mean squared value. In addition, they are required in the finite-precision analysis of the FQR_POS_B algorithms.

Mean Square Value (Continuation)

Simulation	**Theoretical**
$d_{q2_i}(k)$	$d_{q2_i}(k)$
i=0 0.000182101	0.000180263
i=1 0.000749849	0.000744770
i=2 0.000213413	0.000214218
i=3 0.015857762	0.015814596
i=4 0.000616829	0.000646154
$e_{q1}^{(i)}(k+1)$	$e_{q1}^{(i)}(k+1)$
i=0 $0.881162016\ 10^{-3}$	$0.88\ 10^{-3}$
i=1 $0.850300906\ 10^{-3}$	$0.847692308\ 10^{-3}$
i=2 $0.057352799\ 10^{-3}$	$0.056962525\ 10^{-3}$
i=3 $0.046681956\ 10^{-3}$	$0.046251623\ 10^{-3}$
i=4 $0.009183574\ 10^{-3}$	$0.009013137\ 10^{-3}$
i=5 $0.000077200\ 10^{-3}$	0
$f_i(k)$	$f_i(k)$
i=0 0.037887517	0.038689047
i=1 0.039735609	0.040725313
i=2 0.041652819	0.042868750
i=3 0.043647296	0.045125000
i=4 0.045713065	0.047500000
i=5 0.047849413	0.050000000
$aux_i FQR\_POS\_BV.2$	$aux_i FQR\_POS\_BV.2$
i=0 0.037887263	0.038689047
i=1 0.039734548	0.040725313
i=2 0.041652272	0.042868750
i=3 0.043648002	0.045125000
i=4 0.045713712	0.047500000
i=5 0.047849413	0.050000000
$aux_i FQR\_POS\_BV.1$	$aux_i FQR\_POS\_BV.1$
i=0 0.047849413	0.047500000
i=1 0.045713712	0.045125000
i=2 0.043648002	0.042868750
i=3 0.041652272	0.040725313
i=4 0.039734548	0.038689047
$e(k)$	$e(k)$
$6.09\ 10^{-8}$	$7.74\ 10^{-8}$
$cos\theta_i(k)$	$cos\theta_i(k)$
0.953151588	0.95
$sin\theta_i(k)$	$sin\theta_i(k)$
0.047849413	0.05

References

[1] J. M. Cioffi, "The Fast Adaptive ROTOR's RLS Algorithm," *IEEE Transactions on Acoustic, Speech, and Signal Processing*, vol. ASSP-38, pp. 631–653, Apr. 1990.

[2] P. A. Regalia and M. G. Bellanger, "On the Duality between Fast QR Methods and Lattice Methods in Least Squares Adaptive Filtering," *IEEE Transactions on Signal Processing*, vol. SP-39, pp. 879–891, Apr. 1991.

[3] M. D. Miranda and M. Gerken, "A Hybrid QR-Lattice Least Squares Algorithm Using *a Priori* Errors," in *Proceedings of the 38 Midwest Symposium on Circuits and Systems*, pp. 983–986, Rio de Janeiro, RJ, Brazil, Aug. 1995.

[4] A. A. Rontogiannis and S. Theodoridis, "New Fast Inverse QR Least Squares Adaptive Algorithms," *Proceedings of the IEEE International Conference on Acoustic, Speech, and Signal Processing*, pp. 1412–1415, Detroit, MI, USA, 1995.

[5] J. A. Apolinário Jr. and P. S. R. Diniz, "A New Fast QR Algorithm Based on *a Priori* Errors," *IEEE Signal Processing Letters*, vol. 4, pp. 307–309, Nov. 1997.

[6] J. A. Apolinário Jr., M. G. Siqueira, and P. S. R. Diniz, "On Fast QR Algorithms Based on Backward Prediction errors: New Results and Comparisons," *Proceedings of the First Balkan Conference on Signal Processing, Communications, Circuits, and Systems*, Istanbul, Turkey, June 2000.

[7] M. D. Miranda and M. Gerken, "A Hybrid Least Squares QR-Lattice Algorithm Using *a Priori* Errors," *IEEE Transactions on Signal Processing*, vol. 45, pp. 2900–2911, Dec. 1997.

[8] P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*, Norwell, MA, USA, Kluwer Academic Press, 1997.

[9] P. S. R. Diniz and M. G. Siqueira, "Fixed-point error analysis of the QR-recursive least square algorithm," *IEEE Transactions on Circuits and Systems — Part II*, v. 42, pp. 334–348, May 1995.

[10] M. G. Siqueira, P. S. R. Diniz, and A. ALWAN, "Infinite Precision Analysis of the Fast QR Decomposition RLS Algorithm," *Proceedings of the IEEE International Symposium on Circuits and Systems*, London, UK, 1994.

