AN ADAPTIVE LS ALGORITHM BASED ON ORTHOGONAL HOUSEHOLDER TRANSFORMATIONS

Athanasios A. Rontogiannis[1] *and Sergios Theodoridis*[2]
Department of Informatics, Division of Communications and Signal Processing
Panepistimioupolis, TYPA Buildings, University of Athens
GR-157 71 Zografou, GREECE
[1] *e-mail : tronto@di.uoa.gr*
[2] *e-mail : stheodor@di.uoa.gr*

ABSTRACT

This paper presents an adaptive exponentially weighted algorithm for least squares (LS) system identification. The algorithm updates an inverse "square root" factor of the input data correlation matrix, by applying numerically robust orthogonal Householder transformations. The scheme avoids, almost entirely, costly square roots and divisions (present in other numerically well behaved adaptive LS schemes) and provides directly the estimates of the unknown system coefficients. Furthermore, it offers enhanced parallelism, which leads to efficient implementations. A square array architecture for implementing the new algorithm, which comprises simple operating blocks, is described. The numerically robust behaviour of the algorithm is demonstrated through simulations.

1. INTRODUCTION

Adaptive least squares algorithms for system identification [5],[4], are popular due to their fast converging properties and are used in a variety of applications, such as channel equalization, echo cancellation, spectral analysis, control, to name but a few. Among the various efficiency issues, characterizing the performance of an algorithm, those of parallelism and numerical robustness are of particular importance, especially in applications where medium to long filter lengths are required. Sometimes it may be preferable to use an algorithm of higher computational complexity, but with good numerical error properties and high parallelism, since this may allow its implementation with shorter wordlenghts and fixed point arithmetic on an array architecture. This has led to the development of adaptive algorithms based on numerically robust orthogonal transformations.

The use of Householder transformations in the LS framework, has been so far restricted in block type problems [8],[6]. This is a direct consequence of the nature of a Householder matrix, which is usually applied to annul block of elements in vectors or matrices [3]. Thus, Householder transformations have been used either in simple block LS updates (downdates) [8] or in block recirsice LS (RLS) schemes [6]. This paper presents an $O(p^2)$, RLS algorithm (p being the order of the system) which springs from an inverse square root factor of the data correlation matrix and incorporates numerically robust block orthogonal Householder transformations. The algorithm updates a square factor, and computes the filter parameters directly, without involving matrix inversions or backsubstitution steps. The proposed scheme avoids almost entirely divisions and square roots and employs low cost additions and multiplications. Furthermore, it exhibits high degree of parallelism, which makes it amenable to efficient implementation. An array architecture is described that efficiently implements the new scheme. This architecture is designed to take full advantage of the algorithm's parallelism, and it comprises very simple operating blocks. We must state that the derived algorithm consists of matrix-vector and vector-vector operations, which make it suitable for efficient implementations on vector processors. Throughout the paper, the algorithm is compared with the newly proposed scheme of [1] which updates the inverse Cholesky factor of the data correlation matrix via orthogonal Givens rotations.

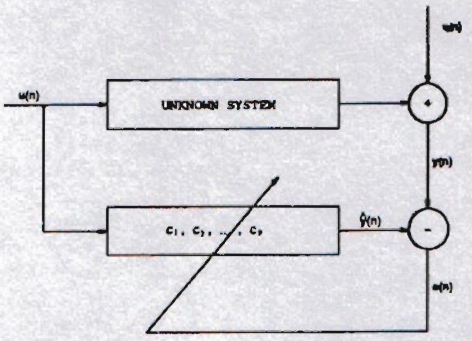

Figure 1. System identification problem

2. PROBLEM FORMULATION

Figure 1 illustrates the typical system identification task, which is our main concern in this paper. Given an unknown FIR system, excited by an input signal $u(n)$, we seek the estimates of the p unknown tap coefficients so that the error between the measured output of the system $y(N)$ and the output of an associated model $\hat{y}(N)$, is minimum in the least squares sense. That is the sum

