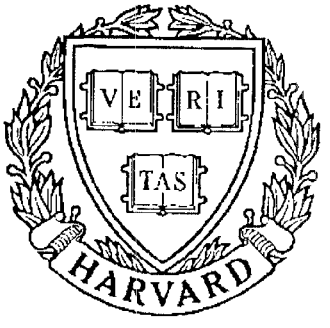
TECHNICAL RESEARCH REPORT

SYSTEMS RESEARCH CENTER



Supported by the National Science Foundation Engineering Research Center Program (NSFD CD 8803012), Industry and the University

Dynamic Range, Stability, and Fault-tolerant Capability of Finite-precision RLS Systolic Array Based on Givens Rotations

by K.J.R. Liu, S.F. Hsieh, K. Yao and C.T. Chiu

TR 90-77

Dynamic Range, Stability, and Fault-tolerant Capability of Finite-precision RLS Systolic Array Based on Givens Rotations

K.J.R. Liu, S.F. Hsieh[1], K. Yao[2], and C.T. Chiu

Electrical Engineering Department
Systems Research Center
University of Maryland
College Park, MD 20742
Ph: (301) 405-6619

ABSTRACT

The QRD RLS algorithm is generally recognized as having good numerical properties under finite-precision implementation. Also, it is very suitable for VLSI implementation since it can be easily mapped onto a systolic array. However, it is still unclear how to obtain the dynamic range of the algorithm such that a wordlength can be chosen to ensure correct operations of the algorithm. In this paper, we first propose a quasi-steady state model by observing the rotation parameters generated by boundary cells will eventually reach *quasi steady-state* regardless of the input data statistics if λ is close to one. With this model, we can obtain upper bounds of the dynamic range of processing cells. Thus, the wordlength can be obtained from upper bounds of the dynamic range to prevent overflow and to ensure correct operations of the QRD RLS algorithm. Then we reconsider the stability problem under quantization effects with more general analysis and obtain tighter bounds than given in a previous work [13]. Finally, two fault-tolerant problems, the missing error detection and the false alarm effect, that arise under finite-precision implementation are considered. Detail analysis on preventing missing error detection with a false alarm free condition is presented.

[1]Dept. of Communication Engineering, Nat'l Chiao Tung University, Hsinchu, Taiwan 30039
[2]Electrical Engineering Dept., UCLA, Los Angeles, CA 90024-1594

1 Introduction

Least-squares (LS) problems have been an integral part of modern signal processing and communications applications such as adaptive filtering, beamforming, array signal processing, channel equalization, etc.. Efficient implementation of the recursive LS (RLS) algorithm is desirable to meet the high throughput and speed requirement of modern signal processing. Among many techniques to implement the RLS algorithm, QR Decomposition (QRD) RLS algorithm is one of the most promising algorithms in that it is numerical stable as well as suitable for parallel processing implementation in a systolic array [1,8]. Gentleman and Kung [6] have proposed a QRD triangular systolic array based on Givens rotation, and McWhirter [20] used the systolic array to implement the QRD RLS algorithm efficiently. Since then, many researchers have considered and proposed various RLS algorithms (either constrainted or non-constrainted) based on methods such as Givens rotation, modified Gram-Schmidt, and Householder transformation for parallel processing architectures [3,4,9,10,14,16,21,24]. In [15], Liu and Yao also present an efficient algorithm-based fault-tolerant scheme that can be easily incorporated with the QRD RLS systolc array. An error resulting from a temporary or permanent faulty cell can be detected in real-time and the faulty cell can be reconfigurated out of service to prevent future contamination of the array. This make the systolic implementation of the RLS algorithm more attractive in the practical real-time applications. In the RSRE (Royal Signals and Radar Establishment) of the United Kingdom, a test bed of the QRD RLS systolic array has been built for radar applications [19]. Furthermore, the same systolic array can be used to solve SVD and eigenvalue problems [17,5] which is the heart of many signal processing applications such as high-resolution spectral estimation, direction of arrivals problems, and speech/image processing.

One of the most important problems that has not been solved is the dynamic range of the QRD RLS systolic algorithm. Without knowing the dynamic range of a algorithm, we are unable to predict the wordlength (number of bits per word) required to ensure correct operations. Furthermore, the wordlength of an algorithm is one of the most crucial factors in designing hardware and circuit [22] since the wordlength affects the hardware complexity. Usually, shorter arithmetic wordlength would realize smaller and faster hardware implementation [22]. At the same time, we also do not want the overflow happens

during the computation. Unfortunately, the dynamic range of the QRD RLS algorithm is still unclear.

In this paper, we first observe that the cosine parameters generated by boundary cells will eventually reach *quasi steady-state* if λ is close to one which is the usual case. We will show that the quasi steady-state and ensemble values of sine and cosine parameters are the same for all boundary cells. It is independent of the statistics of the input data sequence and the position of the boundary cell which generates the sine and cosine parameters. Simulation results are presented to support this observation. These results yield the tools needed to further investigate many properties of the QRD RLS systolic algorithm. Then, we can obtain upper bounds of the dynamic range of processing cells. Thus, lower bounds on the wordlength can be obtained from upper bounds of the dynamic range to prevent overflow and to ensure correct operations of the QRD RLS algorithm.

Though the QRD RLS algorithm is generally recognized as having good numerical properties such as numerical stability under finite-precision implementation [1,13], there is no mathematical proof of this until a recent paper by Leung and Haykin [13]. With the above results, we reconsider the stability problem under quantization effects with a more general analysis and obtain tighter bounds than given in previous work [13].

Given a finite wordlength, the computational precision is thus limited. Two important factors of the fault-tolerant capability, the *missing error detection* and the *false alarm* effects, resulting from the finite-precision implementation are also considered in this paper. They are of tradeoff in nature. We will present analyses to find a system that is capable of detecting given small error size without false alarm problem.

The organization of this paper is as follows. First, a brief review of the fault-tolerant QRD RLS systolic array is given in Section 2. Then quasi steady-state of the rotation parameters is discussed in Section 3. Dynamic range and lower bound on wordlength are derived in Section 4. Stability and quantization effects are considered in Section 5. Finally, the fault-tolerant capability is presented in Section 6 and a conclusion is given in Section 7.

2 Fault-tolerant QRD RLS Systolic Array

Without computing weight vector explicitly, the systolic implementation of the QRD RLS algorithm proposed by McWhirter [20] can obtain the optimal residuals efficiently. The

systolic array is shown in Fig.1. It consists of two parts: a triangular array for computing QRD and a linear column array called response array (RA) for computing LS residual. One of the major features of the array is that multiple RAs can be added to obtained optimal residuals for multiple desired responses.

In [15], Liu and Yao proposed a real-time concurrent error detection scheme for this systolic array based on the algorithm-based fault-tolerance [2,11]. The basic idea is that since the residuals of different desired responses can be computed simultaneously, an artificial desired response can be designed to detect an error produced by a faulty processor. [15] has shown that if the artificial desired response is designed as some proper combinations of the input data, the output residual of the system will be zero if there is no fault. However, any occurring fault in the system will cause the residual to be non-zero and the fault can be detected in real-time. The fault-tolerant QRD RLS systolic array is shown in Fig.2. As we can see, above the QRD triarray, a horizontal linear array called *encoding array* is used to add up the incoming row (the checksum) to be the artificial desired response. The processing cell of the encoding array is an adder which adds both inputs and passes to the next cell. artificial desired response then serves as the input to the new RA called *error detection array* (EDA) at the right side of the QRD triarray. The output of the EDA, e_0, now serves as the error detector. If there is no error, e_0 will always be zero. Whenever there is a faulty cell occurs during the computation, the error generated by the faulty cell will cause $e_0 \neq 0$ and thus the error is detected in real-time [15]. All of these results are based on the assumption that the computation is infinite precision. Under finite-precision computation, there are two major effects, the missing error detection and false alarm effect, which will be considered in a later Section.

3 Quasi Steady-state Model

From the updated recursive equation of the boundary cell (see Fig.1), we have

$$r^2(k+1) = \lambda^2 r^2(k) + x^2(k) = \sum_{i=0}^{k} \lambda^{2i} x^2(k-i), \tag{1}$$

where $0 < \lambda \leq 1$ is the exponentially forgetting factor [8]. Assume the input sequence $\{x\}$ is zero-mean with variance σ^2, the expected mean of $r^2(k+1)$ is

$$E[r^2(k+1)] = \sum_{i=0}^{k} \lambda^{2i} E(x^2(k-i)) = \sigma^2 \frac{1-\lambda^{2(k+1)}}{1-\lambda^2}. \tag{2}$$

When k is very large

$$\lim_{k\to\infty} E[r^2(k)] = \frac{\sigma^2}{1-\lambda^2}. \tag{3}$$

Since $\sqrt{\cdot}$ is a concave function, from Jensen's inequality [23]

$$\lim_{k\to\infty} E(r(k)) \leq \lim_{k\to\infty} \sqrt{E[r^2(k)]} = \frac{\sigma}{\sqrt{1-\lambda^2}}, \tag{4}$$

and from (1)

$$\frac{|x_{min}|}{\sqrt{1-\lambda^2}} \leq \lim_{k\to\infty} r(k) \leq \frac{|x_{max}|}{\sqrt{1-\lambda^2}} \tag{5}$$

where $|x_{max}|$ and $|x_{min}|$ are the maximum and minimum values of sequence $\{|x|\}$.

The cosine parameter of the Givens rotation is computed by $c(k+1) = \lambda r(k)/r(k+1)$. The steady-state of this parameter exists if $\lim_{k\to\infty} c(k)$ exists. For the sequence $\{c(\cdot)\}$ to have a steady-state, we need $\lim_{k\to\infty} r(k)/r(k+1) = \alpha$, where α is a constant. If $\alpha < 1$, then the sequence $\{r(\cdot)\}$ is unbounded which conflicts with (5) that indicates $\{r(\cdot)\}$ should be bounded; if $\alpha > 1$, then $\lim_{k\to\infty} r(k) = 0$ which, again, conflicts with (5). Therefore, α has to be a unity to guarantee the steady-state of $\{c(\cdot)\}$ exists. That is,

$$\lim_{k\to\infty} \frac{r(k)}{r(k+1)} = 1, \tag{6}$$

and the steady-state value of cosine, if exists, is

$$\lim_{k\to\infty} c(k) = \lim_{k\to\infty} \frac{\lambda r(k-1)}{r(k)} = \lambda. \tag{7}$$

From (1), we can see that if $\lambda = 1$, then $\lim_{k\to\infty} r(k) \to \infty$ such that $\lim_{k\to\infty} r(k)/r(k+1) = 1$. In this case, though the steady-state of $\{c(\cdot)\}$ exists, $\{r(\cdot)\}$ is unbounded. Usually λ is chosen between .99 and 1 which is very close to one[3]. When we update $r(k)$ to $r(k+1)$ using (1), a λ portion of $r(k)$ is forgotten and an input $x(k)$ is added into it. If λ is close to one, when k is very large, $r(k)$ will come close to $r(k+1)$ and the input $x(k)$ plays less and less significant role in computing $r(k+1)$ as the case when $\lambda = 1$. It is obvious that

$$\lim_{k\to\infty} Er(k) = \lim_{k\to\infty} Er(k+1).$$

[3]For different expressions as in [8,13,20], λ is between .98 and 1

Therefore, from the averaging principle [18] which has been used successfully in many situations, the expected cosine can be approximated by

$$\lim_{k\to\infty} Ec(k) \simeq \lambda \frac{Er(k-1)}{Er(k)} = \lambda. \tag{8}$$

