Finite Precision Analysis of the Fast QRD-RLS Lattice Algorithm

M. G. Siqueira and A. Alwan
Department of Electrical Engineering
University of California, Los Angeles
Los Angeles, CA 90024

P. S. R. Diniz
Programa de Engenharia Elétrica
COPPE/EE/Federal University of Rio de Janeiro
CP 68504 - Rio de Janeiro - RJ - CEP 21945 - Brazil

Abstract

This work derives relations for the mean squared values of the deviations in the FQRD-Lattice algorithm. The objective is determining the level of accuracy in the error signal of this adaptive filtering algorithm. It is shown that many internal variables have the same dynamic equations of the QRDL-RLS-based algorithms, and results of previous infinite precision studies can be used. Simulations showing the validity of the derived equations with different precisions are included.

1 Introduction

In the last years, many fast RLS algorithms based on QR Decomposition have been proposed [1], [2], [3]. Although many simulation studies have claimed that most of the algorithms in this class are numerically stable, only recent studies have shown that some algorithms in the FQRD-Lattice class are numerically stable. No accuracy analysis has yet been performed.

This work proposes a finite precision analysis for the FQRD-Lattice algorithm proposed by McWhorter [2]. The notation of this reference was followed. The FQDL-code for the FQRD-Lattice algorithm is shown in Table 1, showing all algorithmic steps labeled from step (T.1) through (T.18). It can be seen that this algorithm uses basically two functions, named rotor and circulator for performing all internal computations. In this table, the operator Q[ ] represents quantization.

Exploring the fact that only two basic operations are performed by this algorithm, a very regular structure can be derived as shown in Fig. 1. This figure uses squares to represent rotor cells (that perform rotations) and circles to represent circulator cells (that perform cosine/sine calculations). The small cells in the last stage represent multipliers.

2 Infinite Precision Analysis

This section derives mean-squared values for internal variables in the FQRD-Lattice algorithm. They are of fundamental importance for the finite precision analysis that will be performed in the next section.

2.1 Mean Squared Values of \( c_t^k(k) \) and \( s_t^k(k) \)

Previous studies [5] have shown that the mean squared values of the forward recursion cosines and sines are

\[
E[(c_t^k(k))^2] = \lambda
\]

\[
E[(s_t^k(k))^2] = 1 - \lambda
\]

Simulations for these variables in the QRD-RLS and for the FQRD-Lattice algorithms indicate that these approximations are reasonable.

2.2 Mean Squared Values of \( \beta_t^k(k) \), \( c_t^k(k) \) and \( \alpha_t^k(k) \)

Step (T.11) of the FQRD-Lattice algorithm implies that

\[
\beta_t^k(k) = \lambda^{1/2} c_t^k(k) (k-1) \beta_t^{k-1}(k-1) + s_t^k(k-1) s_t^{k-1}(k)
\]

\[
c_t^k(k) = -\lambda^{1/2} c_t^k(k) (k-1) \beta_t^{k-1}(k-1) + c_t^k(k-1) s_t^{k-1}(k)
\]

If it is supposed that the sines and cosines of the previous equations are uncorrelated with each other, and that the values of the sines are zero-mean, then it is possible to obtain the following relations

\[
E[(\beta_t^k(k))^2] = \lambda E[(c_t^k(k-1))^2] E[(\beta_t^{k-1}(k-1))^2]
\]

\[
+ E[(s_t^k(k-1))^2] E[(\beta_t^{k-1}(k-1))^2]
\]

\[
E[(c_t^k(k))^2] = \lambda E[(s_t^k(k))^2] E[(\beta_t^{k-1}(k))^2]
\]

\[
+ E[(c_t^k(k))^2] E[(\beta_t^{k-1}(k))^2]
\]

Substituting relations (1) and (2) in (5), it is possible to show that

\[
E[(\beta_t^{k-1}(k))^2] = \frac{E[(c_t^k(k))^2]}{1 + \lambda}
\]

Substituting relations (1), (2) and (7) on (6), we get

\[
E[(c_t^k(k))^2] = \frac{2 \lambda}{1 + \lambda} E[(c_t^{k-1}(k))^2]
\]

Since \( c_t^k(k) = x(k) \) according to equation (T.10), it follows

\[
E[(c_t^k(k))^2] = \sigma_x^2 \left[ \frac{2 \lambda}{1 + \lambda} \right]^k
\]

