transforms for improving performance of transform domain normalized LMS algorithm," *Proc. Inst. Elec. Eng.*, pt. F, vol. 139, pp. 327-335, Oct. 1992.
[6] D. L. Duttweiler, "Adaptive filter performance with nonlinearities in the correlation multipliers," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, pp. 578-586, Aug. 1982.
[7] E. Eweda, "Analysis and design of a signed regressor LMS algorithm for stationary and nonstationary adaptive filtering with correlated Gaussian data," *IEEE Trans. Circuits Syst.*, vol. 37, pp. 1367-1374, Nov. 1990.
[8] R. Price, "A useful theorem for nonlinear devices having Gaussian inputs," *IRE Trans. Inform. Theory*, vol. IT-4, pp. 69-72, June 1958.
[9] S. S. Reddi, "A time-domain adaptive algorithm for rapid convergence," *Proc. IEEE*, vol. 72, pp. 533-535, Apr. 1984.
[10] B. Farhang-Boroujeny, "An efficient quasi-LMS/Newton algorithm: Analysis and simulation results," Tech. Rep. 01-10-91, Commun. Division, Dept. Elec. Eng., National Univ. Singapore, Oct. 1991.
[11] D. F. Marshall and W. K. Jenkins, "A fast quasi-Newton adaptive filtering algorithm," in *Proc. 1988 ICASSP* (New York, NY), Apr. 11-14, pp. 1377-1380.
[12] D. F. Marshall and W. K. Jenkins, "A fast quasi-Newton adaptive filtering algorithm," *IEEE Trans. Signal Processing*, vol. 40, pp. 1652-1662, July 1992.

A Unified Square-Root-Free Approach for QRD-Based Recursive Least Squares Estimation

S. F. Hsieh, K. J. R. Liu, and K. Yao

Abstract—**Givens rotation is the most commonly used method in performing the *QR* decomposition (QRD) updating. The generic formula for these rotations requires explicit square-root (sqrt) computations which constitute a computational bottleneck and are quite undesirable from the practical VLSI circuit design point of view. So far, there has been more than ten known sqrt-free algorithms. In this correspondence, we provide a unified systematic approach for the sqrt-free Givens rotation. By properly choosing two parameters, μ and ν, all previously known sqrt-free, as well as new methods, are included in our unified approach. This unified treatment is also extended to the QRD-based recursive least squares (RLS) problem for optimum residual acquisition without sqrt operations.**

I. Introduction

The Givens rotation, which requires a square-root (sqrt) operation in the generic formulation, is a versatile method in performing many signal processing algorithms involving matrix computations, such as the *QR* decomposition (QRD), the singular value decomposition, and the eigendecomposition [7]. While many researchers have worked on reformulating algorithms suitable for parallel computing and VLSI architectures, current VLSI architectures still disapprove if not prohibit sophisticated computations. A noticeable example is the sqrt operation, which may occupy much area in a VLSI chip or may also require many cycles to accomplish such computation. In addition, a recent simulation study presented in [14] by Proudler *et al.* showed that a finite-precision implementation of a sqrt-free lattice algorithm achieved better numerical results than that using the conventional Givens rotation method.

Manuscript received July 10, 1991; revised March 31, 1992. This work was supported in part by the National Science Council of the Republic of China under Grant NSC80-E-SP-009-01A, the NSF Engineering Center Grant ECD-8803012 and Minta Martin Award of the University of Maryland, the NSF Grant NCR-8814407, and a UC MICRO grant.

S. F. Hsieh is with the Department of Communication Engineering, National Chiao Tung University, Hsinchu, Taiwan 30039, Republic of China.
K. J. R. Liu is with the Department of Electrical Engineering, Systems Research Center, University of Maryland, College Park, MD 20742.
K. Yao is with the Department of Electrical Engineering, University of California, Los Angeles, CA 90024-1594.

IEEE Log Number 9206004.