When λ is close to one, from above discussions, we have

$$\lim_{k\to\infty} c(k) = \lim_{k\to\infty} \frac{\lambda r(k)}{r(k+1)} = \lambda + \delta(\lambda, x), \tag{9}$$

where $\delta(\lambda, x)$ represents the small deviation due to the forgotten λ portion of r and input of x. If δ is very small such that it is negligible when k is large, we say that the sequence $\{c(\cdot)\}$ reaches the *quasi steady-state.*

Generally, it is almost impossible to quantitatively describe $\delta(\lambda, x)$. Simulations will be used to demonstrate how small the quantity δ is. Here we model the input signal to the systolic array as a second-order AR process described by

$$u(n) + a_1 u(n-1) + a_2 u(n-2) = v(n), \tag{10}$$

where $v(n)$ is a white Gaussian noise process of zero mean and unit variance. Choose of different AR parameters a_1 and a_2 will give us different realizations of the AR process [8]. In our simulations, three different categories of signal are encountered. The first category consists of three stationary AR processes which are AR1 ($a_1 = -0.1$, $a_2 = -0.8$), AR2 ($a_1 = 0.1$, $a_2 = -0.8$) with real roots and AR3 ($a_1 = -0.975$, $a_2 = 0.95$) with complex-conjugate roots. The second category is a non-stationary AR process, AR4 ($a_1 = -0.6$, $a_2 = -0.5$), and the third category is a white Gaussian noise process, WN, with zero mean and unit variance. All of the AR processes are normalized to unit variance. Table 1 shows the mean distribution of cosine parameters for different input data with different λ values. This table justifies the result in (8). Table 2 shows the variance distribution of δ for different input data with different λ value. The values of those variances are in the order of 10^{-4} to 10^{-6} which implies that δ is indeed very small. They can be closely approximated by using quadratic polynomials as follows,

$$\begin{aligned}
AR1: \quad \sigma_c^2(\lambda) &= 1.5938 - 3.182\lambda + 1.5882\lambda^2 \\
AR2: \quad \sigma_c^2(\lambda) &= 1.5991 - 3.1919\lambda + 1.5928\lambda^2 \\
AR3: \quad \sigma_c^2(\lambda) &= 1.5812 - 3.1595\lambda + 1.5784\lambda^2
\end{aligned}$$

$$
\begin{aligned}
AR4: \quad \sigma_c^2(\lambda) &= 1.4492 - 2.8936\lambda + 1.4444\lambda^2 \\
AR5: \quad \sigma_c^2(\lambda) &= 1.6437 - 3.2904\lambda + 1.6431\lambda^2,
\end{aligned} \tag{11}
$$

where $0.98 \leq \lambda < 1$.

We can see, though the statistics of input data are different, the variances can be described by λ in a very similar way (see Fig.3). This means, when λ is close to one and the quasi steady-state is reached, the size of the variation δ is majorly governed by λ instead of the statistics of the input data. Fig.3 shows the plots of the variances in dB scale.

With these results, we conclude that the sequence $\{c(\cdot)\}$ reaches the quasi steady-state regardless the input statistics if λ is close to one. Thus, we can write

$$
\begin{aligned}
\lim_{k\to\infty} c(k+1) &\simeq \lim_{k\to\infty} Ec(k+1) \simeq \lambda, \\
\lim_{k\to\infty} s(k+1) &\simeq \lim_{k\to\infty} Es(k+1) \simeq \sqrt{1-\lambda^2}.
\end{aligned} \tag{12}
$$

The quasi steady-state and ensemble values of sine and cosine parameters are the same for all boundary cells. It is independent of the statistics of the input data sequence and the position of the boundary cell which generates the sine and cosine parameters. These results yield the tools needed to further investigate many properties of the QRD RLS systolic algorithm.

4 Dynamic Range and Lower Bound on Wordlength

Denote PE_{ij} as the (i,j) processing cell of the array, from Fig.1 the dynamic range of the content of the boundary cell PE_{11} can be upper bounded by

$$
\lim_{k\to\infty} r_{11}^2(k+1) = \lim_{k\to\infty} \sum_{i=0}^{k} \lambda^{2i} x^2(k-i) \leq \lim_{k\to\infty} x_{max}^2 \sum_{i=0}^{k} \lambda^{2i} = \frac{x_{max}^2}{1-\lambda^2}. \tag{13}
$$

Therefore,

$$
\lim_{k\to\infty} |r_{11}(k)| \leq \frac{|x_{max}|}{\sqrt{1-\lambda^2}} \triangleq \Re. \tag{14}
$$

For internal cell PE_{1j} (of the first row), we have

$$
\begin{aligned}
|r_{1j}(k+1)| &= |s(k)x(k) + c(k)\lambda r_{1j}(k)| \\
&= |s(k)x(k) + c(k)\lambda[s(k-1)x(k-1) + c(k-1)\lambda r_{1j}(k-1)]| \\
&\leq \sum_{i=0}^{k} \lambda^i |x(k-i)s(k-i)| \prod_{j=0}^{i-1} c(k-j)
\end{aligned}
$$

$$\leq \quad |x_{max}| \sum_{i=0}^{k} \lambda^i |s(k-i)| \prod_{j=0}^{i-1} c(k-j) \tag{15}$$

From the basic relation between the geometric mean and the arithmetic mean, we know

$$(\frac{a_1 + a_2 + \cdots + a_n}{n})^n \geq a_1 \cdot a_2 \cdots a_n. \tag{16}$$

If n is large enough, then from the law of large number, we know

$$\lim_{n\to\infty} \frac{a_1 + a_2 + \cdots + a_n}{n} \to E(a).$$

Therefore,

$$E(a)^n \geq \prod_{i=1}^{n} a_i,$$

when n is large. We can further simplify the bound for $k \to \infty$ by using this inequality as follows,

$$\begin{aligned} \lim_{k\to\infty} |r_{1j}(k+1)| &\leq |x_{max}| \lim_{k\to\infty} \sum_{i=0}^{k} \lambda^i s(k-i) E(c(k-i))^i \\ &\simeq |x_{max}| \sum_{i=0}^{k} \lambda^{2i} \cdot \sqrt{1-\lambda^2} = \frac{|x_{max}|}{\sqrt{1-\lambda^2}} = \Re. \end{aligned} \tag{17}$$

From (14) and (17), we can see the steady-state dynamic range of the first row is upper bounded by $\Re$ for both boundary and internal cells. The dynamic range of the second row depends on the output of internal cells of the first row. Denote the output of the first row as x_{out}, From Fig.1 we have

$$x_{out}(k+1) = c(k)x(k) - s(k)\lambda r(k). \tag{18}$$

The first term of the right-hand side of (18) can be bounded by

$$\lim_{k\to\infty} |c(k)x(k)| \leq \lambda |x_{max}| \tag{19}$$

and from (17) the second term is bounded by

$$\lim_{k\to\infty} |s(k)\lambda r(k)| \leq \sqrt{1-\lambda^2} \cdot \lambda \frac{|x_{max}|}{\sqrt{1-\lambda^2}} = \lambda |x_{max}|. \tag{20}$$

There are two possible cases:

Case 1: Highly fluctuated input

The value of $x(k)$ may vary differently from time to time such that for most of the time, $s(k)r(k)$ may have opposite sign of $x(k)$. For this case

$$\lim_{k\to\infty} |x_{out}(k)| \leq 2\lambda |x_{max}|. \tag{21}$$

Case 2: Smooth input

For this case, the input data sequence does not change its value rapidly, therefore $s(k)r(k)$ may have the same sign as $x(k)$ for most of the time. The bound is

$$\lim_{k\to\infty} |x_{out}(k)| \leq \lambda |x_{max}|. \tag{22}$$

From (14) and (17), it is obvious the steady-state dynamic range of the second row is bounded by

$$\lim_{k\to\infty} |r_{2j}(k)| \leq \frac{2\lambda |x_{max}|}{\sqrt{1-\lambda^2}} = 2\lambda \Re, \tag{23}$$

for the highly fluctuated input and

$$\lim_{k\to\infty} |r_{2j}(k)| \leq \lambda \Re, \tag{24}$$

for the smooth input. From above results, the steady-state dynamic range of the m^{th} row is bounded by

$$\lim_{k\to\infty} |r_{mj}(k)| \leq (2\lambda)^{m-1} \cdot \Re, \tag{25}$$

for the highly fluctuated input and

$$\lim_{k\to\infty} |r_{mj}(k)| \leq (\lambda)^{m-1} \Re, \tag{26}$$

for the smooth input. For Case 1, the dynamic range is increasing exponentially with a factor of 2λ, and for Case 2, however, decreasing exponentially with a factor of λ.

From (25) and (26), we can see that the dynamic range may increase or decrease with row. It depends on how fast changing the input signal is. For a given row, its dynamic range may follow (25) for some periods (increasing) and then switch to (26) for some periods (decreasing). Either way, (25) represents the worst case scenario.

Denote B_m as the wordlength of the m^{th} row, to prevent overflow and to ensure the correct operation of the QRD RLS algorithm, we require $2^{B_m} \geq (2\lambda)^{m-1}\Re$ for fixed point operation, and therefore

$$B_m \geq \lceil (m-1)(1+\log_2 \lambda) + \log_2 \Re \rceil. \tag{27}$$

For the fluctuated input, when $(2\lambda)^{n-1} = 2$, one more bit is needed for the wordlength of the following rows. The number of rows n for each bit increase is

$$n = \lceil 1 + \frac{1}{1 + \log_2 \lambda} \rceil, \tag{28}$$

which is a monotonically decreasing function of λ. If $\lambda \leq 0.5$, then there is no such m exists. That is, the wordlength of the array can be fixed at $\Re$ without the overflow problem. For smooth input, when $\lambda^{n-1} = \frac{1}{2}$, one bit can be discarded from the wordlength of the following rows. The number of rows n for each bit decrease is

$$n = \lceil 1 - \frac{1}{\log_2 \lambda} \rceil, \tag{29}$$

which is a monotonically increasing function of λ. For $\lambda \leq 0.5$, $n = 2$. That is, for every two rows we can discard one bit for the wordlength.

Our simulations verified the above results. Here we provide some examples. Fig.4 shows a simulation of the contents of internal and boundary cells of different rows as well as the upper bound $\Re$ under AR3 input signal for $\lambda = 0.991$ and $p = 3$. Table 3 compares the upper bound $\Re$ and the maximum value of contents of boundary and internal cells for different input signals. From these, we can see that $\Re$ is a good upper bound for both boundary and internal cells. From (27), we can choose the minimum wordlengths for the AR3 input signal. We found that it needs three bits for the wordlength of the first row, four bits for the second row, and five bits for the third row. As shown in Fig.5, the resultant contents are almost identical to that of Fig.4 which is the results of double-precision implementation.

5 Stability and Quantization Effect

In this Section, we consider the stability under quantization effect. Here, the stability is defined in the sense of bounded input/bounded output (BIBO) as in [13]. From (21) and (22), the output of the m^{th} row is bounded by

$$\lim_{k \to \infty} |x_{out_m}| \leq (2\lambda)^{m-1} |x_{max}|, \tag{30}$$

for the highly fluctuated input and

$$\lim_{k \to \infty} |x_{out_m}| \leq \lambda^{m-1} |x_{max}| \tag{31}$$

for the smooth input.