$$\| \epsilon(N) \|^2 = \sum_{n=1}^{N} \lambda^{N-n} [y(n) - \mathbf{c}^T(N)\mathbf{u}(n)]^2 \qquad (1)$$

to be minimum, where λ is the usual forgetting factor with $0 \ll \lambda \leq 1$, and

$$\epsilon^T(N) = [\epsilon(1), \epsilon(2), \ldots, \epsilon(N)]$$

$$\mathbf{c}^T(N) = [c_1(N), c_2(N), \ldots, c_p(N)]$$

$$\mathbf{u}^T(n) = [u(n), u(n-1), \ldots, u(n-p+1)]$$

The quantity $\eta(n)$ in the figure stands for the measurement noise. It is well known that the LS solution $\mathbf{c}(N)$ is obtained from the following normal equations

$$R(N)\mathbf{c}(N) = \mathbf{d}(N) \tag{2}$$

$R(N), \mathbf{d}(N)$ are the estimates of the input data correlation matrix and the crosscorrelation vector between the input and the desired response, respectively. These quantities are recursively updated as follows [5]

$$R(N) = \lambda R(N-1) + \mathbf{u}(N)\mathbf{u}^T(N) \tag{3}$$

$$\mathbf{d}(N) = \lambda \mathbf{d}(N-1) + \mathbf{u}(N)y(N) \tag{4}$$

The unknown filter taps $\mathbf{c}(N)$ can be computed either from (2) or by employing the following time update formula [5]

$$\mathbf{c}(N) = \mathbf{c}(N-1) + \mathbf{w}(N)\frac{e(N)}{\delta^2(N)} \tag{5}$$

In the last equation $e(N)$ stands for the a priori LS error given by

$$e(N) = y(N) - \mathbf{c}^T(N-1)\mathbf{u}(N) \tag{6}$$

and $\mathbf{w}(N)$ corresponds to the dual Kalman gain defined as

$$\mathbf{w}(N) = \frac{R^{-1}(N-1)\mathbf{u}(N)}{\lambda} \tag{7}$$

The quantity $\delta(N)$ can be expressed as follows

$$\delta(N) = \sqrt{1 + \lambda^{-1}\mathbf{u}^T(N)R^{-1}(N-1)\mathbf{u}(N)} \tag{8}$$

In the following sections we develop a "square-root" algorithm that solves the RLS problem. The proposed algorithm updates and applies a square factor, a fact which necessitates the use of numerically robust orthogonal Householder transformations.

3. DERIVATION OF THE ALGORITHM

An orthogonal Householder matrix has the following special form [3],[8]

$$P = I - 2\frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}} \tag{9}$$

It is obvious from (9) that matrix P is also symmetric. Householder transformations are often used to annul block of elements in matrices or vectors by appropriately selecting the Householder vector $\mathbf{v}$ in (9). More specifically, if $\mathbf{x}$ is a nonzero vector and $\mathbf{e}_i$ stands for the unit vector with 1 in the i-th position, then it can be shown [3] that when

$$\mathbf{v} = \mathbf{x} \pm \| \mathbf{x} \| \mathbf{e}_i \tag{10}$$

then

$$P\mathbf{x} = \mp \| \mathbf{x} \| \mathbf{e}_i \tag{11}$$

Note the sign difference in equations (10) and (11). It is worth emphasizing that vectors $\mathbf{v}$ and $\mathbf{x}$ are identical except for the i-th element. In most cases, explicit formation of the Householder matrix from (9) is not required. Instead, we usually aim to take advantage of the matrice's special structure, which is also the case in our analysis.

