An Order Downdating Algorithm for Tracking System Order and Parameters in Recursive Least Squares Identification

Daniel W. Apley and Jianjun Shi

Abstract— In this correspondence, a new time and order recursive method for on-line tracking of system order and parameters using recursive least squares (RLS) is presented. The method consists of two parts: a time updating portion that uses existing RLS inverse QR decomposition algorithms and a new computationally efficient "order downdating" portion that calculates the model parameters and residual error energies for an entire set of models with order varying from one to some prespecified maximum model order.

Index Terms—Model-order identification, QR factorization, recursive least squares, system identification.

I. INTRODUCTION

Recursive least squares (RLS) is a widely used method for tracking model parameters in system identification and parametric spectral estimation. In many applications, the system order, in addition to the parameters, may be time-varying or otherwise unknown. One common example is in adaptive control of robotic structures, where new modes of vibration are introduced when the robot changes configuration or the payload changes [1]. Other examples come from the area of on-line machine diagnostics. New modes of vibration can be introduced when machine components become weakened or loosened, which changes the system order. On-line detection of component failure requires tracking system order changes. In chatter detection in machining of metals, chatter can occur at simultaneous multiple frequencies [2], which also requires on-line tracking of the system order. Consequently, an efficient RLS scheme for tracking system order and parameters would be useful.

Most of the existing research on "order recursive" methods has been in the context of off-line batch least squares (LS) modeling [3]–[5]. The common idea behind these methods is to iteratively identify models with successively increasing order. The various order models are compared along the way, and the procedure can be stopped when one of the models is deemed an adequate representation of the data. While these methods are efficient as batch methods, they are not suitable for on-line, time recursive identification. In [6] and [7], some of the concepts in the above order-recursive batch methods were extended to develop time- and order-recursive LS methods. However, these methods have two drawbacks. The computational expense per timestep is much larger than that of standard RLS, which is used to identify the highest order model. In addition, the methods apply only to growing length or sliding rectangular windows on the data and not to the widely used exponentially weighted RLS.

This correspondence introduces a new method for obtaining the RLS solution to a set of models of various order: up to some predetermined maximum model order n. The various order models can then be compared to select the most appropriate one, providing an efficient means of on-line tracking of system order and parameters. The method relies on the straightforward geometric interpretation of the QR decomposition in LS.

The new method has a number of attractive features.

1) The computational expense is only slightly larger than that of standard RLS (using, e.g., Bierman's $\mathbf{UDU}^T$ algorithm [8]) applied to the highest order model—roughly $2n^2$, as opposed to $1.5n^2$, multiplications per timestep.
2) The method is generic in the sense that it can be applied to arbitrary RLS situations, such as ARX, ARMAX, or impulse response modeling with single or multiple inputs.
3) The method can be applied to exponentially weighted RLS.

The format of the remainder of the paper is as follows. Section II provides a brief background on the QR decomposition in LS and its geometric interpretation. By exploiting this geometric interpretation, the main order-recursion algorithm is developed in Section III.

II. INVERSE QR FACTORIZATION IN LEAST SQUARES

Consider the following general linear least squares (LS) problem. Let $\{\mathbf{x}_i(t)\}_{i=1}^n$, $y(t)$ be t-dimensional column vectors $\mathbf{x}_i(t) \equiv [\lambda^{t-1}x_i(1), \lambda^{t-2}x_i(2), \cdots, x_i(t)]^T$, $\mathbf{y}(t) \equiv [\lambda^{t-1}y(1), \lambda^{t-2}y(2), \cdots, y(t)]^T$, and let $\mathbf{X}^i(t) \equiv [\mathbf{x}_1(t), \mathbf{x}_2(t), \cdots, \mathbf{x}_i(t)]$ $(i = 1, 2, \cdots, n)$. $\lambda \in (0, 1]$ is a forgetting factor. Here, the superscript T indicates transpose, and t is taken to be the time index. Assume $\mathbf{X}^n(t)$ has rank n. The goal at time t is to find the i-dimensional column vector $\hat{\theta}_i(t)$ defined as