Thus, much effort has been spent on minimizing or even eliminating the sqrt operation from these algorithms. One well-known example is the sqrt-free Givens rotation first proposed by Gentleman [5]. Hammarling generalized his results briefly [9]. Later, other versions of the sqrt-free Givens rotations were also proposed [1], [2], [8]. All of the above algorithms only focus on the sqrt-free Givens rotation itself and/or its applications in solving a least squares (LS) problem. McWhirter [13] was the first to apply the sqrt-free Givens rotation to recursive LS (RLS) problems in computing the desired optimum residual without solving explicitly for the LS coefficients. Closely related to the Givens rotation method is the modified Gram-Schmidt (MGS) orthogonalization, which is another approach in performing the QRD. Ling *et al.* [11], [12] and Kalson and Yao [10] independently developed the sqrt-free MGS methods for the RLS filtering problems. A rank-one updating of Cholesky factorization without sqrt's has also been reported in the literature [3]. Recently, Chen and Yao [4] summarized the works done on the sqrt-free RLS filtering and proposed another more efficient sqrt-free method. So far, there has been more than ten known sqrt-free algorithms [1], [2], [4], [5], [8]-[11]. However, all of the previously known derivations were based on heuristic approaches. There is no known systematic way of generating the sqrt-free algorithms. Motivated by these works, we wish to understand the fundamental relationships among these sqrt-free algorithms. One of the contributions of this correspondence is that these fundamental relationships are characterized in simple manners through only two parameters.

The prototypes of generalized sqrt-free algorithms are given in Section II, where all of the sqrt-free algorithms are found by the selection of two parameters. We proceed in Section III to seek a sqrt-free optimum residual of the RLS filtering problem. A brief conclusions is given in Section IV.

II. The $\mu\nu$ Family of Square-Root-Free Algorithms

A Givens rotation matrix as given by

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix}$$

is used to premultiply a two-row matrix

$$\begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_p \\ \beta_1 & \beta_2 & \cdots & \beta_p \end{bmatrix}$$

to zero out the element at the (2, 1) location such that it becomes

$$\begin{bmatrix} \alpha_1' & \alpha_2' & \cdots & \alpha_p' \\ 0 & \beta_2' & \cdots & \beta_p' \end{bmatrix}$$

where

$$c = \alpha_1/\sqrt{\alpha_1^2 + \beta_1^2}, \quad \text{and } s = \beta_1/\sqrt{\alpha_1^2 + \beta_1^2} \tag{1}$$

$$\alpha_1' = \sqrt{\alpha_1^2 + \beta_1^2} \tag{2}$$

$$\alpha_j' = c\alpha_j + s\beta_j, \quad j = 2, \cdots, p. \tag{3}$$

$$\beta_j' = -s\alpha_j + c\beta_j, \tag{4}$$

In VLSI circuit design, sqrt operation is expensive, because it takes up much area or is slow (due to many iterations). Therefore, it is advantageous to avoid or minimize sqrt operations.

By taking out a scaling factor from each row, the two rows under consideration before and after the Givens orthogonal transformations is denoted by

$$\begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_p \\ \beta_1 & \beta_2 & \cdots & \beta_p \end{bmatrix} = \begin{bmatrix} \sqrt{k_a} & 0 \\ 0 & \sqrt{k_b} \end{bmatrix} \begin{bmatrix} a_1 & a_2 & \cdots & a_p \\ b_2 & b_2 & \cdots & b_p \end{bmatrix} \tag{5}$$

and

$$\begin{bmatrix} \alpha_1' & \alpha_2' & \cdots & \alpha_p' \\ 0 & \beta_2' & \cdots & \beta_p' \end{bmatrix} = \begin{bmatrix} \sqrt{k_a'} & 0 \\ 0 & \sqrt{k_b'} \end{bmatrix} \begin{bmatrix} a_1' & a_2' & \cdots & a_p' \\ 0 & b_2' & \cdots & b_p' \end{bmatrix} \tag{6}$$

where k_a, k_b, k_a', and k_b' are the scaling factors resulting in sqrt-free operations, and α_i' and β_i' are the updated α_i and β_i when β_1 is zeroed out.