3.1. The HRLS algorithm

Let us assume that the $p \times p$ matrix $A(N)$ stands for an arbitrary square root factor of the input data autocorrelation matrix $R(N)$ for every N, that is

$$R(N) = A^T(N)A(N) \tag{12}$$

We define the $p \times 1$ vector $\mathbf{k}(N)$ as

$$\mathbf{k}(N) = \frac{A^{-T}(N-1)\mathbf{u}(N)}{\sqrt{\lambda}} \tag{13}$$

and assume that $P(N)$ is a $(p+1) \times (p+1)$ Householder matrix which nullifies the first p elements of the vector $[\mathbf{k}^T(N), 1]^T$. We will show that $P(N)$ also updates $A^{-T}(N-1)$ and produces a scaled version of the dual Kalman gain. Indeed, let us consider the following expression

$$P(N)\begin{bmatrix} \lambda^{-1/2}A^{-T}(N-1) & \mathbf{k}(N) \\ \mathbf{0}^T & 1 \end{bmatrix} =$$

$$\begin{bmatrix} X(N) & \mathbf{0} \\ \mathbf{z}^T(N) & -\delta(N) \end{bmatrix} \tag{14}$$

where $X(N)$ and $\mathbf{z}(N)$ are quantities to be identified. Note that, according to (8) and (13), $\mathbf{k}(N)$ and $\delta(N)$ are related as follows

$$\delta(N) = \sqrt{1 + \mathbf{k}^T(N)\mathbf{k}(N)} \tag{15}$$

From the definition of the Householder matrix we see that, in order for $P(N)$ to annihilate $\mathbf{k}(N)$ as in (14), it must be of the form

$$P(N) = I - 2\frac{\mathbf{v}(N)\mathbf{v}^T(N)}{\mathbf{v}^T(N)\mathbf{v}(N)} \tag{16}$$

where the Householder vector $\mathbf{v}(N)$ equals to

$$\mathbf{v}(N) = \begin{bmatrix} \mathbf{k}(N) \\ 1 + \delta(N) \end{bmatrix} \tag{17}$$

The selection of the positive sign in the last element of $\mathbf{v}(N)$ aims to prevent from numerical problems, which can arise by division with a very small number in (16). If we now set

$$\beta(N) = \frac{2}{\mathbf{v}^T(N)\mathbf{v}(N)} \tag{18}$$

then combination of (15) and (17) easily results in

$$\beta(N) = \frac{1}{\delta(N)[1 + \delta(N)]} \tag{19}$$

Furthermore, (16) is rewritten as

$$P(N) = I - \beta(N)\mathbf{v}(N)\mathbf{v}^T(N) \tag{20}$$

If we now multiply the matrices in both sides of (14) with their transposes, the orthogonality of $P(N)$ leads to the following expressions

$$\mathbf{z}(N) = -\frac{A^{-1}(N-1)\mathbf{k}(N)}{\sqrt{\lambda}\delta(N)} = -\frac{\mathbf{w}(N)}{\delta(N)} \tag{21}$$

```
For $N = 1 \ldots$ do,
    1. $\mathbf{k}(N) = \lambda^{-1/2} A^{-T}(N-1)\mathbf{u}(N)$;
    2. $\delta(N) = \sqrt{1 + \mathbf{k}^T(N)\mathbf{k}(N)}$;
    3. $\beta(N) = \frac{1}{\delta(N)[1+\delta(N)]}$;
    4. $\tilde{\mathbf{w}}(N) = A^{-1}(N-1)\mathbf{k}(N) \quad (= \lambda \mathbf{w}(N))$;
    5. $\hat{\mathbf{w}}(N) = \lambda^{-1/2}\beta(N)\tilde{\mathbf{w}}(N)$;
    6. $A^{-T}(N) = \lambda^{-1/2} A^{-T}(N-1) - \mathbf{k}(N)\hat{\mathbf{w}}^T(N)$;
    7. $e(N) = y(N) - \mathbf{c}^T(N-1)\mathbf{u}(N)$;
    8. $\mathbf{c}(N) = \mathbf{c}(N-1) + \frac{e(N)}{\sqrt{\lambda\delta^2(N)}}\tilde{\mathbf{w}}(N)$;
Initialization (Soft-constrained)
$\mathbf{c}(0) = \mathbf{0}, \quad A^{-T}(0) = \frac{1}{\mu\lambda^p} diag[1, \lambda, \ldots, \lambda^{p-1}]$
```