The order of least-squares p is always finite. The output of the last row of the QR triarray is bounded, in the worst case, by $\lim_{k\to\infty} |x_{out_p}| \leq (2\lambda)^{p-1}|x_{max}|$. The residual is then asymptotically bounded by

$$\lim_{k\to\infty} |e(k)| = \lim_{k\to\infty} \gamma(k)|x_{out_p}(k)| \leq (2\lambda)^{p-1}|x_{max}|, \tag{32}$$

where $\gamma(k) = \prod_{i=1}^{p} c_i(k)$ and $c_i's$ are related cosine parameters [20]. Thus, for $\lambda < 1$, if the input data are bounded, that is, $|x_{max}| < \infty$, the output is always bounded. The QRD RLS systolic array constitutes a BIBO stable system under unlimited precision implementation. Practically, the wordlength of each processing cell is finite-precision. Leung and Haykin [13] first considered the stability under this effect and showed the QRD RLS algorithm is stable under finite-precision implementation. Here we reconsider this problem and give a more general analysis and a tighter bound.

Denote $Q(\cdot)$ as the quantization operator and $\check{x}$ as the quantized value of x. Since the quantization error for the additions of quantized parameters is much smaller than that of the multiplications of them, to make the analysis simplier, we may express the quantization error for additions as

$$Q(\sum_{i=1}^{n} \check{a}_i) = \sum_{i=1}^{n} \check{a}_i + \delta_n \tag{33}$$

From (1), the square of the quantized content of the boundary cell is

$$\check{r}^2(k+1) = Q(Q(\check{\lambda}^2\check{r}^2(k)) + Q(\check{x}^2(k))) = \sum_{i=0}^{k} Q(\check{\lambda}^{2i}\check{x}^2(k-i)) + \delta_{k+1}. \tag{34}$$

The quantization operator Q is a bounded operator such that $|Q(x)| \leq K|x|$ for all x and some K [13], (34) can be bounded by

$$\begin{aligned} |\check{r}^2(k+1)| &\leq K_0|\check{\lambda}^{2k}\check{x}^2(0)| + K_1|\check{\lambda}^{2(k-1)}\check{x}^2(1)| + \cdots + K_k|\check{x}^2(k)| + \delta_{k+1} \\ &\leq K_{max} \cdot \check{x}^2_{max}(1 + \check{\lambda}^2 + \cdots + \check{\lambda}^{2k}), \end{aligned} \tag{35}$$

where $\check{x}_{max}$ is the maximum quantized value of sequence $\check{x}$. The asymptotic behavior can be obtained by taking the limit on both sides, it becomes

$$\lim_{k\to\infty} |\check{r}^2(k)| \leq K_{max} \cdot \check{x}^2_{max} \frac{1}{1-\check{\lambda}^2}. \tag{36}$$

for the smooth input.

The order of least-squares p is always finite. The output of the last row of the QR triarray is bounded, in the worst case, by $\lim_{k\to\infty} |x_{out_p}| \leq (2\lambda)^{p-1}|x_{max}|$. The residual is then asymptotically bounded by

$$\lim_{k\to\infty} |e(k)| = \lim_{k\to\infty} \gamma(k)|x_{out_p}(k)| \leq (2\lambda)^{p-1}|x_{max}|, \tag{32}$$

where $\gamma(k) = \prod_{i=1}^{p} c_i(k)$ and $c_i's$ are related cosine parameters [20]. Thus, for $\lambda < 1$, if the input data are bounded, that is, $|x_{max}| < \infty$, the output is always bounded. The QRD RLS systolic array constitutes a BIBO stable system under unlimited precision implementation. Practically, the wordlength of each processing cell is finite-precision. Leung and Haykin [13] first considered the stability under this effect and showed the QRD RLS algorithm is stable under finite-precision implementation. Here we reconsider this problem and give a more general analysis and a tighter bound.

Denote $Q(\cdot)$ as the quantization operator and $\check{x}$ as the quantized value of x. Since the quantization error for the additions of quantized parameters is much smaller than that of the multiplications of them, to make the analysis simplier, we may express the quantization error for additions as

$$Q(\sum_{i=1}^{n} \check{a}_i) = \sum_{i=1}^{n} \check{a}_i + \delta_n \tag{33}$$

From (1), the square of the quantized content of the boundary cell is

$$\check{r}^2(k+1) = Q(Q(\check{\lambda}^2\check{r}^2(k)) + Q(\check{x}^2(k))) = \sum_{i=0}^{k} Q(\check{\lambda}^{2i}\check{x}^2(k-i)) + \delta_{k+1}. \tag{34}$$

The quantization operator Q is a bounded operator such that $|Q(x)| \leq K|x|$ for all x and some K [13], (34) can be bounded by

$$\begin{aligned} |\check{r}^2(k+1)| &\leq K_0|\check{\lambda}^{2k}\check{x}^2(0)| + K_1|\check{\lambda}^{2(k-1)}\check{x}^2(1)| + \cdots + K_k|\check{x}^2(k)| + \delta_{k+1} \\ &\leq K_{max} \cdot \check{x}^2_{max}(1 + \check{\lambda}^2 + \cdots + \check{\lambda}^{2k}), \end{aligned} \tag{35}$$

where $\check{x}_{max}$ is the maximum quantized value of sequence $\check{x}$. The asymptotic behavior can be obtained by taking the limit on both sides, it becomes

$$\lim_{k\to\infty} |\check{r}^2(k)| \leq K_{max} \cdot \check{x}^2_{max} \frac{1}{1-\check{\lambda}^2}. \tag{36}$$

Therefore, the quantized content is

$$\begin{aligned}\lim_{k\to\infty}|\check{r}(k)| &= \lim_{k\to\infty} Q(\sqrt{\check{r}^2(k)}) \\ &\leq K'_{max}\frac{|\check{x}_{max}|}{\sqrt{1-\check{\lambda}^2}} \triangleq K'_{max}\check{\Re}. \end{aligned} \tag{37}$$

With the same arguments as in Section 3, we then have

$$\lim_{k\to\infty}\frac{\check{r}(k)}{\check{r}(k+1)} \simeq 1, \tag{38}$$

if $\check{\lambda}$ is close to 1. and the quantized steady-state value of cosine is

$$\lim_{k\to\infty}\check{c}(k+1) = \lim_{k\to\infty}\frac{\check{\lambda}\check{r}(k)}{\check{r}(k+1)} \simeq \check{\lambda}, \tag{39}$$

and the quantized steady-state value of sine is

$$\lim_{k\to\infty}\check{s}(k+1) = Q(\sqrt{1-\check{\lambda}}).$$

Analogous to Section 3, we can further obtain $\lim_{k\to\infty} E\check{c}(k) = \check{\lambda}$ and $\lim_{k\to\infty} E\check{s}(k) = Q(\sqrt{1-\check{\lambda}})$.

Now consider the quantized content of the internal cell, from (15)

$$\begin{aligned}|\check{r}_{1j}(k+1)| &= |Q(Q(\check{s}(k)\check{x}(k)) + Q(\check{c}(k)\check{\lambda}\check{r}(k))| \\ &= \sum_{i=0}^{k} Q(\check{\lambda}^i|\check{x}(k-i)\check{s}(k-i)|\prod_{j=0}^{i-1}\check{c}(k-j)) + \delta_{k+1} \\ &\leq K"_{max}|\check{x}_{max}|\sum_{i=0}^{k}\check{\lambda}^i|\check{s}(k-i)|\prod_{j=0}^{i-1}\check{c}(k-j), \end{aligned} \tag{40}$$

where $K"_{max}$ results from quantization error including δ_{k+1}. From Section 4 and (39), (40), the quantized steady-state dynamic range of the internal cell is bounded by

$$\lim_{k\to\infty}|\check{r}_{1j}(k)| \leq K"_{max}\frac{|\check{x}_{max}|}{\sqrt{1-\check{\lambda}}} = K"_{max}\check{\Re}. \tag{41}$$

The output of the m^{th} row is bounded, under the quantization effect, by

$$\lim_{k\to\infty}|\check{r}_{1j}(k)| \leq K"_{max}(2\check{\lambda})^{m-1}\check{\Re} \tag{42}$$

for the highly fluctuated input and

$$\lim_{k\to\infty}|\check{r}_{1j}(k)| \leq K"_{max}(\check{\lambda})^{m-1}\check{\Re} \tag{43}$$

for smooth input.

From these results, the quantized asymptotic value of the residual can be obtained as

$$\lim_{k\to\infty} |\check{e}(k)| \leq K^{\cdot\cdot}{}_{max}(2\check{\lambda})^{p-1}\check{\Re}. \tag{44}$$

Thus, if $\lambda < 1$ and the input data are bounded, the QRD RLS systolic array constitutes a BIBO stable system under the quantization effect.

6 Finite Wordlength Effects of Fault-tolerant Capability

In this Section, we discuss the finite-length effects of the fault-tolerant capability. The first problem is that of missing error detection which results from the cumulative multiplications of the cosine value with a small error. Since $|cosine| \leq 1$, the error will be then getting smaller and smaller. With a finite-precision implementation, this may result in a failure of error detection. The minimum wordlength to circumvent this problem is then derived. The second problem is called the false alarm. With the quantization effects, the system without fault may produce quantization errors to cause the false alarm problem. A threshold device is then introduced to tackle this problem.

6.1 Missing Error Detection

By *missing error detection* we mean that a small error generated by a faulty processing cell is not detected due to the finite-precision computation. Assume a fault occurs in an internal cell $PE_{ij}, i \neq j$, at a faulty moment. The output of this faulty cell is thus erroneous and can be described by $x^{\epsilon}_{out} = x_{out} + \delta$, where x_{out} is the fault-free output and δ is the error generated by the fault. The error propagation path can be described by

$$PE_{ij} \rightarrow PE_{(i+1)j} \rightarrow \cdots \rightarrow PE_{jj},$$

and then PE_{kl}, $k \geq j$, $l \geq j$ are all contaminated [15]. From the operations executed by the internal cell, the error is modified to $c_{i+1}\delta$ by $PE_{(i+1)j}$ and the cumulative modifications of the error before reaching the boundary cell, PE_{jj}, is

$$\eta = \delta \prod_{k=i+1}^{j-1} c_k, \tag{45}$$

where c_i is the cosine parameter generated by the boundary cell PE_{ii}. Let c'_j and s'_j denote the erroneous c_j and s_j respectively. The c'_j and s'_j are then given by

$$c'_j = \frac{\lambda r}{\sqrt{\lambda^2 r^2 + (x_{in} + \eta)^2}}, \qquad s'_j = \frac{x_{in} + \eta}{\sqrt{\lambda^2 r^2 + (x_{in} + \eta)^2}}. \tag{46}$$