Consequently, according to (7)

\[
E[(\beta_t^k(k))^2] = \sigma_x^2 \left[ \frac{2 \lambda}{1 + \lambda} \right]^{k+1}
\]

\[0-7803-2570-295 4.00 ©1995 IEEE 1616\]
Table 1: C Language Pseudo-Code for FQRD-Lattice Algorithm

```c
void rotor (double x_in, double y_in, double x_out, double y_out, double c_in, double s_in)
{
    x_out = Q|λ|^1/2 x_in * c_in + y_in * s_in;  
    y_out = -Q|λ|^1/2 x_in * s_in + y_in * c_in;  
}

void cisor (double x_in, double y_in, double x_out, double y_out, double c_out, double s_out)
{
    double aux;
    x_out = Q|λ|^1/2 x_in * c_out + y_in * s_out;  
    c_out = Q|λ|^1/2 x_in / x_out;  
    s_out = Q|λ|^1/2 y_in / x_out;  
    y_out = Q|λ|^1/2 s_out;  
}

void FQRDLattice (double α(k), double d(k), double e(k))
{
    int i;
    double aux;
    e1(k) = e(k) / x(k);  
    e0(k) = d(k);  
    α0(k) = 1.0;  
    for (i = 1; i <= N + 1; i++)
    {
        cisor (α0(k-1), e0(k), α1(k-1), e1(k), α2(k-1), e2(k), α3(k-1), e3(k));  
        rotar (β0(k-1), e0(k), δ0(k-1), e1(k), β1(k-1), e2(k), β2(k-1), e3(k));  
        e0(k) = Q|λ|^1/2 (e0(k-1) - e2(k));  
        e1(k) = Q|λ|^1/2 (e1(k-1) - e3(k));  
        e2(k) = Q|λ|^1/2 (e2(k-1) - e0(k));  
        e3(k) = Q|λ|^1/2 (e3(k-1) - e1(k));  
    }
    e(k) = Q|λ|^1/2 (e0(k)) / x(k);  
}
```

The recursion formula for α(k-1) is given by

\[ α_{k-1}^c(k) = \sqrt{λ (α_{k-1}^c(k-1))^2 + [e_{k-1}^c(k)]^2} \]  
(11)

according to step (T.15). Supposing that α_{k-1}^c(k) and e_{k-1}^c(k) are statistically stationary for \( k \rightarrow \infty \), it follows

\[ E[(α_{k-1}^c(k))^2] = \frac{E[(e_{k-1}^c(k))^2]}{1 - λ} \]  
(12)

Using eq. (9) it follows

\[ E[(α_{k-1}^c(k))^2] = \frac{σ^2}{1 - λ} \left[ \frac{2λ}{1 + λ} \right]^i \]  
(13)

2.3 Mean Squared Values of \( c_i^c(k) \) and \( s_i^c(k) \)

According to the algorithm step (T.15), it follows that sines and cosines are calculated by

\[ c_i^c(k) = \frac{λ^{1/2} α_{k-1}^c(k-1)}{α_{k-1}^c(k-1)} \]  
(14)

\[ s_i^c(k) = \frac{α_{k-1}^c(k-1)}{α_{k-1}^c(k-1)} \]  
(15)

Thus, the mean squared values of the backward recursion sines and cosines are

\[ E[(c_i^c(k))^2] = \frac{λ E[(α_{k-1}^c(k-1))^2]}{E[(e_{k-1}^c(k))^2]} \]  
(16)

\[ E[(s_i^c(k))^2] = \frac{E[(e_{k-1}^c(k))^2]}{E[(e_{k-1}^c(k))^2]} \]  
(17)

In the above equations the averaging principle [7] was used. Considering that α_{k-1}^c(k) is statistically stationary as \( k \rightarrow \infty \) and using the fundamental trigonometric relation, it follows

\[ E[(c_i^c(k))^2] = \lambda \]  
(18)

\[ E[(s_i^c(k))^2] = 1 - \lambda \]  
(19)

Surprisingly, these mean square values are the same as the ones for the forward recursion sines and cosines, and different from those in the Fast QRD-RLS proposed by Bollinger [3], [6]. This is due to the different nature of these variables [2].