Figure 2. The new HRLS algorithm

$$X^T(N)X(N) = \frac{R^{-1}(N-1)}{\lambda^{-1}} - \frac{\mathbf{w}^T(N)\mathbf{w}(N)}{\delta^2(N)} \qquad (22)$$

Equation (21) verifies that $\mathbf{z}(N)$ is a scaled version of the dual Kalman gain. Moreover, application of the well known matrix inversion lemma [4] to (3) results in

$$R^{-1}(N) = \lambda^{-1}R^{-1}(N-1) - \frac{\mathbf{w}(N)\mathbf{w}^T(N)}{\delta^2(N)} \qquad (23)$$

From the last two equations it is straightforward that $X(N) = A^{-T}(N)$.
Combination of equations (13),(15),(19), (17),(20),(14) (with $X(N) = A^{-T}(N)$ and $\mathbf{z}(N)$ given as in (21)), (6),(5), leads to the algorithm of figure 2. As mentioned before, explicit formation of matrix $P(N)$ is not necessary. The soft-constrained approach [5], is adopted for the algorithm's initialization. Note that the filter taps are directly computed without the use of matrix inversions or back-substitution steps. This is also a characteristic of the inverse QR algorithm, which is based on Givens rotations and is described in [1]. The complexities of these two algorithms are shown in table 1. The complexity of the HRLS algorithm is higher with respect to additions and slightly higher w.r.t multiplications. This was expected since the new scheme manipulates a square matrix instead of a triangular one that is the case in [1]. However, note that the HRLS algorithm avoids almost exclusively divisions and square roots while employing low cost additions and multiplications, in a multiply-add fashion, which is desirable for efficient DSP implementations. On the other hand, the inverse QR scheme requires $O(p)$ divisions and square roots. The numerical properties of the HRLS algorithm are expected to be very favorable because of the use of numerical robust Householder reflections. Moreover, the derived scheme offers enchanced parallelism, as compared to [1], a notion that is further analysed in the next section.

4. PARALLEL POTENTIAL OF THE ALGORITHM

An array architecture for implementing the HRLS algorithm is depicted in figure 3 (for the case $p = 3$). At time N, the square cells of the array store and update the elements of matrix $A^{-T}(N-1)$ (step 6). They also perform the inner product computations, required in the derivation of the vectors $\mathbf{k}(N)$ (step 1) and $\tilde{\mathbf{w}}(N)$ (step 4). The cyclic cells, at the bottom of the array, store and update the filter taps (step 8) and produce the a priori error $e(N)$ (step 7). The cyclic cells on the right of the array compute $\delta^2(N)$ and $\beta(N)$ (steps 2 and 3 of the algorithm). The emboldened cyclic cells simply transform the gain $\tilde{\mathbf{w}}(N)$ in the form $\lambda^{-1/2}\beta(N)\tilde{\mathbf{w}}(N)$, while the bottom right cell produces the quantity $\frac{e(N)}{\sqrt{\lambda\delta^2(N)}}$ that is required in the calculation of $\mathbf{c}(N)$. The functionality of all the individual cells is shown in figure 3.