$$\hat{\theta}_i(t) \equiv \arg \min_{\theta} \|\mathbf{y}(t) - \mathbf{X}^i(t)\theta\|_2 \qquad (i = 1, 2, \cdots, n) \quad (1)$$

where $\| \bullet \|_2$ is the standard Euclidean norm. The set $\{\hat{\theta}_i(t)\}_{i=1}^n$ of LS coefficient vectors forms a set of candidate models that can be compared in order to select the most appropriate model. To have suitable criteria for comparison, it is also desirable to have available $\mathbf{e}_i(t) \equiv \mathbf{y}(t) - \mathbf{X}^i(t)\hat{\theta}_i(t)$ and $\alpha_i(t) \equiv \mathbf{e}_i^T(t)\mathbf{e}_i(t)$. $\mathbf{e}_i(t)$ is thus the residual error vector associated with the ith-order LS problem in (1), and $\alpha_i(t)$ is the residual error energy.

It is assumed that the LS modeling is to be done time recursively. In other words, at time $t+1$, the new "input" data $\{x_1(t+1), x_2(t+1), \cdots, x_n(t+1)\}$ and "output" data $y(t+1)$ becomes available, and $\mathbf{X}^n(t+1)$ and $\mathbf{y}(t+1)$ are given by

$$\mathbf{X}^n(t+1) = \begin{bmatrix} \lambda\mathbf{X}^n(t) \\ x_1(t+1) \quad x_2(t+1) \quad \cdots \quad x_n(t+1) \end{bmatrix}$$

and

$$\mathbf{y}(t+1) = [\lambda\mathbf{y}^T(t) \quad y(t+1)]^T.$$

Consider the QR factorization of $\mathbf{X}^n$, i.e., $\mathbf{X}^n = \mathbf{QR}$, where $\mathbf{Q} \equiv [\mathbf{q}_1 \quad \mathbf{q}_2 \quad \cdots \quad \mathbf{q}_n]$ is a $t \times n$ matrix with orthogonal columns, and $\mathbf{R}$ is an $n \times n$ upper triangular matrix with ones on the diagonal. The time index t has been dropped for notational simplicity. Note that $\mathbf{Q}$ and $\mathbf{R}$ are more commonly defined so that $\mathbf{Q}$ has orthonormal columns, and the diagonal elements of $\mathbf{R}$ are positive but not necessarily one.

In the context of the LS problem (1), $\mathbf{Q}$ and $\mathbf{R}^{-1}$ have an important interpretation that will subsequently be exploited to develop

an efficient method of solving (1) for all of the various order models. Define $\mathbf{J} \equiv \mathbf{Q}^T \mathbf{Q} \equiv \operatorname{diag}(\{J_i\}_{i=1}^n)$ and

$$\mathbf{W} \equiv \mathbf{R}^{-1} \equiv \begin{bmatrix} 1 & w_2 & w_3 & \cdots & w_n \\ & 1 & & & \\ & & 1 & & \\ \mathbf{0} & & & \ddots & \\ & & & & 1 \end{bmatrix}.$$

Here, $\mathbf{J}$ is a diagonal matrix, and $\mathbf{w}_i$ is a column vector of length $i-1$. It is well known (see, e.g., [8, p. 50]) that the following relationships between $\mathbf{W}$, $\mathbf{Q}$, and the LS problem holds:

$$\mathbf{w}_i = -\arg \min_{\mathbf{w}} \|\mathbf{x}_i - \mathbf{X}^{i-1}\mathbf{w}\|_2, \quad \mathbf{q}_i = \mathbf{x}_i + \mathbf{X}^{i-1}\mathbf{w}_i,$$

and

$$J_i = \mathbf{q}_i^T \mathbf{q}_i. \tag{2}$$

holds. In other words

$\mathbf{w}_i$ negative of the LS coefficient vector in projecting $\mathbf{x}_i$ onto the column space of $\mathbf{X}^{i-1}$;

$\mathbf{q}_i$ error vector associated with the LS projection;

J_i residual error energy.