[11] M. D. Miranda, L. Aguayo, and M. Gerken, "Performance of the *a Priori* and *a Posteriori* QR-LSL Algorithms in a Limited Precision Environment," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Munich, Germany, April 1997.

APPENDIX A: THE DETAILED FQR_POS_B ALG. V. 1

FQR_POS_B – Version 1 – Ref. [6]

Initialization:
ϵ = small positive value;
$\| \boldsymbol{e}_f(k) \| = \epsilon$;
$\boldsymbol{d}_{fq2}(k) =$ zeros$(N+1,1)$;
$\boldsymbol{d}_{q2}(k) =$ zeros$(N+1,1)$;
$\boldsymbol{cos\theta}(k) =$ ones$(N+1,1)$;
$\boldsymbol{sin\theta}(k) =$ zeros$(N+1,1)$;
$\boldsymbol{f}(k) =$ zeros$(N+1,1)$;
for $k = 1, 2, \ldots$
{ $e^{(0)}_{fq_1}(k+1) = x(k+1)$;
for $i = 1 : N+1$
{ $e^{(i)}_{fq_1}(k+1) = cos\theta_{i-1}(k) e^{(i-1)}_{fq_1}(k+1)$
$\qquad - sin\theta_{i-1}(k) \lambda^{1/2} d_{fq2_{N+2-i}}(k)$;
$d_{fq2_{N+2-i}}(k+1) = sin\theta_{i-1}(k) e^{(i-1)}_{fq_1}(k+1)$
$\qquad + cos\theta_{i-1}(k) \lambda^{1/2} d_{fq2_{N+2-i}}(k)$;
}
$e_{fq_1}(k+1) = e^{(N+1)}_{fq_1}(k+1)$;
$\| \boldsymbol{e}_f(k+1) \| = \sqrt{e^2_{fq_1}(k+1) + \lambda \| \boldsymbol{e}_f(k) \|^2}$;
$\| \boldsymbol{e}^{(N+1)}_f(k+1) \| = \| \boldsymbol{e}_f(k+1) \|$;
for $i = 1 : N+1$
{ $\| \boldsymbol{e}^{(N+1-i)}_f(k+1) \|$
$\qquad = \sqrt{\| \boldsymbol{e}^{(N+2-i)}_f(k+1) \|^2 + d^2_{fq2_i}(k+1)}$;
$cos\theta'_{f_{N+1-i}}(k+1)$
$\qquad = \| \boldsymbol{e}^{(N+2-i)}_f(k+1) \| \, / \, \| \boldsymbol{e}^{(N+1-i)}_f(k+1) \|$;
$sin\theta'_{f_{N+1-i}}(k+1)$
$\qquad = d_{fq2_i}(k+1) / \| \boldsymbol{e}^{(N+1-i)}_f(k+1) \|$;
}
$aux_0 = x(k+1) / \| \boldsymbol{e}^{(0)}_f(k+1) \|$;
$f_{N+1}(k+1) = aux_0$;
for $i = 1 : N$
{ $f_{N+1-i}(k+1) = \frac{f_{N+2-i}(k) - sin\theta'_{f_{i-1}}(k+1) aux_{i-1}}{cos\theta'_{f_{i-1}}(k+1)}$;
$aux_i = -sin\theta'_{f_{i-1}}(k+1) f_{N+1-i}(k+1)$
$\qquad + cos\theta'_{f_{i-1}}(k+1) aux_{i-1}$;
}
$\gamma^{(0)}(k+1) = 1$;
for $i = 1 : N+1$
{ $sin\theta_{i-1}(k+1) = f_{N+2-i}(k+1) / \gamma^{(i-1)}(k+1)$;
$cos\theta_{i-1}(k+1) = \sqrt{1 - sin^2\theta_{i-1}(k+1)}$;
$\gamma^{(i)}(k+1) = cos\theta_{i-1}(k+1) \gamma^{(i-1)}(k+1)$;
}
$\gamma(k+1) = \gamma^{(N+1)}(k+1)$;
$e^{(0)}_{q_1}(k+1) = d(k+1)$;
for $i = 1 : N+1$
{ $e^{(i)}_{q_1}(k+1) = cos\theta_{i-1}(k+1) e^{(i-1)}_{q_1}(k+1)$
$\qquad - sin\theta_{i-1}(k+1) \lambda^{1/2} d_{q2_{N+2-i}}(k)$;
$d_{q2_{N+2-i}}(k+1) = sin\theta_{i-1}(k+1) e^{(i-1)}_{q_1}(k+1)$
$\qquad + cos\theta_{i-1}(k+1) \lambda^{1/2} d_{q2_{N+2-i}}(k)$;
}
$e_{q_1}(k+1) = e^{(N+1)}_{q_1}(k+1)$;
$e(k+1) = e_{q_1}(k+1) \gamma(k+1)$;
}