2.4 Mean Squared Values of \( β_i^L(k) \), \( c_i^L(k) \) and \( α_i^L(k) \)

The relations for \( β_i^L(k) \) and \( c_i^L(k) \) derived from step (T.16) are totally analogous to the ones for \( β_i^L(k) \) and \( c_i^L(k) \) shown in (3) and (4). Considering that the mean squared values for cosines and sines are the same in the backward and forward rotations and using the same statistical independence assumptions, it is possible to obtain

\[ E[(c_i^L(k))^2] = \sigma^2 \left[ \frac{2λ}{1 + λ} \right]^i \]  
(20)

\[ E[(β_i^L(k))^2] = \frac{σ^2}{1 + λ} \left[ \frac{2λ}{1 + λ} \right]^i \]  
(21)

\[ E[(α_i^L(k))^2] = \frac{σ^2}{1 + λ} \left[ \frac{2λ}{1 + λ} \right]^i \]  
(22)

2.5 Mean Squared Values of \( β_i(k) \) and \( ε_i(k) \)

Using properties of the triangulized input signal matrix [8], a very simple relationship for the mean square value of β_i(k) can be derived. It is supposed that the reference input d(k) is an MA process added with white gaussian measurement noise r(k) so that \( d(k) = w^0(k) * z(k) + r(k) \). In this case, \( w^0(k) \) is a sequence with the coefficients of the MA process with nonzero values for \( k = 0, \ldots, N \).

\[ E[(β_i^L(k))^2] = \frac{2λ}{1 + λ} \left[ \frac{2λ}{1 + λ} \right]^i \sigma^2 \]  
(23)

Using the norm conservation property of Givens Rotations, a relation between \( E[(ε_i^c(k))^2] \) and \( E[(β_i^L(k))^2] \) can be derived

\[ E[(ε_i^c(k))^2] = \sigma^2 ||w^0||^2 + (λ - 1) \sum_{j=0}^{N} E[(β_j^L(k))^2] \]  
(24)

where \( w^0 \) is a vector with \( N + 1 \) entries with the sequence \( w^0(k) \), \( k = 0, \ldots, N \).
3 Finite Precision Analysis

We assume that the input signal has been properly scaled in order to avoid overflow. Two's complement arithmetic is used for numeric representation. We assume in the analysis that no overflow occurs such that additions and subtractions do not introduce quantization errors. We also assume that multiplication, division, and square root operations introduce, respectively, quantization errors described by

\[ \eta_M(a, b) \triangleq a \bar{b} - Q[a \bar{b}] \]  \hspace{1cm} (25)

\[ \eta_D(a, b) \triangleq a/b - Q[a/b] \]  \hspace{1cm} (26)

\[ \eta_S(a) \triangleq \sqrt{a} - Q[\sqrt{a}] \]  \hspace{1cm} (27)

where \( a \) and \( b \) are scalars. Quantization is performed after additions whenever possible.

We also assume that the instantaneous quantizations are performed by rounding, for any type of arithmetic. The quantization error has zero mean and variance \( 2^{-(2B+1)/2} \), where \( B \) is the number of bits excluding the sign.

The accumulated quantization error in each quantity is defined as the difference between its value in infinite precision implementation and its value in finite precision implementation, as follows

\[ \Delta_\alpha(k) \triangleq \alpha(k) - a_Q(k) \]  \hspace{1cm} (28)

The input signal \( x(k) \) and the measurement error signal \( r(k) \) are assumed to be white gaussian white noise. The reference signal \( d(k) \) is assumed to be an MA process derived from \( x(k) \) and added with \( r(k) \).

3.1 Mean Squared Value of \( \Delta_\alpha(k) \)

According to step (T.10), the finite precision version of \( \alpha_\ell(k) \), denoted by \( \alpha_\ell(k) \), can be modeled as

\[ \alpha_\ell(k) = \sqrt{\lambda[\alpha_{\ell-1, Q}(k-1)]^2 + [\alpha_{\ell-1, Q}(k)]^2 + \eta_{\alpha_\ell}(k)} \]  \hspace{1cm} (29)

where \( \eta_{\alpha_\ell}(k) \) and \( \eta_{\alpha_\ell}(k) \) are instantaneous quantization errors. Using definitions (28), and considering only first order terms, it is possible to obtain

\[ \Delta_\alpha_{\ell-1}(k) = \frac{\lambda \alpha_{\ell-1, Q}(k-1)\Delta_\alpha_{\ell-1}(k-1) + \alpha_{\ell-1, Q}(k)}{\alpha_{\ell-1, Q}(k)} + \eta_{\alpha_{\ell-1}}(k) \]  \hspace{1cm} (30)

Squaring the above equation, supposing the deviations and instantaneous quantization noises are zero mean and uncorrelated with each other, and substituting equations (20) and (22), it follows

\[ E[\Delta_\alpha_{\ell-1}(k)]^2 = \left\{ \begin{array}{l} E[\Delta_\alpha_{\ell-1}(k)]^2 + \frac{\sigma^2}{1 + \lambda} + \frac{\sigma^2}{1 + \lambda} \frac{1 + \lambda}{2\lambda} \end{array} \right. \]  \hspace{1cm} (31)

The averaging principle [7] was used on the derivation.