Alg.	Add.'s	Mlt.'s	D.'s/S.'s
IQR	$\frac{3}{2}p^2 + O(p)$	$\frac{7}{2}p^2 + O(p)$	$3p + 1$
HRLS	$3p^2 + O(p)$	$4p^2 + O(p)$	3

Table 1. Comparison of complexities of Inverse QR [1] and HRLS

We must note that the elements of both vectors $\mathbf{k}(N)$ and $\tilde{\mathbf{w}}(N)$ are generated in parallel. As it is illustrated in figure 3, the new input vector $\mathbf{u}(N)$ is applied in parallel to the row cells of $A^{-T}(N-1)$ allowing for the simultaneous calculation of $k_i(N)$'s. This is accomplished by performing inner product computations in a left-to-right procedure. The thus computed vector $\mathbf{k}(N)$ concurrently excites the column cells of $A^{-T}(N-1)$ and is stored there. Then, the elements of $\tilde{\mathbf{w}}(N)$ can be simultaneously produced as inner products of $\mathbf{k}(N)$ with the columns of $A^{-T}(N-1)$ in a top-to-bottom procedure.
It is worth stating that, different steps of the HRLS algorithm can be executed in parallel, thus reducing the overall computation time. More specifically

1. Having computed $\mathbf{c}(N-1)$ in the previous time instant, vector $\mathbf{k}(N)$ can be calculated in parallel with the a priori error $e(N)$ at the bottom of the array.
2. The computation of $\tilde{\mathbf{w}}(N)$ and that of $\delta^2(N)$ on the right of the array, can be simultaneously performed.
3. All the elements of $A^{-T}(N-1)$ are simultaneously updated according to the formula

   $$a_{ij}(N) = \lambda^{-1/2}a_{ij}(N-1) - k_i(N)\tilde{w}_j(N)$$

   This can be accomplished in parallel with the update of the filter taps at the bottom of the array.
4. The vector $\lambda^{-1/2}\beta(N)\tilde{\mathbf{w}}(N)$ in the intermediate cells and the quantity $\frac{e(N)}{\sqrt{\lambda\delta^2(N)}}$ at the bottom right cell can be calculated concurrently.

It is clear from the above discussion that the computation time per time update iteration is that required for the completion of $2p+1$ multiplications/additions (MADs) and 2 divisions/square roots. Moreover, if we exploit the inherent pipelining of inner products, the above measure can be reduced from $(2p+1)$ to $(2\log_2 p + 1)$ MADs. In any case, a substantial improvement is offered w.r.t the algorithm of [1],[2] where the computation time per time update corresponds to $8p$ multiplications, $3p$ additions, $2p$ divisions and p square roots. Thus, a considerable reduction in the throughput rate is achieved. Moreover, the processing units are of the

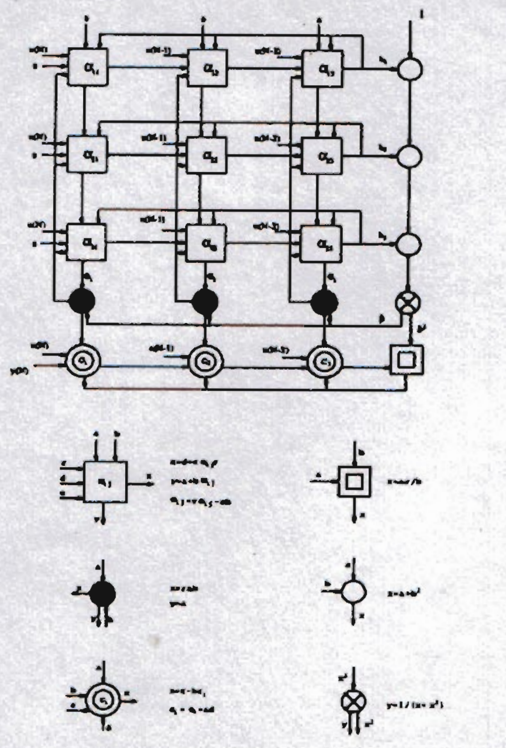

Figure 3. An array architecture that implements the HRLS algorithm, $r = \lambda^{-1/2}$

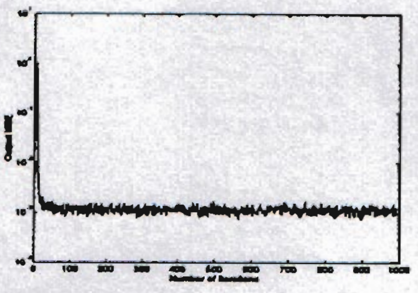

Figure 4. Initial convergence curves

simple multiply-add type, and the number of divisions is independent of p. In contrast, $O(p)$ dividers are needed in [2], which is usually not a desirable feature if VLSI implementation is considered.