There exist numerous algorithms for time updating $\mathbf{J}$, the last row of $\mathbf{Q}$, and $\mathbf{R}^{-1}$ in the inverse QR factorization of $\mathbf{X}^n$ (e.g., [9] or Bierman's $\mathbf{UDU}^T$ algorithm [8]). The inverse QR factorization algorithms can also be used to directly obtain $\hat{\theta}_n$ in (1) as follows [9]. Suppose that instead of factoring $\mathbf{X}^n$, the inverse QR factorization of the augmented matrix $\mathbf{A} \equiv [\mathbf{X}^n \ \mathbf{y}]$ was available. By analogy with (2)

$\mathbf{w}_{n+1}$;   exactly $-\hat{\theta}_n$;

$\mathbf{q}_{n+1}$ error vector;

J_{n+1} residual error energy.

From (2), it is apparent that in some sense, order recursiveness is built into the inverse QR factorization algorithms. The LS coefficients, residual errors, and residual error energies for the various order input prediction problems of projecting $\mathbf{x}_i$ onto the column space of $\mathbf{X}^{i-1}$ $(i = 2, 3, \cdots, n)$ are automatically available. In RLS applications such as system identification and parametric spectral estimation, where the order of the model is unknown *a priori* or is time varying, what are needed, instead, are the LS solutions to (1) for projecting $\mathbf{y}$ onto the column space of $\mathbf{X}^i$ $(i = 1, 2, \cdots, n)$. The following section presents a computationally efficient algorithm for accomplishing this.

III. OBTAINING THE LS COEFFICIENTS VIA ORDER DOWNDATING

The method presented in this section for obtaining $\{\hat{\theta}_i\}_{i=1}^n$ relies on the inverse QR factorization of $\mathbf{A}$ after its columns have been rearranged in the manner described in the following paragraphs. Define, for $1 \leq i \leq n$, $\mathbf{A}^i \equiv \mathbf{A}^{i+1}\mathbf{P}^{i+1}$ with $\mathbf{A}^{n+1} \equiv \mathbf{A} \equiv [\mathbf{X}^n \ \mathbf{y}]$, where $\mathbf{P}^{i+1}$ is the permutation matrix exchanging the ith and $(i+1)$st columns of the matrix it operates on, i.e., $\mathbf{P}^{i+1}$ is the identity matrix with its ith and $(i+1)$st columns exchanged. It follows that $\mathbf{A}^i = [\mathbf{X}^{i-1} \ \mathbf{y} \ \mathbf{x}_i \ \mathbf{x}_{i+1} \cdots \mathbf{x}_n]$.

Let the following variables be defined similarly to those in Section II, except that the superscript "i" indicates they are with respect to the (inverse) QR factorization of $\mathbf{A}^i$ $(1 \leq i \leq n+1)$

$$\mathbf{A}^i \equiv \mathbf{Q}^i \mathbf{R}^i$$
$$\mathbf{J}^i \equiv (\mathbf{Q}^i)^T \mathbf{Q}^i \equiv \operatorname{diag}(\{J_j^i\}_{j=1}^{n+1})$$

and

$$\mathbf{W}^i \equiv (\mathbf{R}^i)^{-1} \equiv \begin{bmatrix} 1 & w_2^i & w_3^i & \cdots & w_{n+1}^i \\ & 1 & & & \\ & & 1 & & \\ \mathbf{0} & & & \ddots & \\ & & & & 1 \end{bmatrix}$$

where $\mathbf{w}_j^i$ is a column vector of length $j-1$. Let $\mathbf{q}_j^i$ $(j = 1, 2, \cdots, n+1)$ denote the jth column of $\mathbf{Q}^i$, $w_{k,j}^i$ denote the kth element of $\mathbf{w}_j^i$, and $q_{k,j}^i$ denote the kth element of $\mathbf{q}_j^i$.