3.2 Mean Squared Values of \( \Delta_\alpha_\ell(k) \) and \( \Delta_\alpha_\ell(k) \)

Using relations derived from step (T.15), definitions (28), (26) and first order approximations it is possible to write

\[ E[(\Delta_\alpha_\ell(k))^2] = \frac{\lambda E[\Delta_\alpha_\ell(k))^2]}{1 + \lambda} \]  \hspace{1cm} (32)

Using only first order terms, supposing that the deviations and quantization noise are all zero mean and uncorrelated with each other and using the averaging principle [7], it is possible to derive

\[ E[(\Delta_\alpha_\ell(k))^2] = \frac{\lambda E[\Delta_\alpha_\ell(k))^2]}{1 + \lambda} \]  \hspace{1cm} (33)

The same methodology can be used to obtain the mean squared value of \( \Delta_\alpha_\ell(k) \). The result is

\[ E[(\Delta_\alpha_\ell(k))^2] = \frac{\lambda E[\Delta_\alpha_\ell(k))^2]}{1 + \lambda} \]  \hspace{1cm} (34)

3.3 Mean Squared Values of \( \Delta_\beta_\ell(k) \) and \( \Delta_\epsilon_\ell(k) \)

The evolution of \( \beta_\ell(k) \) is described by eq. (5). If definitions (25) and (28) are used and second order errors are neglected, it is possible to write

\[ \Delta_\beta_{\ell-1}(k) = \lambda \frac{\beta_{\ell-1}(k-1) - \beta_{\ell-1}(k-1)}{1 - \lambda^2} \]  \hspace{1cm} (35)

Supposing that all the deviations and the instantaneous quantization noises are zero mean and uncorrelated with each other, it is possible to get

\[ E[(\Delta_\beta_{\ell-1}(k))^2] = \frac{\lambda E[\Delta_\beta_{\ell-1}(k))^2]}{1 + \lambda} \]  \hspace{1cm} (36)

Using eq. (6) and following the same steps, it can be shown that

\[ E[(\Delta_\epsilon_\ell(k))^2] = \frac{\lambda E[\Delta_\epsilon_\ell(k))^2]}{1 + \lambda} \]  \hspace{1cm} (37)

3.4 Mean Squared Values of \( \Delta_\alpha_\ell(k) \), \( \Delta_\alpha_\ell(k) \) and \( \Delta_\alpha_\ell(k) \)

According to step (T.13), it is possible to write

\[ \epsilon_{\alpha_\ell}(k) = \alpha_{\ell-1}(k) - \epsilon_{\alpha_\ell}(k) \]  \hspace{1cm} (38)

Using the same methodology of previous derivations, it is possible to get

\[ E[(\Delta_\alpha_\ell(k))^2] = \frac{\lambda E[\Delta_\alpha_\ell(k))^2]}{1 + \lambda} \]  \hspace{1cm} (39)
Step (T.10) implies that
\[ a_i(k) = c_i(k-1)a_{i-1}(k) \]  
(40)
The mean squared value for \( \Delta a_i(k) \) can be shown to be
\[ E[\Delta a_i(k)^2] = E[(\Delta c_i(k)^2)\lambda^{i-1}] + \lambda E[\Delta a_{i-1}(k)^2] + \sigma_n^2 \]  
(41)
It can be seen that \( c_i(k) \) is described by step (T.14). The mean squared value for \( \Delta c_i(k) \) can be calculated as
\[ E[\Delta c_i(k)^2] = E[\Delta a_i(k)^2]E[c_i(k)] + \sigma_n^2 \]
\[ + \frac{\sigma_n^2}{1 - \lambda} \left[ \frac{2\lambda}{1 + \lambda} \right]^{i-1} E[\Delta c_i(k)^2] \]  
(42)