Now, our task is to find the expressions for k_a', k_b', a_1', $\{(a_j', b_j'), j = 2, \cdots, p\}$, in terms of k_a, k_b, $\{(a_j, b_j), j = 1, \cdots, p\}$, such that *no* sqrt operation is actually needed. The sqrt expressions of $\sqrt{k_a}$, $\sqrt{k_b}$, $\sqrt{k_a'}$, and $\sqrt{k_b'}$ in (5) and (6) are used for representational purposes only and are not actually performed.

Replacing $\alpha_j = \sqrt{k_a} a_j$, $\beta_j = \sqrt{k_b} b_j$, $\alpha_j' = \sqrt{k_a'} a_j'$, $\beta_j' = \sqrt{k_b'} b_j'$, $j = 1, \cdots, p$, in (1)–(4) leads to

$$c = \frac{\sqrt{k_a} a_1}{\sqrt{k_a a_1^2 + k_b b_1^2}}, \quad \text{and} \quad s = \frac{\sqrt{k_b} b_1}{\sqrt{k_a a_1^2 + k_b b_1^2}} \tag{7}$$

$$a_1' = \sqrt{(k_a a_1^2 + k_b b_1^2)/k_a'} \tag{8}$$

$$a_j' = \frac{1}{\sqrt{k_a'}\sqrt{k_a a_1^2 + k_b b_1^2}} [k_a a_1 a_j + k_b b_1 b_j] \qquad j = 2, \cdots, p. \tag{9}$$

$$b_j' = \frac{\sqrt{k_a k_b}}{\sqrt{k_b'}\sqrt{k_a a_1^2 + k_b b_1^2}} [-b_1 a_j + a_1 b_j], \tag{10}$$

To avoid sqrt computation, we need to determine k_a' and k_b' such that a_1', a_j', and b_j' will not require sqrt operation. It is clear that if we choose k_a' and k_b' as

$$k_a' = \frac{k_a a_1^2 + k_b b_1^2}{\mu^2} \tag{11}$$

$$k_b' = \frac{k_a k_b}{\nu^2(k_a a_1^2 + k_b b_1^2)} \tag{12}$$

where μ and ν are parameters that will be determined later to be any sqrt-free function of k_a, k_b, a_1, and b_1, then (8)–(10) can be computed without sqrt operation. We then have the following updating formulas without sqrt operation:

$$k_a' = (k_a a_1^2 + k_b b_1^2)/\mu^2 \tag{13}$$

$$k_b' = \frac{k_a k_b}{\nu^2(k_a a_1^2 + k_b b_1^2)} = \frac{k_a k_b}{\mu^2 \nu^2 k_a'} \tag{14}$$

$$a_1' = \mu \tag{15}$$

$$a_j' = \frac{\mu}{k_a a_1^2 + k_b b_1^2} [k_a a_1 a_j + k_b b_1 b_j] = \frac{k_a a_1 a_j + k_b b_1 b_j}{\mu k_a'} \tag{16}$$

$$b_j' = \nu[-b_1 a_j + a_1 b_j] \tag{17}$$

$$c = (a_1/\mu)\sqrt{k_a/k_a'}, \quad \text{and} \quad s = (b_1/\mu)\sqrt{k_b/k_a'}. \tag{18}$$

Notice that the sqrt operations disappear in our formulas of (13)–(17), while they are needed in the Givens rotations. Also, the c and s expressions in (18) are not explicitly needed in the computation of (13)–(17). The use of the rotation parameter c in (18) (with one sqrt operation) will be further considered in Section III when the optimum residual e is desired. Furthermore, Section III will show that it is possible to obtain e without any sqrt operation and the explicit computation of the rotation parameter c can be bypassed. To avoid repetitive computations and take the advantage of previously computed results, (14), (16), and (18) use the newly updated k_a' of (13). As stated earlier, we are free to choose those two parameters μ and ν. Different choices of μ and ν will affect the number of multiplications and divisions, as well as the numerical stability and parallelism of these computations.