In this case, s'_j is no longer proportional to x_{in}, $\underline{a}(j)$ will not be zeroed out by the j^{th} cell of the EDA [15]. The size of the error generated by this cell is

$$\eta_j = c'_j x_{in} - s'_j \lambda r = -\frac{\lambda r \eta}{\sqrt{r'^2 + 2\eta x_{in} + \eta^2}} = -c'_j \eta, \tag{47}$$

where $r' = \sqrt{\lambda^2 r^2 + x_{in}^2}$ is the new updated uncontaminated value of the content of PE_{jj}. When η_j propagates down to the output of the EDA, η_j is influenced by the contaminated cosines c' of each following row. The error output at e_0 due to an error δ generated at PE_{ij} is then given by

$$\begin{aligned} e_0^{\delta}(i,j) &= -\gamma \prod_{m=j+1}^{p} c'_m \eta_j = -\gamma \prod_{m=j}^{p} c'_m \eta \\ &= -\gamma \prod_{k=i+1}^{j-1} c_k \cdot \prod_{m=j}^{p} c'_m \delta. \end{aligned} \tag{48}$$

where $\gamma = \prod_{l=1}^{j-1} c_l \prod_{k=j}^{p} c'_k$ [20]. It becomes

$$e_0^{\delta}(i,j) = -\prod_{l=1}^{i} c_l \prod_{k=i+1}^{j-1} c_k^2 \prod_{m=j}^{p} c'^2_m \delta. \tag{49}$$

Next, assume a fault occurs in a boundary cell, PE_{jj}, $1 \le j \le p$, at the faulty moment. Both erroneous c'_j and s'_j produced by PE_{jj} can be written by

$$c'_j = \frac{\lambda r + \delta_c}{r'_\epsilon}, \qquad s'_j = \frac{x_{in} + \delta_s}{r'_\epsilon}, \tag{50}$$

where δ_c and δ_s represent errors in the numerators while r'_ϵ represents the erroneous content of the denominators of c_j a nd s_j. The error produced by the j^{th} cell of the EDA is then given by

$$\eta_j = c'_j x_{in} - s'_j \lambda r = \frac{x_{in}\delta_c - \lambda r \delta_s}{r'_\epsilon}, \tag{51}$$

and the output error at e_0 due to a faulty boundary cell is given by

$$\begin{aligned} e_0^{\delta}(j,j) &= \gamma \prod_{m=j+1}^{p} c'_m \cdot \frac{x_{in}\delta_c - \lambda r \delta_s}{r'_\epsilon} \\ &= \prod_{l=1}^{j} c_l \cdot \prod_{m=j+1}^{p} c'^2_m \cdot \eta_j. \end{aligned} \tag{52}$$

From (49) and (52), we can see that $e_0^\delta \neq 0$, under unlimited precision condition, if there is a fault occurs in the system, except when $u_{in}\delta_c = \lambda r \delta_s$ in (51). However, this is unlikely to happen. From [15,20], we have $0 < c_i \leq 1$. The error may not be detected after multiple multiplications of c_i in (49) and (52) under finite-precision implementation. It is obvious there is no such problem when δ is large. Since r in (46) tends to be a large number asymptotically, it is reasonable to assume the error size δ generated by a fault is much smaller than r when δ is small. Under this circumstance, from (46), we have $c'_j \cong c_j$. In the quasi steady-state, the asymptotic behavior of erroneous cosine is $c'_j \cong c_j = \lambda$. From (49) and (52), the error output e_0^δ due to an error size δ is then approximated by

$$e_0^\delta(i,j) \cong -\lambda^{2p-i}\delta \tag{53}$$

for a faulty internal cell and

$$e_0^\delta(j,j) \cong \lambda^{2p-j}\eta_j \tag{54}$$

for a faulty boundary cell. Denote $B_\triangle$ be the wordlength of each memory and register of fixed point arithmetics. That is, each wordlength is of $B_\triangle$ bits and let $\triangle = \min(\delta, \eta_j)$. To ensure the detection of error size $\triangle$, we need

$$\lambda^{2p-i}\triangle \geq \lambda^{2p}\triangle \geq 2^{-B_\triangle}, \tag{55}$$

Therefore, the wordlength should be at least

$$B_\triangle \geq \lceil -2p\log_2\lambda - \log_2\triangle \rceil \tag{56}$$

such that the small error size $\triangle$ can be detected. The second term of the right-hand size is obvious since the error size $\triangle$ must be detected; the first term is to account for the effects that the error propagates through the array of LS order p with forgetting factor λ.

We can verify this by the following example. A systolic array with order $p = 3$, $\lambda = 0.999$ has an error $\delta = 3{\cdot}10^{-4}$ occurring at the internal cell PE_{12} at time 25. Due to the asymptotic behavior of the cosine parameters, η_j can be approximated as $\eta_j = \lambda \cdot \delta = 2.997 \cdot 10^{-4}$ and $\triangle = \eta_j$. From (56), we have $B_\triangle \geq 12$. Fig.6 shows that the small error size can be detected for $B_\triangle = 12$ at time 30. However, as shown in Fig.7 for smaller wordlength $B_\triangle = 5$, the error size that can be seen at the output becomes very small and is buried in the noise resulted from the quantization effect of small wordlength. The detector not only miss the error but also has the false alarm phenomenun that will be mentioned in the next Section.

6.2 False Alarm

Due to the finite-precision implementation, the residual output of the EDA will not be an actual zero if there is no fault in the system. we call this effect a false alarm. Fig.8 shows the false alarm problem for the above example with wordlength equals nine bits. Here, we are going to model and quantitatively describe the false alarm effect and introduce a threshold device to overcome this problem.

6.2.1 Cancellation Principle

Suppose now we have a fault-tolerant QRD RLS array of order $p = 3$. Denote the first and second rows of data input as $(x_1, x_2, x_3, x_1 + x_2 + x_3)$ and $(x'_1, x'_2, x'_3, x'_1 + x'_2 + x'_3)$ respectively, where the checksums $x_1 + x_2 + x_3$ and $x'_1 + x'_2 + x'_3$ are inputs to the EDA. After both data pass through the array, according to the operations of the processing cells, the contents of the cells of the first row are

$$\begin{aligned} r_{11} &= \sqrt{x_1^2 + x'^2_1}, \\ r_{12} &= sx'_2 + cx_2, \\ r_{13} &= sx'_3 + cx_3, \\ r_{14} &= s(x'_1 + x'_2 + x'_3) + c(x_1 + x_2 + x_3), \end{aligned} \tag{57}$$

where $c = x_1/r_{11}$ and $s = x'_1/r_{11}$ are the rotation parameters generated by the boundary cell and r_{ij} is the content of PE_{ij}. The output of the internal cells are

$$\begin{aligned} z_{12} &= cx'_2 - sx_2, \\ z_{13} &= cx'_3 - sx_3, \\ z_{14} &= c(x'_1 + x'_2 + x'_3) - s(x_1 + x_2 + x_3). \end{aligned} \tag{58}$$

Since $sx'_1 + cx_1 = \sqrt{x_1^2 + x'^2_1}$ and $cx'_1 - sx_1 = 0$, we have $r_{14} = r_{11} + r_{12} + r_{13}$ and $z_{14} = z_{12} + z_{13}$. That is, both the contents and the outputs of the first row still meet the checksum. The output of the first cell of EDA, z_{14}, can be rewritten as

$$z_{14} = c(x'_2 + x'_3) - s(x_2 + x_3). \tag{59}$$

We can see that the data from the first column got cancelled out by the first cell of the EDA. Since the outputs meet the checksum, with the same principle, the data from the

second column will get cancelled out by the 2^{nd} cell of the EDA. Thus, this observation can be generalized and stated as bellowed:

Cancellation Principle: With the checksum encoding data inputted to EDA, the data from the i^{th} column got cancelled out by the i^{th} cell of the EDA. □

For a finite-precision implementation, due to the roundoff error, the data from the i^{th} column will not be totally cancelled out by the i^{th} cell of the EDA. This effect results in the false alarm problem.

6.2.2 Finite-precision Floating Point Error Model

A floating point number f can be represented by [7]

$$f = \pm .d_1 d_2 \cdots d_t \times \beta^e \quad 0 \leq d_i < \beta, d_1 \neq 0, L \leq e \leq U, \tag{60}$$

where β is the base, t is the precision, and $[L, U]$ is the exponent range. The floating point operator fl can be shown to satisfy [7]

$$\begin{aligned} \hat{x} = fl(x) &= x(1+\epsilon) \\ fl(a\ op\ b) &= (a\ op\ b)(1+\epsilon) \quad |\epsilon| \leq \mathbf{u}, \end{aligned} \tag{61}$$

where $\mathbf{u}$ is the unit roundoff defined by

$$\mathbf{u} = \frac{1}{2}\beta^{1-t} \quad \text{for rounded arithmetics,}$$

and op denote any of the four arithmetic operations $+, -, \times, \div$.

6.2.3 Roundoff Analysis

For a QRD RLS systolic array of order p with finite-precision floating point arithmatics, denote the first row of input vector as $(\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_p, \sum_{i=1}^{p} \hat{x}_i + \epsilon_p)$, where $\hat{x}_i = fl(x_i)$, $\epsilon_p = \epsilon(\sum_{i=1}^{p} \hat{x}_i)$, and $|\epsilon| < \mathbf{u}$ is a constant[4], and the second row of input vector as $(\hat{x}'_1, \hat{x}'_2, \cdots, \hat{x}'_p, \sum_{i=1}^{p} \hat{x}'_i + \epsilon_p)$. The content of the first boundary cell is given by

$$\hat{r}_{11} = fl(\sqrt{\hat{x}_1^2 + \hat{x'}_1^2}) = \sqrt{\hat{x}_1^2 + \hat{x'}_1^2}(1+\epsilon), \tag{62}$$

[4]To simplify the notation, we do not give indexs to different ϵ's.

and the rotation parameters are $\hat{c} = fl(\hat{x}_1/\hat{r}_{11})$ and $\hat{s} = fl(\hat{x}'_1/\hat{r}_{11})$. The contents of the internal cells can then be obtained as

$$\begin{aligned} \hat{r}_{ij} &= fl(fl(\hat{s}\hat{x}'_j) + fl(\hat{c}\hat{x}_j)) \\ &= [\hat{s}\hat{x}'_j(1+\epsilon) + \hat{c}\hat{x}_j(1+\epsilon)](1+\epsilon) \\ &\approx (1+2\epsilon)(\hat{s}\hat{x}'_j + \hat{c}\hat{x}_j), \quad 1 < j \leq p \end{aligned} \tag{63}$$

and the content of the first cell of the EDA is

$$\begin{aligned} \hat{r}_{1,p+1} &= fl(fl(\hat{s}(\sum_{i=1}^{p} \hat{x}'_i + \epsilon_p)) + fl(\hat{c}(\sum_{i=1}^{p} \hat{x}_i + \epsilon_p))) \\ &\approx (\hat{s}\sum_{i=1}^{p} \hat{x}'_i + \hat{c}\sum_{i=1}^{p} \hat{x}_i) + 6\epsilon_p. \end{aligned} \tag{64}$$

From (62), (63), and (64), the mismatch τ_1 resulted from the finite precision computation of the first row is

$$\tau_1 = 6\epsilon_p - (\epsilon\sqrt{\hat{x}_1^2 + \hat{x'}_1^2} + 2\epsilon\sum_{i=2}^{p}(\hat{s}\hat{x}'_j + \hat{c}\hat{x}_j)) \tag{65}$$

and it can be bounded by

$$\begin{aligned} |\tau_1| &\leq 6p|\epsilon x_{max}| + |2\epsilon x_{max}| + 4(p-1)|\epsilon x_{max}| \\ &= (10p-2)|\epsilon x_{max}| \leq 10p|\epsilon x_{max}|. \end{aligned} \tag{66}$$

For the second row, with the same principle, the mismatch is bounded by $10(p-1)|\epsilon x_{max}|$. The total mismatch from the whole array is given by

$$|\tau| \leq \sum_{i=0}^{p-1} 10(p-i)|\epsilon x_{max}| = 5p(p+1)|\epsilon x_{max}|. \tag{67}$$

The possible mismatch is thus bounded by

$$|\tau| \leq 5p(p+1)|\epsilon x_{max}|. \tag{68}$$

This bound can be interpreted as: For each row of input, each processing cell contributes about $|\epsilon x_{max}|$ amount of roundoff error. Since there are about $p(p+1)$ processing cells, the total possible roundoff error is then $p(p+1)|\epsilon x_{max}|$.