By analogy with (2), the desired order recursions can be obtained from the inverse QR factorization of $\mathbf{A}^i$ $(1 \leq i \leq n+1)$. Specifically, LS coefficient vector, the error vector, and residual error energy in the projection of $\mathbf{y}$ onto the column space of $\mathbf{X}^i$ are given by

$$\hat{\theta}_i = -\mathbf{w}_{i+1}^{i+1}, \quad \mathbf{e}_i = \mathbf{q}_{i+1}^{i+1}, \quad \text{and} \quad \alpha_i = J_{i+1}^{i+1}. \tag{3}$$

The following proposition provides an efficient method for determining $\mathbf{W}^i$, $\mathbf{Q}^i$, and $\mathbf{J}^i$, given $\mathbf{W}^{i+1}$, $\mathbf{Q}^{i+1}$, and $\mathbf{J}^{i+1}$. It is assumed that $\mathbf{W}^{n+1}$, $\mathbf{Q}^{n+1}$, and $\mathbf{J}^{n+1}$ are time updated from their values at the previous timestep using, e.g., the algorithm of [9].

Proposition:

$$\mathbf{Q}^i = \mathbf{Q}^{i+1}\tilde{\mathbf{Q}}^i,$$
$$\mathbf{W}^i = \mathbf{P}^{i+1}\mathbf{W}^{i+1}\tilde{\mathbf{Q}}^i$$

and

$$\mathbf{J}^i = (\tilde{\mathbf{Q}}^i)^T \mathbf{J}^{i+1} \tilde{\mathbf{Q}}^i \tag{4}$$

where

$$\tilde{\mathbf{Q}}^i \equiv \begin{bmatrix} \mathbf{I}_{i-1} & & & \mathbf{0} \\ & -w_{i,i+1}^{i+1} & g^i & \\ & 1 & h^i & \\ \mathbf{0} & & & \mathbf{I}_{n-i} \end{bmatrix}$$

$$g^i \equiv \frac{J_{i+1}^{i+1}}{J_{i+1}^{i+1} + (w_{i,i+1}^{i+1})^2 J_i^{i+1}}$$

and

$$h^i \equiv \frac{w_{i,i+1}^{i+1} g^i J_i^{i+1}}{J_{i+1}^{i+1}}. \tag{5}$$

Proof: Define $\mathbf{G} \equiv \mathbf{Q}^{i+1}\tilde{\mathbf{Q}}^i$ and $\mathbf{H} \equiv \mathbf{P}^{i+1}\mathbf{W}^{i+1}\tilde{\mathbf{Q}}^i$. Then, $\mathbf{GH}^{-1} = \mathbf{Q}^{i+1}\tilde{\mathbf{Q}}^i(\tilde{\mathbf{Q}}^i)^{-1}\mathbf{R}^{i+1}\mathbf{P}^{i+1} = \mathbf{Q}^{i+1}\mathbf{R}^{i+1}\mathbf{P}^{i+1} = \mathbf{A}^{i+1}\mathbf{P}^{i+1} = \mathbf{A}^i$. By the uniqueness of the QR factorization, the proof will be complete if we show that the columns of $\mathbf{G}$ are orthogonal and $\mathbf{H}$ is upper triangular with ones on the diagonal. That the columns of $\mathbf{G}$ are orthogonal follows from

$$\begin{aligned} \mathbf{G}^T\mathbf{G} &= (\tilde{\mathbf{Q}}^i)^T (\mathbf{Q}^{i+1})^T \mathbf{Q}^{i+1} \tilde{\mathbf{Q}}^i \\ &= (\tilde{\mathbf{Q}}^i)^T \mathbf{J}^{i+1} \tilde{\mathbf{Q}}^i \\ &= \operatorname{diag}\Big(J_1^{i+1}, \cdots, J_{i-1}^{i+1}, J_i^{i+1}(w_{i,i+1}^{i+1})^2 + J_{i+1}^{i+1}, J_i^{i+1}(g^i)^2 \\ &\quad + J_{i+1}^{i+1}(h^i)^2, J_{i+2}^{i+1}, \cdots, J_{n+1}^{i+1}\Big) \end{aligned} \tag{6}$$

which also gives the expression for $\mathbf{J}^i$. The last equality follows by simple algebra after substituting in the expressions in (5). That $\mathbf{H}$ is upper triangular with ones on the diagonal can be easily verified by substituting (5) into the definition of $\mathbf{H}$, completing the proof.