It can be easily shown that this unified view can generate all of the previously known sqrt-free algorithms via a proper choice of μ and ν. In fact, there has been more than ten sqrt-free algorithms known so far. Among them are Gentleman [5], Hammarling [9], Bareiss [1], Kalson and Yao [10], Ling *et al.* [11], [12], Barlow and Ispen [2], Chen and Yao [4], Götze and Schwiegelsohn [8]. For example, if we choose $\mu = 1$ and $\nu = 1$, it becomes the sqrt-free algorithm proposed by Gentleman in [5] and can be updated as follows:

$$k_a' = k_a a_1^2 + k_b b_1^2 \tag{19}$$

$$k_b' = k_a k_b / k_a' \tag{20}$$

$$a_1' = 1 \tag{21}$$

$$a_j' = (k_a a_1 a_j + k_b b_1 b_j)/k_a', \qquad j = 2, \cdots, p$$

$$b_j' = -b_1 a_j + a_1 b_j. \tag{22}$$

To check that these results are correct, we find that

$$a_j' = \sqrt{k_a'} a_j' = \frac{k_a a_1 a_j + k_b b_1 b_j}{\sqrt{k_a'}} = \frac{\alpha_1 \alpha_j + \beta_1 \beta_j}{\sqrt{\alpha_1^2 + \beta_1^2}} = c_1 \alpha_j + s_1 \beta_j \tag{23}$$

and

$$\beta_j' = \sqrt{k_b'} b_j' = \frac{\sqrt{k_a k_b} b_j'}{\sqrt{k_a'}} = \frac{\alpha_1 \beta_j - \beta_1 \alpha_j}{\sqrt{\alpha_1^2 + \beta_1^2}} = -s\alpha_j + c\beta_j \tag{24}$$

which are consistent with the results in (3) and (4). For the systolic array implementation described in [13], we choose $a_1 = 1$ and define the generalized rotational parameters

$$\bar{c} = k_a/k_a', \quad \text{and} \quad \bar{s} = k_b b_1/k_a'. \tag{25}$$

Then we have

$$k_a' = k_a + k_b b_1^2 \tag{26}$$

$$a_j' = \bar{c} a_j + \bar{s} b_j \tag{27}$$

$$b_j' = b_j - b_1 a_j. \tag{28}$$

These results are consistent with the works by Gentleman [5] and McWhirter [13]. The details of the systolic implementation can be found in [13].

In Table I, we list various sqrt-free algorithms and the corresponding choices of μ and ν. Hence, this class of sqrt-free algo-

TABLE I
SOME KNOWN SQRT-FREE ALGORITHMS OF THE $\mu\nu$ FAMILY

μ	ν	Authors (Year)	Remark
1	1	Gentleman (1973)	$a_1 = 1$
$\frac{k_a a_1^2 + k_b b_1^2}{k_b b_1}$	$\frac{-1}{b_1}$	"	
$\frac{k_a a_1^2 + k_b b_1^2}{k_a a_1}$	$\frac{1}{a_1}$	"	
$\frac{k_a a_1^2 + k_b b_1^2}{k_a a_1}$	$\frac{1}{a_1}$	Hammarling (1974)	
$\frac{k_a a_1^2 + k_b b_1^2}{k_a a_1}$	$\frac{-1}{b_1}$	"	
$\frac{k_a a_1^2 + k_b b_1^2}{k_a a_1}$	$\frac{k_b b_1}{k_a a_1^2}$	"	
1	$\frac{1}{a_1}$	Bareiss (1982)	
$a_1 + k_b b_1^2$	$\frac{1}{a_1 + k_b b_1^2}$	Ling (1989), Kalson/Yao (1985)	$k_a a_1 = 1$
$\frac{k_a a_1^2 + k_b b_1^2}{k_a a_1}$	$\frac{1}{a_1}$	Chen/Yao (1988)	$k_a a_1 = 1$
$\frac{k_a a_1^2 + k_b b_1^2}{k_a k_b}$	1	Götze/Schwiegelshohn (1989)	
$2^{\tau_1}(k_a a_1^2 + k_b b_1^2)$	2^{τ_2}	Barlow/Ispen (1987)	Scaled
$k_a a_1^2 + k_b b_1^2$	$\frac{1}{k_a a_1^2 + k_b b_1^2}$	New algorithm	