APPENDIX B: THE DETAILED FQR_POS_B ALG. V. 2

FQR_POS_B – Version 2 – Ref. [2]

Initialization:
ϵ = small positive value;
$\| \boldsymbol{e}_f(k) \| = \epsilon$;
$\boldsymbol{d}_{fq2}(k) =$ zeros$(N+1,1)$;
$\boldsymbol{d}_{q2}(k) =$ zeros$(N+1,1)$;
$\boldsymbol{cos\theta}(k) =$ ones$(N+1,1)$;
$\boldsymbol{sin\theta}(k) =$ zeros$(N+1,1)$;
$\boldsymbol{f}(k) =$ zeros$(N+1,1)$;
for $k = 1, 2, \ldots$
{ $e^{(0)}_{fq_1}(k+1) = x(k+1)$;
for $i = 1 : N+1$
{ $e^{(i)}_{fq_1}(k+1) = cos\theta_{i-1}(k) e^{(i-1)}_{fq_1}(k+1)$
$\qquad - sin\theta_{i-1}(k) \lambda^{1/2} d_{fq2_{N+2-i}}(k)$;
$d_{fq2_{N+2-i}}(k+1) = sin\theta_{i-1}(k) e^{(i-1)}_{fq_1}(k+1)$
$\qquad + cos\theta_{i-1}(k) \lambda^{1/2} d_{fq2_{N+2-i}}(k)$;
}
$e_{fq_1}(k+1) = e^{(N+1)}_{fq_1}(k+1)$;
$\| \boldsymbol{e}_f(k+1) \| = \sqrt{e^2_{fq_1}(k+1) + \lambda \| \boldsymbol{e}_f(k) \|^2}$;
$\| \boldsymbol{e}^{(N+1)}_f(k+1) \| = \| \boldsymbol{e}_f(k+1) \|$;
for $i = 1 : N+1$
{ $\| \boldsymbol{e}^{(N+1-i)}_f(k+1) \|$
$\qquad = \sqrt{\| \boldsymbol{e}^{(N+2-i)}_f(k+1) \|^2 + d^2_{fq2_i}(k+1)}$;
$cos\theta'_{f_{N+1-i}}(k+1)$
$\qquad = \| \boldsymbol{e}^{(N+2-i)}_f(k+1) \| \, / \, \| \boldsymbol{e}^{(N+1-i)}_f(k+1) \|$;
$sin\theta'_{f_{N+1-i}}(k+1)$
$\qquad = d_{fq2_i}(k+1) / \| \boldsymbol{e}^{(N+1-i)}_f(k+1) \|$;
}
$aux_0 = \frac{\gamma(k) e_{fq_1}(k+1)}{\| \boldsymbol{e}_f(k+1) \|}$;
for $i = 1 : N+1$
{ $f_{i-1}(k+1) = cos\theta'_{f_{N+1-i}}(k+1) f_i(k)$
$\qquad - sin\theta'_{f_{N+1-i}}(k+1) aux_{i-1}$;
$aux_i = sin\theta'_{f_{N+1-i}}(k+1) f_i(k)$
$\qquad + cos\theta'_{f_{N+1-i}}(k+1) aux_{i-1}$;
}
$\frac{e_b(k+1)}{\| \boldsymbol{e}_b(k+1) \|} = f_0(k+1)$;
$f_{N+1}(k+1) = aux_{N+1}$;
$\gamma^{(0)}(k+1) = 1$;
for $i = 1 : N+1$
{ $sin\theta_{i-1}(k+1) = f_{N+2-i}(k+1) / \gamma^{(i-1)}(k+1)$;
$cos\theta_{i-1}(k+1) = \sqrt{1 - sin^2\theta_{i-1}(k+1)}$;
$\gamma^{(i)}(k+1) = cos\theta_{i-1}(k+1) \gamma^{(i-1)}(k+1)$;
}
$\gamma(k+1) = \gamma^{(N+1)}(k+1)$;
$e^{(0)}_{q_1}(k+1) = d(k+1)$;
for $i = 1 : N+1$
{ $e^{(i)}_{q_1}(k+1) = cos\theta_{i-1}(k+1) e^{(i-1)}_{q_1}(k+1)$
$\qquad - sin\theta_{i-1}(k+1) \lambda^{1/2} d_{q2_{N+2-i}}(k)$;
$d_{q2_{N+2-i}}(k+1) = sin\theta_{i-1}(k+1) e^{(i-1)}_{q_1}(k+1)$
$\qquad + cos\theta_{i-1}(k+1) \lambda^{1/2} d_{q2_{N+2-i}}(k)$;
}
$e_{q_1}(k+1) = e^{(N+1)}_{q_1}(k+1)$;
$e(k+1) = e_{q_1}(k+1) \gamma(k+1)$;
}