The previous proposition and the preceding paragraphs provide a method for obtaining a solution to all of the various order LS problems in (1). Starting with the time update of $\mathbf{W} (\equiv \mathbf{W}^{n+1})$, $\mathbf{J} (\equiv \mathbf{J}^{n+1})$, and the last row of $\mathbf{Q} (\equiv \mathbf{Q}^{n+1})$, the solution to the nth-order LS problem is immediately available from

(3) with $i = n$. Only the last element of e_n, which represents the nth-order output prediction error $(y(t) - [x_1(t) \ x_2(t) \ \cdots \ x_n(t)]\hat{\theta}_n(t))$ at the current time, is available in the time-recursive situation. The solution to all of the lower order LS problems can then be calculated time recursively by order downdating via the above proposition.

The order downdating can be efficiently carried out by noting the following. The solution to the ith-order LS problem is given by the $(i+1)$st columns of $\mathbf{W}^{i+1}$, $\mathbf{J}^{i+1}$, and $\mathbf{Q}^{i+1}$. It is not necessary to order downdate any of the columns to the right of the $(i+1)$st column since they are not needed in the subsequent lower order recursions. In addition, the first i columns of $\mathbf{Q}^{i+1}$, $\mathbf{W}^{i+1}$, and $\mathbf{J}^{i+1}$ are unchanged from those of $\mathbf{Q}$, $\mathbf{W}$, and $\mathbf{J}$. Consequently, the portion of (4) necessary to obtain all the lower order LS solutions, starting with the highest order solution and working downwards, is

$$J_i^i = J_{i+1}^{i+1} + (w_{i,\,i+1}^{i+1})^2 J_i$$

$$\mathbf{w}_i^i = \begin{bmatrix} w_{1,\,i+1}^{i+1} \\ w_{2,\,i+1}^{i+1} \\ \vdots \\ w_{i-1,\,i+1}^{i+1} \end{bmatrix} - w_{i,\,i+1}^{i+1} \begin{bmatrix} w_{1,\,i} \\ w_{2,\,i} \\ \vdots \\ w_{i-1,\,i} \end{bmatrix}$$

and

$$q_{t,\,i}^i = q_{t,\,i+1}^{i+1} - w_{i,\,i+1}^{i+1} q_{t,\,i}. \tag{7}$$

Quantities that do not have superscripts in (7) are from the original time updated matrices $\mathbf{Q}$, $\mathbf{W}$, and $\mathbf{J}$. The expression for J_i^i follows from (6). $q_{t,\,i}^i$ denotes the last element of $\mathbf{q}_i^i$. $\{\hat{\theta}_i\}_{i=1}^n$, $\{\alpha_i\}_{i=1}^n$, and $\{e_i\}_{i=1}^n$ are then directly available from (3), where $e_i \equiv y(t) - [x_1(t) \ x_2(t) \ \cdots \ x_i(t)]\hat{\theta}_i(t)$ is the last element of $\mathbf{e}_i$.

The complete algorithm for time updating $\mathbf{W}$, $\mathbf{J}$, and the last row of $\mathbf{Q}$ and performing all the order downdating is given in Table I. All quantities in the table are scalars, and the time index t has been included for clarity. The time updating portion of Table I is the algorithm of [9], which is very similar to Bierman's algorithm, and requires $1.5n^2 + 6.5n + 2$ multiplications per timestep. The additional computational expense of the order downdating portion is $0.5n^2 + 2.5n - 3$ multiplications per timestep. Thus, the total computational expense for solving all of the LS problems of (1) is only $2n^2 + 9n - 1$ multiplications per timestep.