In order to prevent the false alarm, a threshold device is needed at the output of ϵ_0 and the threshold has to set at least $|\tau|$. Suppose $\beta = 2, t = 16$, then $\mathbf{u} = 2^{-16}$. Given an scaled input data such that $|x_{max}| = 1$, the threshold of a QRD RLS array of order $p = 20$ is

$$th \geq |\tau|_{max} \simeq 5 \cdot 20 \cdot 21 \cdot 2^{-16} = 0.032. \tag{69}$$

Table 4 shows the comparisons of the maximum values of the output residuals e_0 obtained from a period of $n = 10^4$ and the threshold th derived from (68). We can see that the estimated threshold can prevent the false alarm problem. Since the threshold is obtained from a conservative derivations, it can always provide a false alarm free output. However, as shown in Table 4, the estimated threshold may be much higher than that of the actual maximum of the residuals. We can relax the estimated threshold from information obtained in previous data to ensure the threshold will not be too high. A high threshold usually means a small error size may not be able to be detected.

6.3 Overall Wordlength Consideration

To prevent missing error detection, we want to detect the error size $\triangle = \min(\delta, \eta_j)$ as small as possible. While to prevent the false alarm, we also want to choose a threshold high enough for a false alarm free condition. Both situations cannot be satisfied simultaneously since they are of tradeoff in nature.

To detect the error size $\triangle$, from (53), (54), and (55), we need the threshold $th \leq \lambda^{2p}\triangle$. Otherwise, the propagated error will be eventually truncated to zero by the threshold device. Accordingly,

$$B_\triangle \leq \lceil -\log_2 th \rceil, \tag{70}$$

since a smaller error size is unable to be detected. From (56), a criterion to choose $B_\triangle$ is then given by

$$B_\triangle = \min(\lceil -2p\log_2 \lambda - \log_2 \triangle \rceil, \lceil -\log_2 th \rceil). \tag{71}$$

If $B_\triangle = \lceil -\log_2 th \rceil$, the minimal detectable error size is $\triangle = \lambda^{-2p} \cdot th$. For a threshold set at $th = 10^{-4}$ as given in (69) and a LS order $p = 50$ and $\lambda = 0.98$, we have $\triangle = 7.54 \cdot 10^{-4}$. However, for a smaller LS order p, a smaller error size can be detected. For example, with $p = 20$, we have $\triangle = 1.5 \cdot 10^{-4}$. To prevent overflow, from (27), the minimum wordlength of the m^{th} row is

$$B_m = \lceil (m-1)(1 + \log_2 \lambda) + \log_2 \Re \rceil. \tag{72}$$

For a QRD RLS systolic array to detect small error size $\triangle$ without false alarm and overflow problems, the minimum wordlength of the m^{th} row should be

$$B_{min}(m) = \max(B_m, B_\triangle). \tag{73}$$

7 Conclusions

We present an important observation that the rotation parameters of the RLS algorithm based on Givens rotation method will eventually reach the *quasi steady-state* if the forgetting factor λ is very close to 1. With this model, the dynamic range of each processing cell can be derived and from this, a proper wordlength can be chosen to ensure correct operations of the algorithm. Our proposed solutions are simple and effective. Our simulations have demonstrated that the wordlengths chosen by the proposed dynamic range work very well. Also, we can prove the stability of the QRD RLS algorithm under finite-precision implementation with this observation. Finally, the missing error detection and false alarm problems are considered based on the results obtained from the model. We present a design of the wordlength which is overflow free without missing error detection and false alarm problems.

The results in this paper is of practical importance. Not only can we design a finite-precision QRD RLS systolic array with a minimum wordlength that ensures correct oooperations, but also provide a fault-tolerant system that can detect a given error size and is false alarm free under the quantization effect.

References

[1] M.G. Bellanger, "Computational complexity and accuracy issues in fast least squares algorithms for adaptive filtering", Proc. IEEE ISCAS, pp.2635-2639, Finland, 1988.

[2] C.-Y. Chen and J.A. Abraham, "Fault-tolerant systems for the computation of eigenvalues and singular values", Proc. SPIE, Vol 696, Advanced Algorithms and Architectures for Signal Processing, pp.228-237, 1986.

[3] M.J. Chen, "On realizations and performances of least-squares estimation and Kalman filterring by systolic array", Ph.D. dissertation, Electrical Engineering Dept., UCLA, 1987.

[4] J.M. Cioffi, "The fast adaptive ROTOR's RLS algorithm", IEEE Trans. Acoustics, Speech, and Signal Processing, pp.631-653, April 1990.

[5] G.D. de Villiers, "A Gentleman-Kung architecture for finding the singular value of a matrix", Proc. Int'l Conf. Systolic Array, pp.545-554, Ireland, 1989.

[6] W.M. Gentleman and H.T. Kung, "Matrix triangularization by systolic arrays", Proc. SPIE, Vol 298, Real Time Signal Processing IV, pp.298, 1981.

[7] G.H. Golub and C.F. Van Loan, **Matrix Computation,** 2*nd* edition, Johns Hopkins, 1989.

[8] S. Haykin, **Adaptive Filter Theory**, Prentice Hall, 1986.

[9] S.F. Hsieh and K. Yao, "Hyperbolic Gram-Schmidt pseudo-orthogonalization with applications to sliding window RLS filtering", 24-th Annual Conference on Information Science and System, Princeton University, Mar. 21-3, 1990.

[10] S.F. Hsieh and K. Yao, "Systolic implementation of windowed recursive LS estimation", Proc. IEEE ISCAS, pp.1931-1934, New Orleans, May 1990.

[11] J.-Y Jou and J.A. Abraham, "Fault-tolerant matrix arithmetic and signal processing on highly concurrent computing structures", Proc. IEEE, Vol 74, pp.732-741, May, 1986.

[12] S. Kalson and K. Yao, "Systolic array processing for order and time recursive generalized least-squares estimation," Proc. SPIE, Vol. 564, Real Time Signal Processing VIII, pp. 28-38, 1985.

[13] H. Leung and S. Haykin, "Stability of recursive QRD LS algorithms using finite-precision systolic array implementation", IEEE Trans. ASSP, VOL37 pp.760-763, May 1989.

[14] F. Ling, D. Manolakis, and J.G. Proakis, "A recursive modified Gram-Schmidt algorithm for least-squares estimation", IEEE Trans. ASSP, Vol. ASSP-34, pp.829-836, Aug. 1986.

[15] K.J.R. Liu and K. Yao, "Gracefully degradable real-time algorithm-based fault-tolerant method for QR recursive least-squares systolic array", in Systolic Array Processors, Ed. McCanny, McWhirter, and Swartzlander, pp. 401-410, Prentice Hall (UK), 1989.

[16] K.J.R. Liu, S.F. Hsieh, and K. Yao, "Two-level pipelined implementation of systolic block Householder transformations with application to RLS algorithm", Proc. Int'l Conf. on Application-Specific Array Processors, pp.758-769, Princeton, Sep. 1990.

[17] K.J.R. Liu and K. Yao, "Spectral decomposition via systolic triarray based on QR iteration", Proc. IEEE ICASSP, pp.1017-1020, Albuquerque, April 1990.

[18] V.J. Mathews and Z. Xie, "Fixed-point error analysis of stochastic gradient adaptive lattice filters", IEEE Trans. ASSP, Vol 38, pp.70-80, Jan. 1990.

[19] J.V. McCanny and J.G. McWhitter, "Some systolic array developments in the United Kingdom", IEEE Computer, Vol 20, pp.51-64, July 1987.

[20] J.G. McWhirter, "Recursive least-squares minimization using a systolic array", Proc. SPIE, Vol 431, Real Time Signal Processing VI, pp.105-112, 1983.

[21] J. G. McWhirter and T. J. Shepherd, "Systolic array processor for MVDR beamforming," IEE Proceedings, Vol. 136, Pt. F, No. 2, pp. 75-80, 1989.

[22] H. Tsubokawa, H. Kubota, and S. Tsujii, "Effect of floating-point error reduction with recursive least square for parallel architecture", Proc. IEEE ICASSP, pp.1487-1490, Albuquerque, April 1990.

[23] Viterbi and Omura, **Principle of Digital Communication and Coding**, McGraw-Hill, 1979.

[24] B. Yang and J.F. Bohme, "Systolic implementation of a general adaptive array processing algorithm", IEEE ICASSP, pp.2785-2788, New York, 1988.

[25] J. H. Wilkinson, **The Algebraic Eigenvalue Problem.** Oxford, 1965.

	AR1	AR2	AR3	AR4	WN
λ= 980	.9800	.9800	.9802	.9799	.9801
λ=.985	.9849	.9849	.9851	.9848	.9850
λ=.990	.9897	.9897	.9900	.9897	.9899
λ=.991	.9907	.9907	.9910	.9907	.9909
λ=.993	.9927	.9927	.9930	.9927	.9929
λ=.995	.9947	.9947	.9950	.9947	.9949
λ=.997	.9967	.9967	.9970	.9967	.9969
λ=.999	.9985	.9985	.9987	.9985	.9986

Table 1 Mean distributation of cosine parameters for different input signals.

	AR1	AR2	AR3	AR4	WN
λ= 980	7.3885e-4	7.5465e-4	6.8163e-4	6.6721e-4	7.3367e-4
λ=.985	4.3970e-4	4.5144e-4	3.9577e-4	3.9517e-4	4.3308e-4
λ=.990	2.0903e-4	2.1463e-4	1.8376e-4	1.8918e-4	2.0080e-4
λ=.991	1.7154e-4	1.7875e-4	1.4883e-4	1.5562e-4	1.6659e-4
λ=.993	1.0991e-4	1.1390e-4	9.1016e-5	9.6440e-5	1.0323e-4
λ=.995	5.9724e-5	6.0796e-5	4.6789e-5	5.1856e-5	5.3525e-5
λ=.997	2.3007e-5	2.4735e-5	1.6808e-5	1.9908e-5	2.0504e-5
λ=.999	4.1127e-6	3.1590e-6	3.5167e-6	4.3511e-6	4.6490e-6

Table 2 Variance distributation of δ for different input signals.

	AR1	AR2	AR3	AR4
$\Re$	47.5737	16.6493	6.8209	17.1317
Max r_{ii}	12.1135	5.6755	2.5770	6.3590
Max r_{ij}	5.4948	3.3982	0.9036	4.2805

Table 3 Comparisons of the upper bound $\Re$ and the maximum values of the contents of the boundary and internal cells.