3.5 Mean Squared Values of \( \Delta a_i^T(k), \Delta a_i^T(k), \Delta a_i^T(k) \), and \( \Delta c_i(k) \)
The relations, according to Table I, have very similar to their "dual" (forward or backward) counterparts. Using the same methodology described in the previous sections, the following relations are easily derived
\[ E[\Delta a_i^T(k)^2] = E[\Delta a_{i-1}(k)^2]1 + \frac{\lambda}{1 + \lambda^2} \left[ \frac{2\lambda}{1 + \lambda} \right]^{i-1} \]
\[ + \frac{\sigma_n^2}{1 - \lambda} \left[ \frac{2\lambda}{1 + \lambda} \right]^{i-1} E[\Delta a_i^T(k)^2] \]  
(43)
\[ E[\Delta a_i^T(k)^2] = \frac{E[\Delta a_{i-1}(k)^2]}{\sigma_n^2}(1 - \lambda) \left[ \frac{1 + \lambda}{2\lambda} \right]^{i-1} + \sigma_n^2 \]
\[ + \lambda E[\Delta a_{i-1}(k)^2] \sigma_n^2(1 - \lambda) \left[ \frac{1 + \lambda}{2\lambda} \right] \]  
(44)
\[ E[\Delta a_i^T(k)^2] = \left[ \frac{2\lambda}{1 + \lambda} \right]^{i-1} E[\Delta a_0^T(k)^2] + \sigma_n^2 \]  
(45)
\[ E[\Delta a_i^T(k)^2] = \left[ \frac{2\lambda}{1 + \lambda} \right]^{i-1} E[\Delta a_i^T(k)^2] + \lambda E[\Delta a_{i-1}(k)^2] \]
\[ + \frac{\lambda}{1 + \lambda} \left[ \frac{2\lambda}{1 + \lambda} \right]^{i-1} \]
\[ + \lambda E[\Delta a_i^T(k)^2] \sigma_n^2(1 - \lambda) \left[ \frac{1 + \lambda}{2\lambda} \right] \]  
(46)
The other two remaining values necessary for computing \( E[\Delta a_i^T(k)^2] \) can be easily derived from step (T.12) and are shown below
\[ E[\Delta a_i^T(k)^2] = \frac{\lambda}{1 - \lambda^2} E[\Delta a_i^T(k)^2]E[\Delta a_i^T(k)] \]
\[ + \frac{\lambda}{1 + \lambda^2} E[\Delta a_{i-1}(k)^2]E[\Delta c_i(k)] + \sigma_n^2 \frac{\lambda}{1 - \lambda^2} \]
\[ + \frac{\lambda}{1 + \lambda^2} E[\Delta a_i^T(k)^2] \]  
(47)
\[ E[\Delta a_i^T(k)^2] = \lambda E[\Delta a_i^T(k)^2] + E[\Delta a_i^T(k)^2]E[\Delta c_i(k)] \]
\[ + (1 - \lambda) E[\Delta a_{i-1}(k)^2] + \sigma_n^2 \]
\[ + \lambda E[\Delta a_i^T(k)^2]E[\Delta c_i(k)] \]  
(48)

3.6 Mean Squared Value of \( \Delta c(k) \)
Using step (T.18) the following relation can be derived
\[ E[\Delta c(k)^2] = \lambda^N E[\Delta a_{N+1}(k)^2] + E[\Delta a_{N+1}(k)^2] \]
\[ \frac{\sigma_n^2}{\lambda^N} + \sigma_n^2 \]  
(49)

Table 2: Simulation Results of \( E[\Delta c(k)^2] \) - Different Number of Bits.

<table>
<thead>
<tr>
<th>Number of Bits</th>
<th>Simulated (dB)</th>
<th>Calculated (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-70.69</td>
<td>-70.74</td>
</tr>
<tr>
<td>4</td>
<td>-124.84</td>
<td>-124.66</td>
</tr>
</tbody>
</table>

Table 3: Simulation Results of \( E[\Delta c(k)^2] \) - Different Values of \( \lambda \).

<table>
<thead>
<tr>
<th>( \lambda )</th>
<th>Simulated (dB)</th>
<th>Calculated (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.50</td>
<td>-57.74</td>
<td>-57.74</td>
</tr>
<tr>
<td>0.75</td>
<td>-59.30</td>
<td>-59.30</td>
</tr>
</tbody>
</table>

4 Simulations and Conclusions
A simple computational procedure was developed to predict the mean squared deviation in the error signal \( e(k) \).
Intensive simulations were performed to verify the accuracy of derived relations in both infinite and finite precision. Different values of \( \lambda, \sigma_n^2 \) and different number of bits were used.
In the simulations, 2's complement rounding was used, the input was white gaussian noise with \( \sigma_n^2 = -30 \) dB, \( \lambda = 0.99 \), the measurement error signal had variance \( \sigma_n^2 = -70 \) dB and an MA process of order 2 was used. A total of 10000 points were calculated in both finite precision and infinite precision and the last 9000 samples were averaged. The results of simulated and calculated results for \( E[\Delta c(k)^2] \) are displayed in Table 2.
Simulations with different values of \( \lambda \) were also performed. On these simulations, 15 bits were used. The input signals were the same as the in the previous simulations. The results are shown in Table 3.
By analyzing Tables 2 and 3, it can be verified that the simulated and calculated results are very close, with differences below 1 dB. This indicates that the proposed relations are sufficient for characterizing the performance of the FQRD-Lattice algorithm under finite precision.

References

1619