TABLE II
COMPARISONS OF COMPUTATIONAL COMPLEXITY OF SOME MEMBERS IN $\mu\nu$ FAMILY

	Multiplication	Division	Addition	Square Root
Givens Rotation	$4p$	1	$2p - 1$	1
$\mu = 1, \nu = 1$	$4p + 3$	1	$2p - 1$	0
$\mu = \frac{k_a a_1^2 + k_b b_1^2}{k_b b_1}, \nu = \frac{-1}{b_1}$	$2p + 6$	2	$2p - 1$	0
$\mu = \frac{k_a a_1^2 + k_b b_1^2}{k_a a_1}, \nu = \frac{1}{a_1}$	$2p + 6$	2	$2p - 1$	0
$\mu = 1, \nu = \frac{1}{a_1}$	$4p + 4$	2	$2p - 1$	0
$\mu = a_1 + k_b b_1^2, \nu = \frac{1}{a_1 + k_b b_1^2}$ $(k_a a_1 = 1)$	$4p + 5$	1	$2p - 2$	0
$\mu = \frac{k_a a_1^2 + k_b b_1^2}{k_a k_b}, \nu = 1$	$4p + 6$	2	$2p - 1$	0
$\mu = k_a a_1^2 + k_b b_1^2, \nu = \frac{1}{k_a a_1^2 + k_b b_1^2}$	$4p + 4$	1	$2p - 1$	0

rithms is called the $\mu\nu$ family of sqrt-free Givens rotation algorithms.

Now, not only can we generate those known sqrt-free algorithms, but we are also able to find new sqrt-free algorithms by choosing new pairs of (μ, ν) parameters. As an example, let us choose

$$\mu = k_a a_1^2 + k_b b_1^2$$

$$\nu = \frac{1}{k_a a_1^2 + k_b b_1^2}$$

then we can readily verify that this is a new sqrt-free algorithm. In fact, the new sqrt-free algorithm is among the best in the list of Table I in terms of number of divisions, e.g., it only requires one division and no square root. In principle, there are unlimited choices of μ and ν for sqrt-free algorithms. Table II shows comparisons of computational complexity of some algorithms listed in Table I.

III. SQRT-Free Triangular Array Updating and Optimum Residual Acquisition

Solving a full rank LS problem $Aw \approx b$ ($A \in \Re^{m \times n}$, $m \geq n$) without the sqrt operation can be easily achieved [5]. Let the QRD of A be $Q^T A = R$, where R is an upper triangular matrix, then

$$Q^T[A \quad b] = \begin{bmatrix} R & u \\ 0 & v \end{bmatrix} \tag{29}$$

and the optimum weight vector can be obtained by solving $R\hat{w} = u$. Now, starting with a full dense augmented matrix $\sqrt{K}[A \quad b]$, a series of sqrt-free rotations can be applied to zero out the subvector below the main diagonal of the underlying matrix to obtain

$$\sqrt{K}\begin{bmatrix} \bar{R} & \bar{u} \\ 0 & \bar{v} \end{bmatrix}$$

where $\sqrt{K} = \text{diag}(\sqrt{k_1}, \cdots, \sqrt{k_m})$ and $R = \sqrt{K}\bar{R}$, $u = \sqrt{K}\bar{u}$. Since the explicit computation of $\sqrt{K}$ is not required, the optimum weight vector can be obtained without the sqrt operation by solving $\bar{R}\hat{w} = \bar{u}$.

In the following, we will apply the developed prototypes of sqrt-free rotations developed before to the QRD-based RLS estimation problem where we are only interested in the optimum residual. How to obtain the optimum residual by using the systolic array [13] has been well known. To be specific, we are interested in updating from

$$\begin{bmatrix} R & u \\ x^T & y \end{bmatrix} \quad to \quad \begin{bmatrix} R' & u' \\ 0^T & v \end{bmatrix}. \tag{30}$$

It has been shown [13] that the $p \times p$ upper triangular matrix R' can be obtained through a sequence of p Givens rotations, and the optimum residual e for the newly appended data $[x^T \; y]$ is given by

$$e = -\left(\prod_{i=1}^{p} c_i\right) v \tag{31}$$

with c_i representing the cosine value of the ith rotation angle.