5. SIMULATIONS

In order to verify the correctness of the HRLS algorithm a system identification problem is considered. The unknown FIR system is time invariant of order 8, the SNR is 30dB, the forgetting factor $\lambda = 0.98$ and the initialization parameter $\mu = 0.01$. The input signal and the noise are chosen to be Gaussian white noise processes. In figure 4 two initial convergence curves are overlaid although they are not distinguished. One corresponds to the HRLS scheme and the other to the inverse QR algorithm of [1]. The curves are the average of 100 realizations.
In order to validate the performance of the HRLS algorithm in a low precision environment, we implemented the new scheme using floating point arithmetic with different (decreasing) mantissa lengths. It was observed that even for very small mantissa lengths (2 and 3) the algorithm does not diverge, that is, accumulation of round-off errors does not

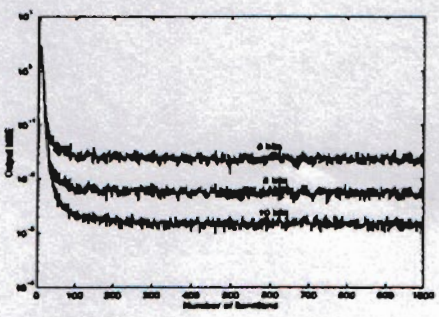

Figure 5. Initial convergence curves for different mantissa lengths

occur. In figure 5, the ensemble average squared error of the previous example for mantissa lengths 6,8 and 10 is illustrated. It is clear that the squared error retains an acceptable level even under low precision conditions.

6. CONCLUSIONS

An adaptive least squares algorithm based on Householder reflections, was developed in this paper. The new scheme employs low cost additions and multiplications and computes directly the unknown coefficient estimates without the use of backsubstitution. The algorithm's enhanced parallelism combined to its numerically stable behaviour can lead to efficient implemenation of the new scheme on parallel architectures with short wordlenghts and fixed-point arithmetic. The algorithm is favorably compared to the recently developed inverse QR scheme (both being square root inverse factorization schemes [7]) in terms of computational complexity, parallel properties and numerical behaviour.

REFERENCES

[1] S.T. Alexander, A.L. Ghirnikar, "A method for recursive least squares adaptive filtering based upon an inverse QR decomposition", *IEEE Trans. SP-41*, no. 1, pp. 20-30, Jan. 1993.

[2] A.L. Ghirnikar, S.T. Alexander, R.J. Plemmons, "A parallel implementation of the inverse QR adaptive filter", *Comput. Elec. Eng.*, vol. 18, no. 3/4, pp. 291-300, 1992.

[3] G.H. Golub, C. Van Loan, *Matrix Computations*, 2nd Edition, Baltimore : Johns Hopkins Press, 1989.

[4] S. Haykin, *Adaptive filter theory*, Engelwood Cliffs, NJ : Prentice Hall, 1991.

[5] N. Kalouptsidis and S. Theodoridis eds., *Adaptive system identification and signal processing algorithms*, Engelwood Cliffs, NJ : Prentice Hall, 1993.

[6] K.R. Liu, S.F. Hsieh, K. Yao, "Systolic block Householder transformation for RLS algorithm with two-level pipelined implementation", *IEEE Trans. SP-40*, no. 4, pp. 946-958, April 1992.

[7] A.A. Rontogiannis, S. Theodoridis, "On inverse factorization adaptive least squares algorithms", (to appear), *Signal Processing*, 1996.

[8] A.O. Steinhardt, "Householder transforms in signal processing", *IEEE ASSP Magazine*, pp. 4-12, July 1988.