Wordlength	6	7	9	12	16	20	24
Max e_0	2.114e-3	2.12e-4	3.41e-5	2.011e-9	6.74e-13	5.696e-13	4.5856e-13
Threshold	9.375e-1	4.69e-1	1.172e-1	1.465e-2	9.1e-4	5.722e-5	3.58e-6

Table 4 Comparisons of the thresholds and the maximum values of e_0.

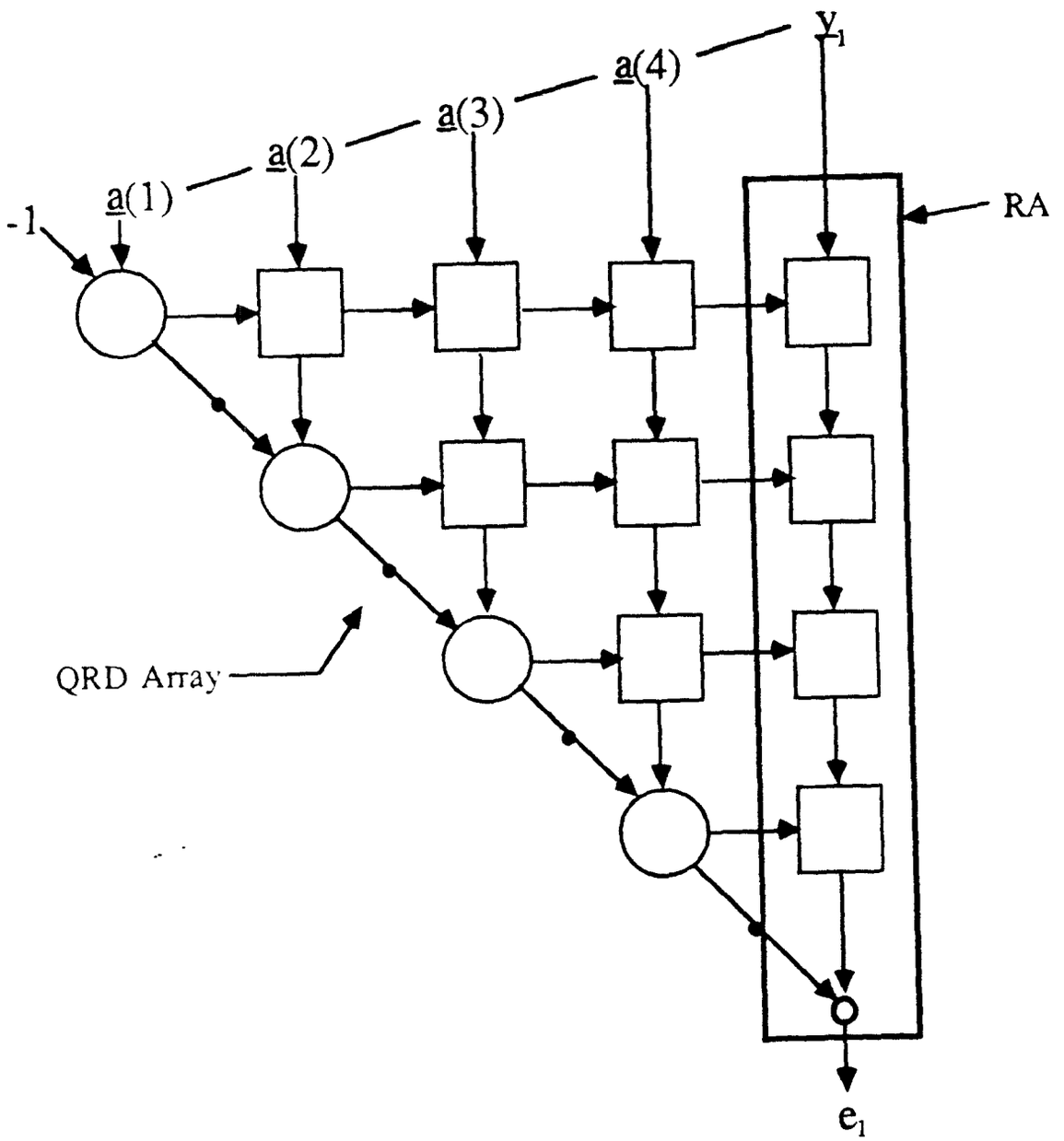
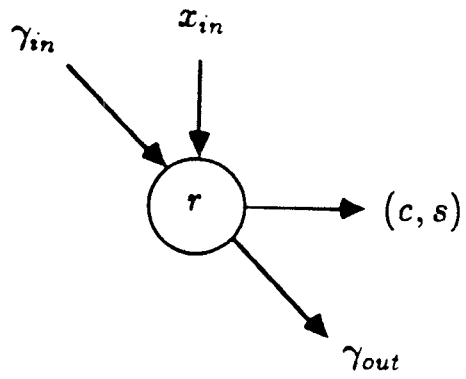


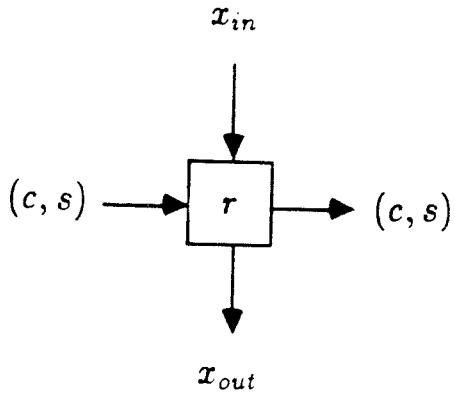
Fig.1a QRD RLS systolic array using Givens rotation method.

(1) Boundary Cell



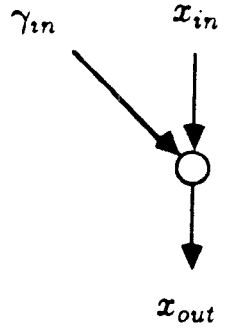
If $x_{in} = 0$ **then**
$c \leftarrow 1;\ s \leftarrow 0;\ \gamma_{out} \leftarrow \gamma_{in};$
$r = \lambda r,$
otherwise
$r' = \sqrt{\lambda^2 r^2 + x_{in}^2};$
$c \leftarrow \lambda r / r';\ s \leftarrow x_{in}/r'$
$r \leftarrow r';\ \gamma_{out} = c\gamma_{in}$
end

(2) Internal Cell



$$x_{out} \leftarrow cx_{in} - s\lambda r$$
$$r \leftarrow sx_{in} + c\lambda r$$

(3) Final Cell



$$x_{out} \leftarrow \gamma_{in} x_{in}$$

Fig.1b Processing cells of the Givens rotation method.

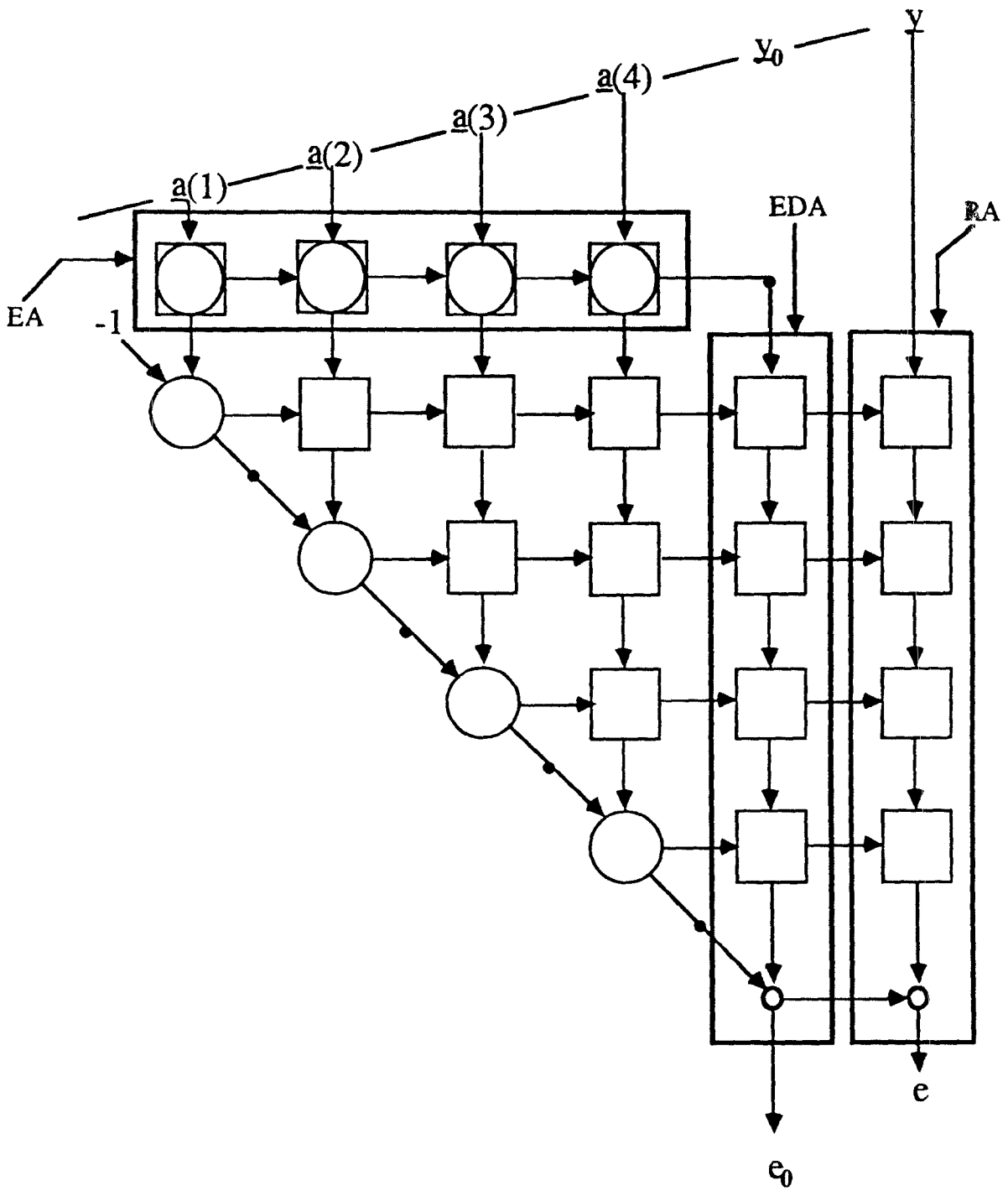


Fig.2 Fault-tolerant QRD RLS systolic array.