Factoring out the scaling constants into the premultiplying diagonal matrix leads (30) to the form of

$$\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1p} & u_1 \\ & r_{22} & \cdots & r_{2p} & u_2 \\ & & \ddots & \vdots & \vdots \\ & & & r_{pp} & u_p \\ x_1 & x_2 & \cdots & x_p & y \end{bmatrix} = \begin{bmatrix} \sqrt{k_1} & & & & \\ & \sqrt{k_2} & & & \\ & & \ddots & & \\ & & & \sqrt{k_p} & \\ & & & & \sqrt{k_q} \end{bmatrix} \cdot \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1p} & a_{1,p+1} \\ & a_{22} & \cdots & a_{2p} & a_{2,p+1} \\ & & \ddots & \vdots & \vdots \\ & & & a_{pp} & a_{p,p+1} \\ b_1 & b_2 & \cdots & b_p & b_{p+1} \end{bmatrix}. \tag{32}$$

Unlike the previously developed formula, where we are only interested in updating k_i, a_{ij}, to k_i', a_{ij}' and zeroing out all the b_i's, this time we also need to know the cosine values explicitly as required in the optimum residual given in (31).

After the first rotation, b_1 will be zeroed out and we have

$$k_1' = \frac{k_1 a_{11}^2 + k_q b_1^2}{\mu_1^2} \tag{33}$$

$$k_q^{(1)} = \frac{k_1 k_q}{\mu_1^2 \nu_1^2 k_1'} \tag{34}$$

$$a_{11}' = \mu_1 \tag{35}$$

$$a_{1j}' = \frac{1}{\mu_1 k_1'}[k_1 a_{11} a_{1j} + k_q b_1 b_j], \qquad j = 2, \cdots, p+1 \tag{36}$$

$$b_j^{(1)} = \nu_1[-b_1 a_{1j} + a_{11} b_j], \tag{37}$$

$$c_1 = \frac{a_{11}}{\mu_1}\sqrt{\frac{k_1}{k_1'}} \tag{38}$$

with (μ_1, ν_1) being the parameter pair which are still free to be chosen later. Note the close analogy of (33)–(38) to those of (13)–(18). Similarly, after the ith rotation ($1 < i \leq p$), we have

$$k_i' = \frac{k_i a_{ii}^2 + k_q^{(i-1)} b_i^{(i-1)^2}}{\mu_i^2} \tag{39}$$

$$k_q^{(i)} = \frac{k_i k_q^{(i-1)}}{\mu_i^2 \nu_i^2 k_i'} \tag{40}$$

$$a_{ii}' = \mu_i, \tag{41}$$

$$a_{ij}' = \frac{1}{\mu_i k_i'}[k_i a_{ii} a_{ij} + k_q^{(i-1)} b_i^{(i-1)} b_j^{(i-1)}], \qquad j = 2, \cdots, p+1 \tag{42}$$

$$b_j^{(i)} = \nu_i[-b_i^{(i-1)} a_{ij} + a_{ii} b_j^{(i-1)}], \tag{43}$$

$$c_i = \frac{a_{ii}}{\mu_i}\sqrt{\frac{k_i}{k_i'}}. \tag{44}$$

Finally, after p rotations are finished, (32) becomes

$$\begin{bmatrix} \sqrt{k_1'} & & & & \\ & \sqrt{k_2'} & & & \\ & & \ddots & & \\ & & & \sqrt{k_p'} & \\ & & & & \sqrt{k_q} \end{bmatrix} \begin{bmatrix} a_{11}' & a_{12}' & \cdots & a_{1p}' & a_{1,p+1}' \\ & a_{22}' & \cdots & a_{2p}' & a_{2,p+1}' \\ & & \ddots & \vdots & \vdots \\ & & & a_{pp}' & a_{p,p+1}' \\ 0 & 0 & \cdots & 0 & b_{p+1}^{(p)}, \end{bmatrix} \tag{45}$$

which has the form of

$$\begin{bmatrix} R' & u' \\ 0^T & v \end{bmatrix}$$

in (30). The optimum residual e in (31) now becomes

$$e = -\left(\prod_{i=1}^{p} \frac{a_{ii}}{\mu_i}\sqrt{\frac{k_i}{k_i'}}\right)\sqrt{k_q^{(p)}}\, b_{p+1}^{(p)} \tag{46}$$

where $k_q^{(p)}$ is defined in (40). To further simplify the expression in (46), we notice that $k_q^{(p)}$ can be computed recursively as follows:

$$k_q^{(p)} = \left(\frac{\sqrt{k_p/k_p'}}{\mu_p \nu_p}\right)^2 k_q^{(p-1)} \tag{47}$$

$$= \left(\frac{\sqrt{k_p/k_p'}}{\mu_p \nu_p}\right)^2 \left(\frac{\sqrt{k_{p-1}/k_{p-1}'}}{\mu_{p-1}\nu_{p-1}}\right)^2 k_q^{(p-2)} \tag{48}$$

$$\vdots$$

$$= \left[\prod_{i=1}^{p}\left(\frac{\sqrt{k_i/k_i'}}{\mu_i \nu_i}\right)^2\right] k_q \tag{49}$$

where (40) is used in the recursion.

With (49) substituted into (46), we have

$$e = -\left(\prod_{i=1}^{p} \frac{a_{ii}}{\mu_i^2 \nu_i} \frac{k_i}{k_i'}\right) \sqrt{k_q}\, b_{p+1}^{(p)}. \tag{50}$$

Because $\sqrt{k_q}\,[b_1, b_2, \cdots, b_p, b_{p+1}] = [x_1, x_2, \cdots, x_p, y]$ is the appended new data row, we are certainly free to choose $k_q = 1$ to reduce the arithmetic complexity and simplify the expression in (50). Therefore, a lemma on obtaining the sqrt-free optimum residual is given below.

Lemma 1: (Sqrt-free Optimum Residual): The optimum residual e can be computed with no sqrt operations and is given by

$$e = -\left(\prod_{i=1}^{p} \frac{a_{ii}}{\mu_i^2 \nu_i} \frac{k_i}{k_i'}\right) b_{p+1}^{(p)} \tag{51}$$

if k_q is chosen to be unity. □

McWhirter [13] successfully employed Gentleman's proposition [5] in computing the residual e without sqrt operations. By choosing $\mu_i = \nu_i = a_{ii} = 1$, $1 \le i \le p$, the optimum residual can be reduced to

$$e = -\left(\prod_{i=1}^{p} \frac{k_i}{k_i'}\right) b_{p+1}^{(p)} = -\left(\prod_{i=1}^{p} \bar{c}_i\right) b_{p+1}^{(p)}$$

$$\text{(Gentleman/McWhirter)} \tag{52}$$

where $\bar{c}_i$ is defined in (25). This result is again consistent with the work by McWhirter [13].

Another example can be taken from Hammarling's suggestion [9] as follows:

$$\mu_i = \frac{k_i a_{ii}^2 + k_q^{(i-1)} b_i^{(i-1)2}}{k_i a_{ii}}, \qquad i = 1, \cdots, p. \tag{53}$$

$$\nu_i = 1/a_{ii}, \tag{54}$$

Then it follows that $k_i' = k_i a_{ii}/\mu_i$ and the Hammarling optimum residual is given by

$$e = -\left(\prod_{i=1}^{p} \frac{a_{ii}}{\mu_i^2 \nu_i} \frac{\mu_i}{a_{ii}}\right) b_{p+1}^{(p)} \tag{55}$$

$$= -\left(\prod_{i=1}^{p} \frac{1}{\mu_i \nu_i}\right) b_{p+1}^{(p)} \tag{56}$$

$$= -\left(\prod_{i=1}^{p} \frac{a_{ii}}{a_{ii}'}\right) b_{p+1}^{(p)}. \tag{57}$$

IV. Conclusions

The Givens rotation is the most commonly used method in performing QRD updating. Most of these rotation-based methods require explicit sqrt computations which are undesirable from the practical VLSI circuit design point of view. Our work is the first effort to establish basic understanding of all known sqrt-free QRD algorithms, from which the basic criterion is seen to be simple. We have shown that all the current known sqrt-free algorithms belong to the $\mu\nu$ family. New sqrt-free algorithms can be easily obtained from this $\mu\nu$ family. The issue of choosing optimal parameters μ and ν in terms of computational complexity (hardware and software) and numerical properties still remains an open question. This unified approach also provides a fundamental framework for the sqrt-free RLS algorithm, which is essential for fast operations and practical VLSI implementations.