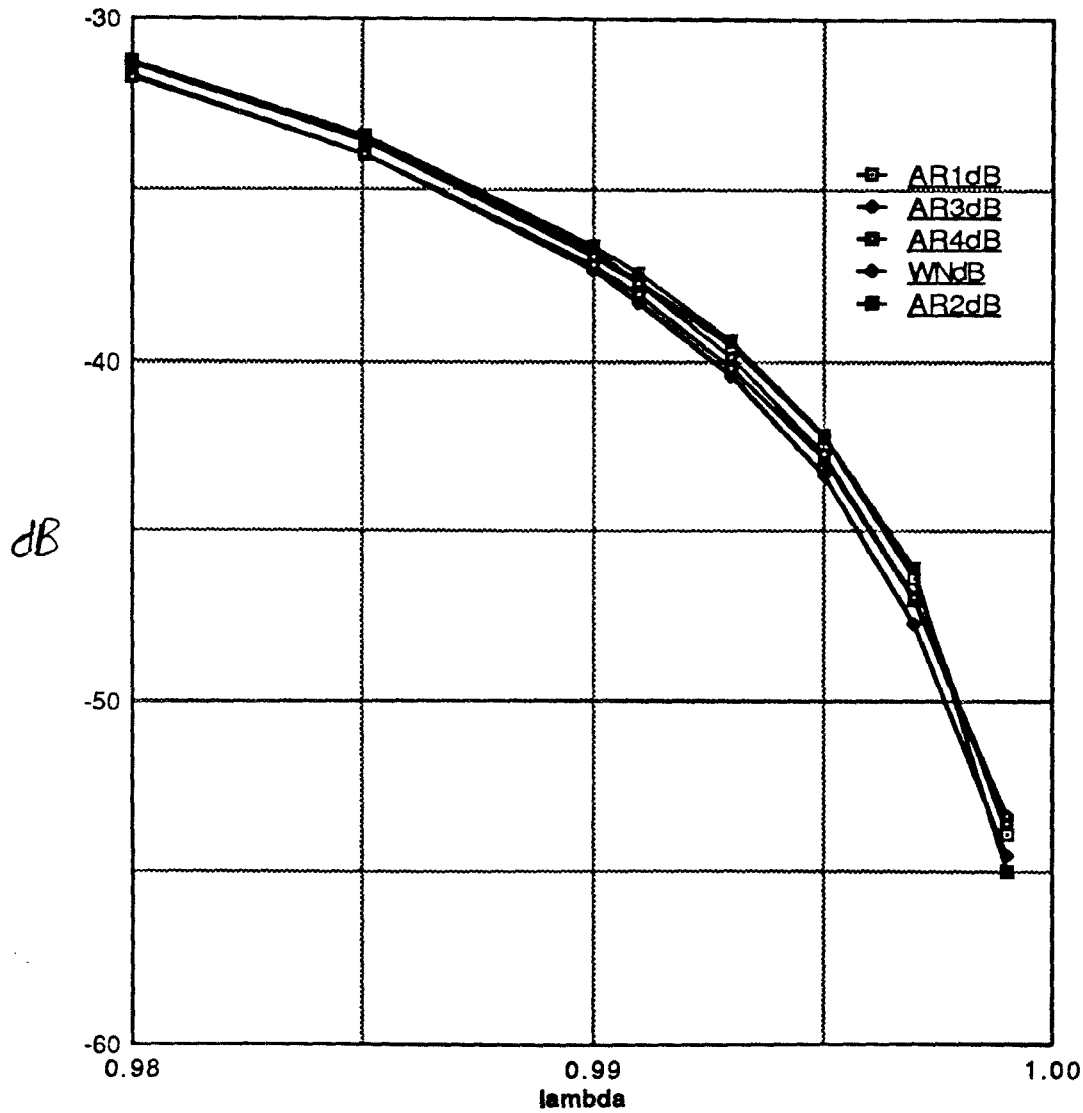


Fig.3 Plots of variances in dB scale.

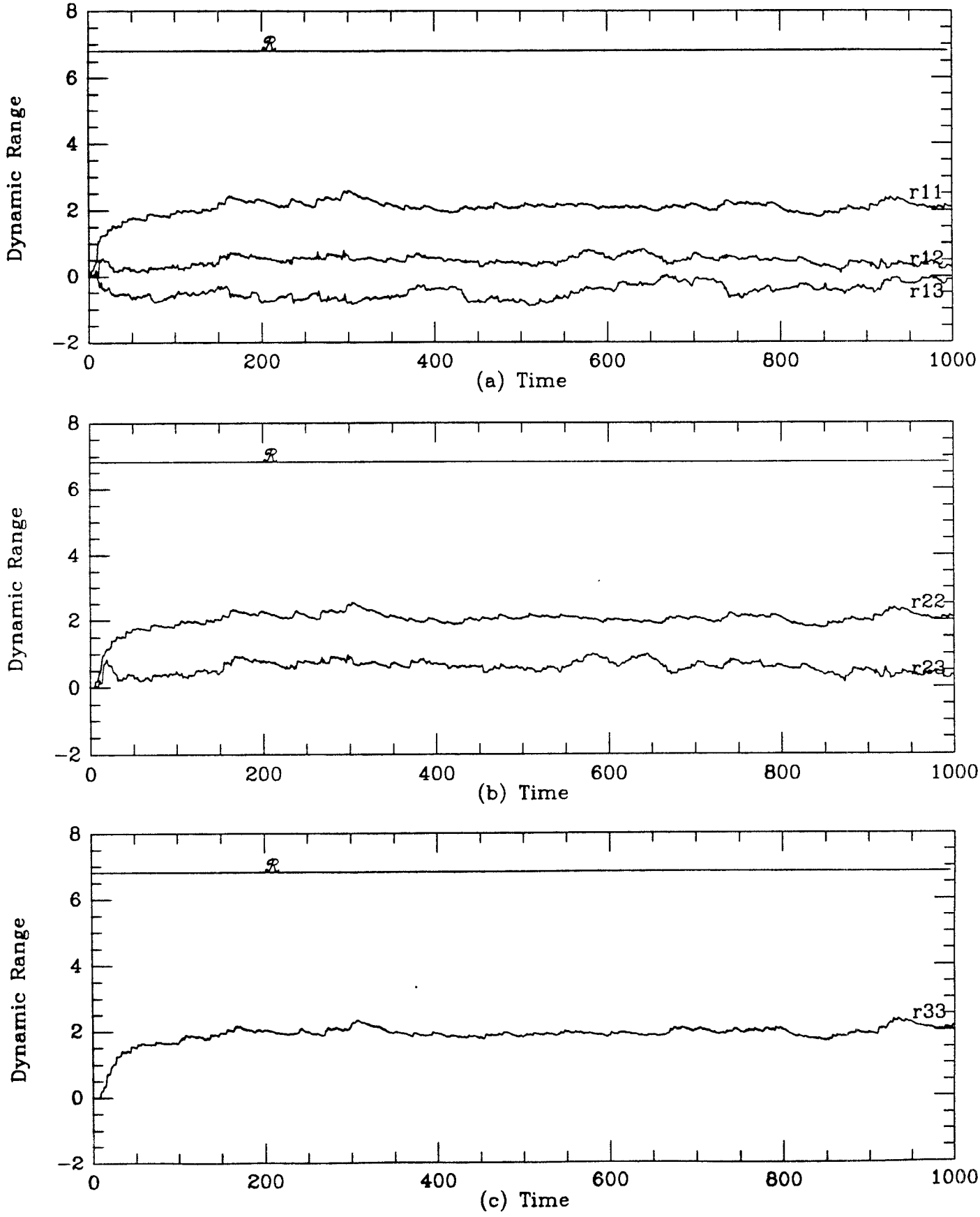


Fig.4 Plots of the contents of processing cells with AR3 signal for $\lambda = 0.911$ and $p = 3$: (a) The first row, (b) The second row, (c) The third row.

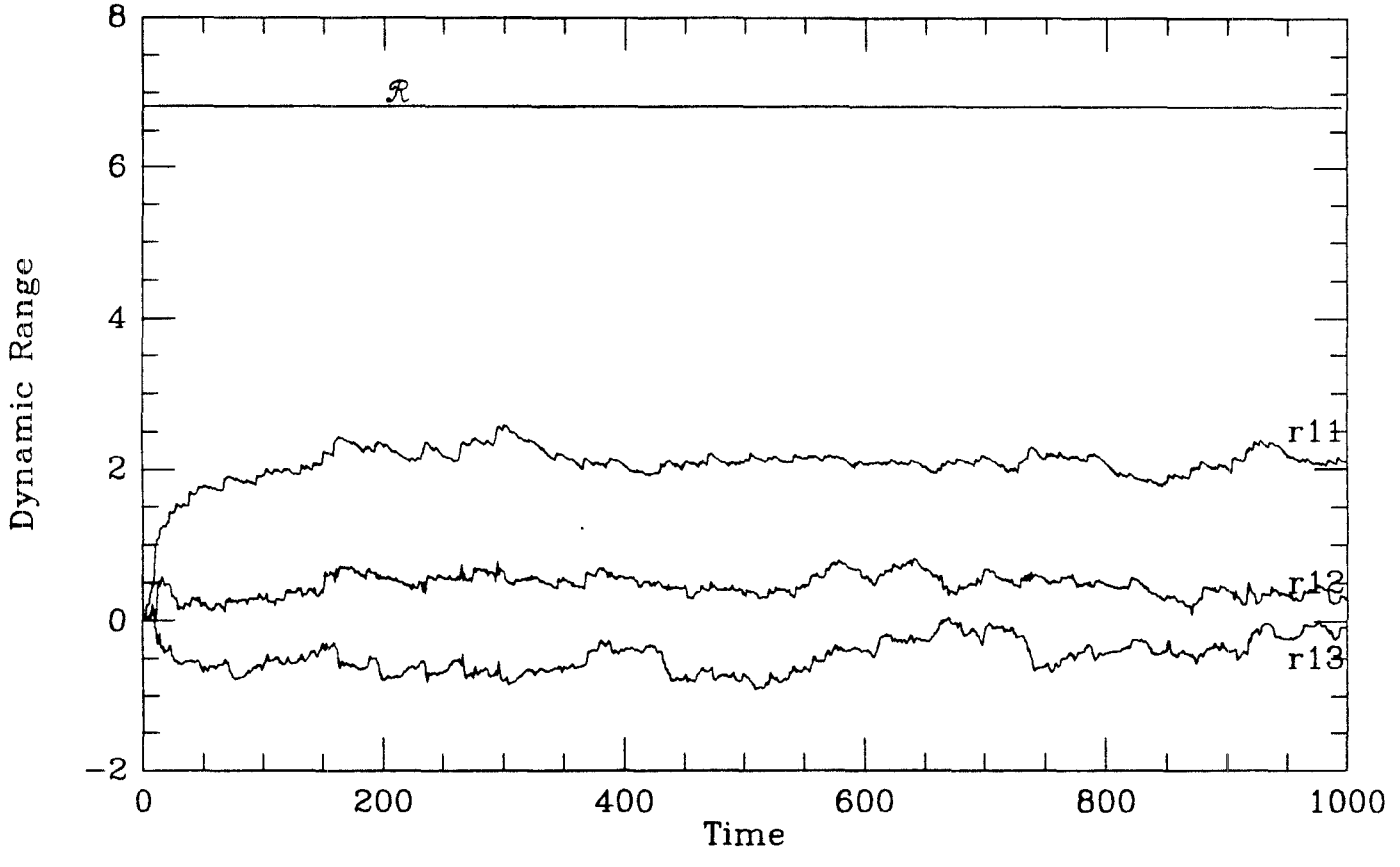


Fig.5 Plots of the contents of the first row processing cells with finite wordlengths: 3 bits (row 1), 4 bits (row 2), 5 bits (row3), and 4 bits for others.