References

[1] E. H. Bareiss, "Numerical solution of the weighted least squares problems by G-transformations," Tech. Rep. 82-03-NAM-03, Dep. Elec. Eng. Comput. Sci., Northwestern University, Evanston, IL, Apr. 1982.

[2] J. L. Barlow and I. C. F. Ipsen, "Scaled Givens rotations for the solution of linear least squares problems on systolic arrays," *SIAM J. Sci. Stat. Comput.*, vol. 8, no. 5, pp. 716-733, Sept. 1987.

[3] G. J. Bierman, *Factorization Methods for Discrete Sequential Estimation*. New York: Academic, 1977, p. 44.

[4] M. J. Chen and K. Yao, "Comparisons of *QR* least squares algorithms for systolic array processing," in *Proc. Conf. Inform. Sci. Syst.*, Mar. 1988, pp. 683-688.

[5] W. M. Gentleman, "Least squares computations by Givens transformations without square roots," *J. Inst. Math. Appl.*, vol. 12, pp. 329-336, 1973.

[6] W. M. Gentleman and H. T. Kung, "Matrix triangularization by systolic arrays," *Proc. SPIE Int. Soc. Opt. Eng.*, vol. 298, pp. 19-26, 1981.

[7] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed. Baltimore, MD: Johns Hopkins Press, 1989.

[8] J. Götze and U. Schwiegelshohn, "An orthogonal method for solving systems of linear equations without square roots and with few divisions," in *Proc. IEEE ICASSP*, 1989, pp. 1298-1301.

[9] S. Hammarling, "A note on modifications to the Givens plane rotation," *J. Inst. Math. Appl.*, vol. 13, pp. 215-218, 1974.

[10] S. Kalson and K. Yao, "Systolic array processing for order and time recursive generalized least squares estimation," *Proc. SPIE Int. Soc. Opt. Eng.*, vol. 564, pp. 28-38, 1985.

[11] F. Ling, D. Manolakis, and J. G. Proakis, "A recursive modified Gram-Schmidt algorithm for least squares estimation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, no. 4, pp. 829-836, Aug. 1986.

[12] F. Ling, "Efficient least squares lattice algorithm based on Givens rotations with systolic array implementation," in *Proc. IEEE ICASSP*, 1989, pp. 1290-1293.

[13] J. G. McWhirter, "Recursive least squares minimization using a systolic array," *Proc. SPIE Int. Soc. Opt. Eng.*, vol. 431, pp. 105-112, 1983.

[14] I. K. Proudler, J. G. McWhirter, and T. J. Shepherd, "The QRD-based least squares lattice algorithm: Some computer simulations using finite wordlength," in *Proc. IEEE ISCAS*, New Orleans, May 1990, pp. 258-261.

[15] S. F. Hsieh, K. J. R. Liu, and K. Yao, "A unified sqrt-free rank-1 up/downdating approach for recursive least squares problems," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, Toronto, May 1991, pp. 1017-1020.

Phase Retrieval Using a Window Function

Wooshik Kim and Monson H. Hayes

***Abstract*—In this correspondence, we consider the problem of reconstructing a signal from the Fourier intensity of the signal and the Fourier intensities of two windowed signals: one by a window $w(n)$ and the other by its complementary window $[1 - w(n)]$. We develop several conditions under which a signal can be uniquely specified to within several trivial ambiguities such as sign, translation, and time reversal from the given conditions. We present a possible reconstruction algorithm derived from the Gerchberg-Saxton algorithm.**

Manuscript received February 5, 1992; revised August 6, 1992. This work was supported by the Joint Services Program under Contract DAAL-03-90-C-0004.

The authors are with the School of Electrical Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0250.

IEEE Log Number 9205997.