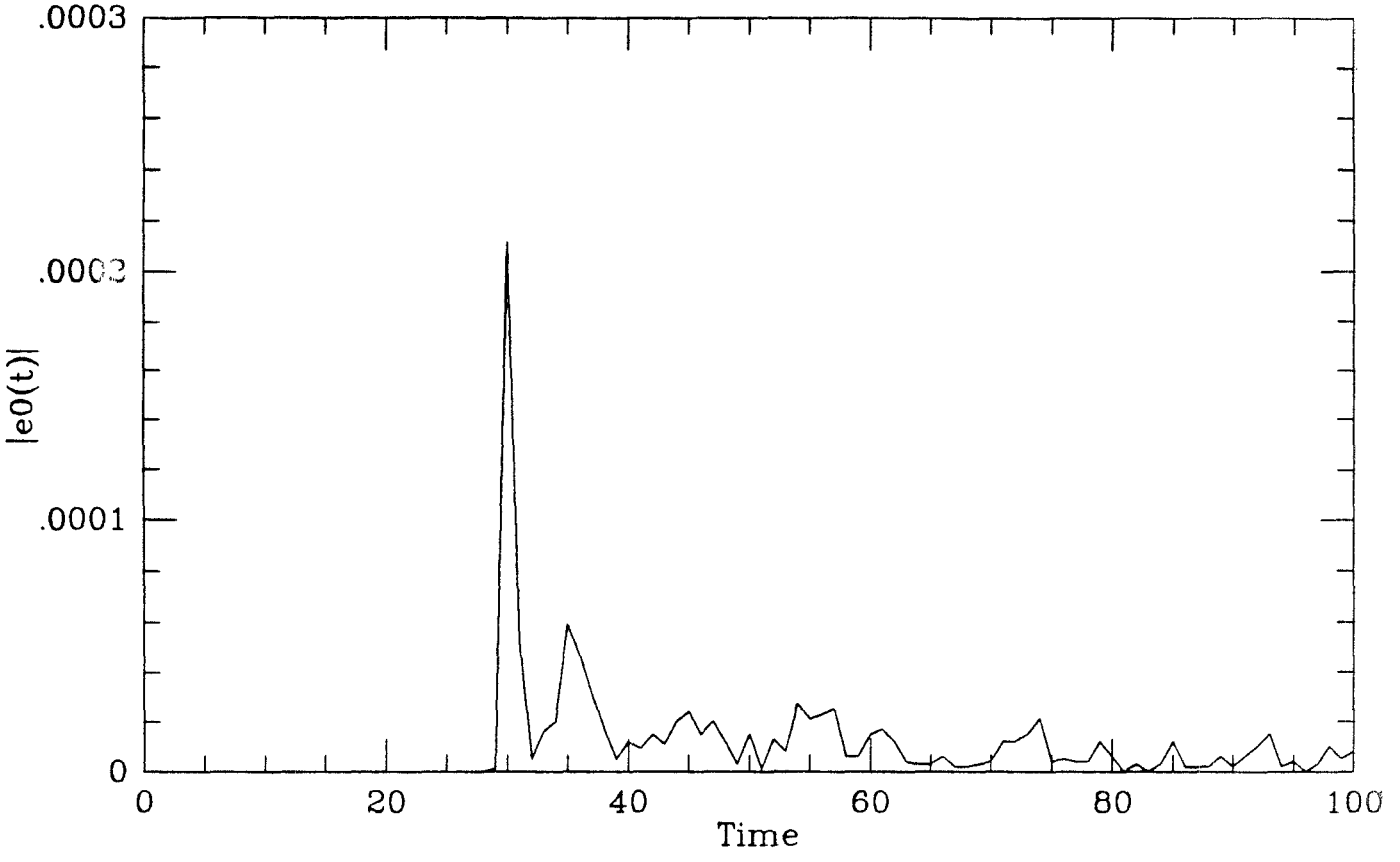


Fig.6 The error size $\delta = 3 \cdot 10^{-4}$ occurring at PE_{12} can be detected for $B_\Delta = 12$.

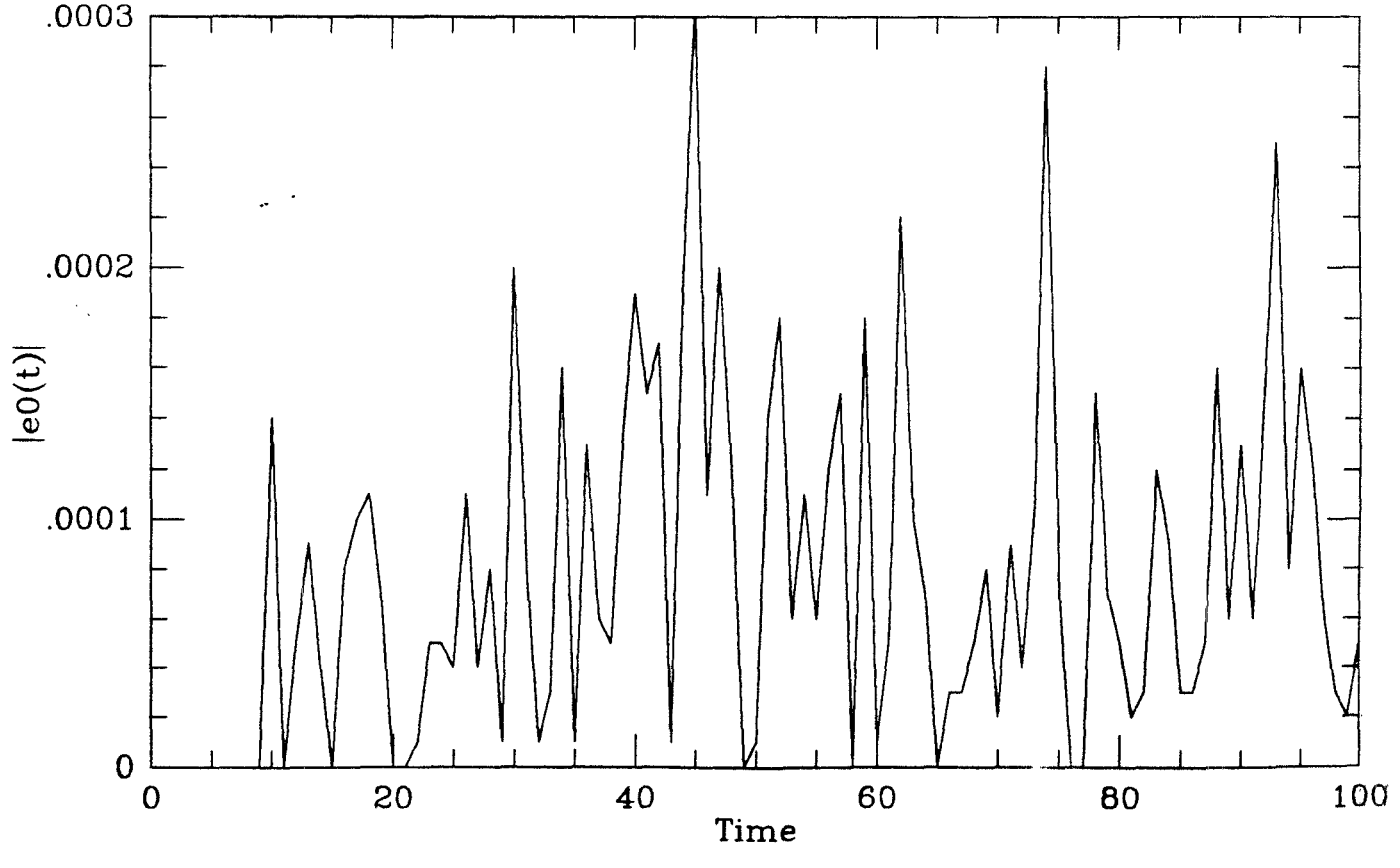


Fig.7 The error size $\delta = 3 \cdot 10^{-4}$ occurring at PE_{12} cannot be detected for $B_\Delta = 5$.

It is too small and is buried in the noise resulted from quantization.

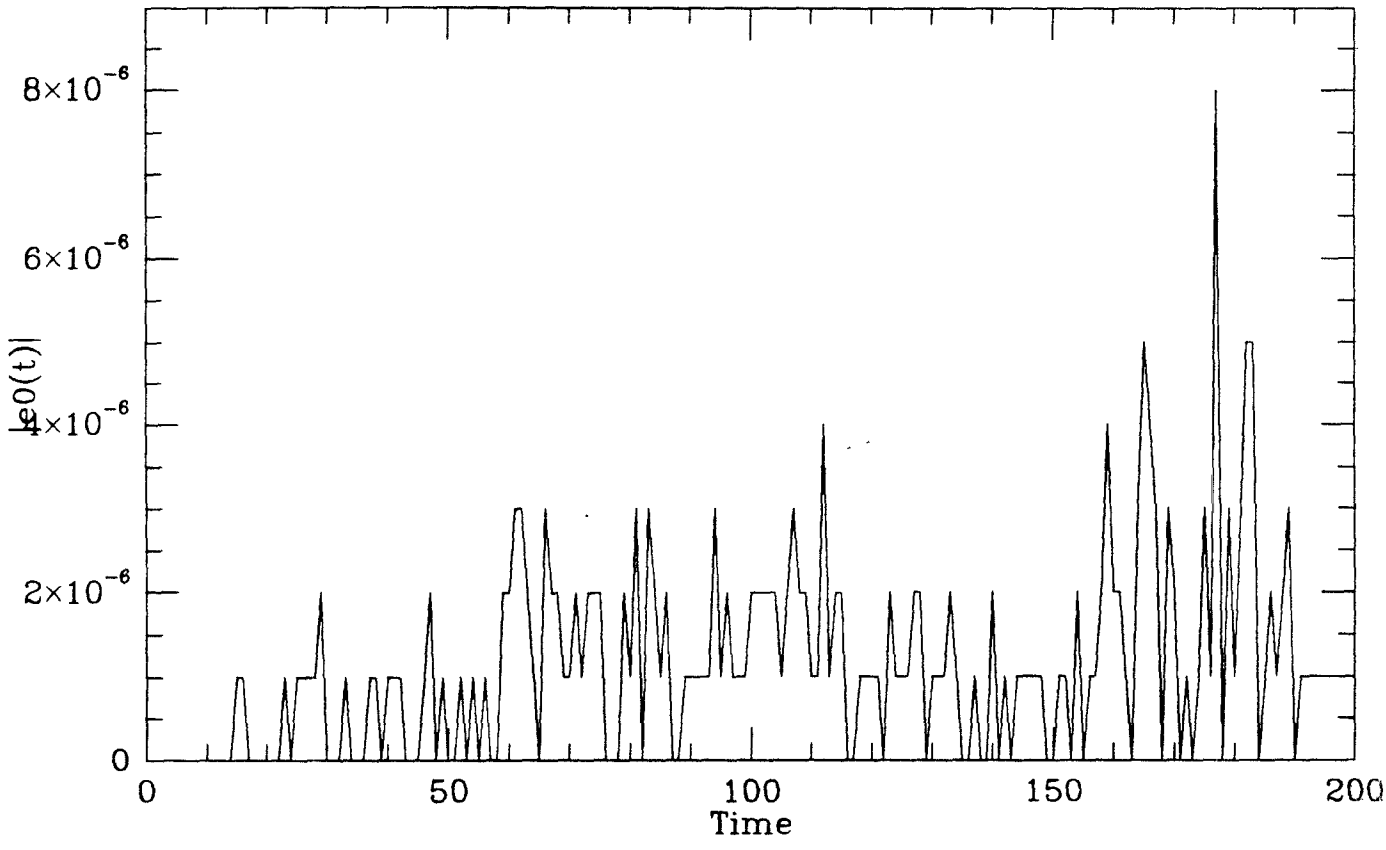


Fig.8 False alarm effect for finite wordlength with 9-bit implementation.

List of Tables and Figures