For comparison, consider the algorithm of [5]. Although it was developed for time-recursive batch LS, parts of it can be combined with the algorithm of [9] ([9] for time updating the backward predictor coefficients and [5] for the order recursions) to provide $\{\hat{\theta}_i\}_{i=1}^n$ and $\{\alpha_i\}_{i=1}^n$ time recursively. For the order recursions, we need [5, (29), (33), (53)–(55), (103), and Fig. 1], where (29), (33), and (103) must be modified for the case of time recursive LS with exponential weighting. The computational expense for the order recursions is $2n^2 + 3n - 5$ multiplications, which, when used with the algorithm of [9], results in a total computational expense of $3.5n^2 + 9.5n - 3$ multiplications per timestep. To the best of our knowledge, this represents the most computationally efficient of the existing algorithms for solving the above problem, requiring slightly less than double the computational expense of the algorithm of this correspondence.

IV. CONCLUSIONS

A new time and order recursive method for on-line tracking of system order and parameters using recursive least squares (RLS) has been developed. The method relies on the geometric properties of the QR and inverse QR decomposition in least squares problems and consists of two parts: a time updating portion, which uses existing inverse QR decomposition algorithms, and a new "order downdating"

TABLE I
ALGORITHMN FOR TIME RECURSIVELY SOLVING THE SET OF n LS PROBLEMS IN (1)

Time update W(t), J(t), and the last row of Q(t): (see [9])

initialize $x_{n+1}(t) = y(t)$, $\varepsilon_1(t) = q_{t,1}(t) = x_1(t)$, $\gamma_1(t) = 1$,
　　and $g_i(t) = 0$ $(i=1,2,\ldots,n)$

$J_1(t) = \lambda^2 J_1(t-1) + (\varepsilon_1(t))^2$

for $i = 2, 3, \ldots, n+1$

　$\varepsilon_i(t) = x_i(t) + \sum_{j=1}^{i-1} x_j(t) w_{j,i}(t-1)$

　$b_i(t) = q_{t,i-1}(t)/J_{i-1}(t)$

　$\gamma_i(t) = \gamma_{i-1}(t) - q_{t,i-1}(t)b_i(t)$

　$q_{t,i}(t) = \varepsilon_i(t)\gamma_i(t)$

　$J_i(t) = \lambda^2 J_i(t-1) + \gamma_i(t)(\varepsilon_i(t))^2$

　for $j = 1, 2, \ldots, i-2$ (skip for i=2)

　　$g_j(t) = g_j(t) + w_{j,i-1}(t)b_i(t)$

　end

　$g_{i-1}(t) = b_i(t)$

　for $j = 1, 2, \ldots, i-1$

　　$w_{j,i}(t) = w_{j,i}(t-1) - g_j(t)\varepsilon_i(t)$

　end

Perform order downdating:

initialize $J_{n+1}^{n+1}(t) = J_{n+1}(t)$, $q_{t,n+1}^{n+1}(t) = q_{t,n+1}(t)$

　　and $w_{i,n+1}^{n+1}(t) = w_{i,n+1}(t)$ $(i=1,2,\ldots,n)$

for $i = n, n-1, \ldots, 2$

　$J_i^i(t) = J_{i+1}^{i+1}(t) + (w_{i,i+1}^{i+1}(t))^2 J_i(t)$

　$q_{t,i}^i(t) = q_{t,i+1}^{i+1}(t) - w_{i,i+1}^{i+1}(t)q_{t,i}(t)$

　for $j = 1, 2, \ldots, i-1$

　　$w_{j,i}^i(t) = w_{j,i+1}^{i+1}(t) - w_{i,i+1}^{i+1}(t)w_{j,i}(t)$

　end

end

for $i = 1, 2, \ldots, n$, set:

$\hat{\theta}_i(t) = -\left[w_{1,i+1}^{i+1}(t) \ w_{2,i+1}^{i+1}(t) \ \cdots \ w_{i,i+1}^{i+1}(t)\right]^T$

$e_i(t) = q_{t,i+1}^{i+1}(t)$

$\alpha_i(t) = J_{i+1}^{i+1}(t)$

---

portion, which calculates the model parameters and residual error energies for an entire set of models with order varying from one to some prespecified maximum model order. The various order models can then be compared to select the most appropriate one, providing an efficient means of on-line tracking of system order and parameters. Advantages of the new method are its low computational expense per timestep and its generality, which is applicable to arbitrary RLS identification situations. The total computational expense ($2n^2 + 9n - 1$ multiplications per timestep) is only slightly larger than that of standard RLS ($1.5n^2 + 6.5n + 2$ multiplications per timestep using, e.g., Bierman's $\mathbf{UDU}^T$ algorithm) applied to the highest order model.

REFERENCES

[1] J. Shi and J. Ni, "Supervisory adaptive control for the structural vibration of a coordinate-measuring machine," *Int. J. Adv. Manuf. Technol.*, vol. 11, pp. 240–248, 1996.

[2] R. Du, M. Elbestawi, and B. Ullagaddi, "Chatter detection in milling based on the probability distribution of cutting force signal," *Mech. Syst. Signal Process.*, vol. 6, no. 4, pp. 345–362, 1992.

[3] S. L. Marple, Jr., "Efficient least squares FIR system identification," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-29, pp. 62–73, Feb. 1981.

[4] N. Kalouptsidis, G. Carayannis, D. Manolakis, and E. Koukoutsis, "Efficient recursive in order least squares FIR filtering and prediction," *IEEE Trans. Acoust., Speech, Signal Processing* vol. ASSP-33, pp. 1175–1187, Oct. 1985.

[5] G. Glentis and N. Kalouptsisis, "Efficient multichannel FIR filtering using a single step versatile order recursive algorithm," *Signal Process.,* vol. 37, pp. 437–462, 1994.

[6] B. Porat and T. Kailath, "Normalized lattice algorithms for least-squares FIR system identification," *IEEE Trans. Acoust., Speech, Signal Processing* vol. ASSP-31, pp. 122–128, Feb. 1983.

[7] X. Yu and Z. He, "Efficient block implementation of exact sequential least-squares problems," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. 36, pp. 392–399, Mar. 1988.

[8] N. Kalouptsidis and S. Theodoridis, Eds., *Adaptive System Identification and Signal Processing Algorithms.* Englewood Cliffs, NJ: Prentice-Hall, 1993.

[9] H. Sakai, "Recursive least-squares algorithms of modified Gram–Schmidt type for parallel weight extraction," *IEEE Trans. Signal Processing,* vol. 42, pp. 429–433, Feb. 1994.

On the Implementation of Nonseparable Two-Dimensional Haar Wavelet Transforms

Patrick Lenders and Anne Sjöström

Abstract— We introduce a transformation to localize the equations defining the successive levels of the Mallat pyramid for two-dimensional (2-D) Haar wavelets. We propose a methodology for implementing these wavelet transforms in parallel architectures like systolic arrays. More specifically, we show that there is a perfect match between the wavelet algorithms and the multiphase multirate array (MPRA) architectures.

Index Terms— Multirate arrays, systolic arrays, VLSI synthesis, wavelets.

I. INTRODUCTION

In [3], Gröchenig and Madych present Haar bases for n-dimensional wavelets. Some of these wavelets are *nonseparable*, which is a characteristic that has advantages over their separable counterparts (cf. for example [4]). In the present paper, we propose a methodology for implementing two-dimensional (2-D) Haar nonseparable wavelets. We propose specific implementations that generalize the Mallat pyramid algorithm (cf. [8]) and whose buffering and delay requirements are independent of the size of the data set. The key idea of the present paper is to localize the equations defining the successive levels of the pyramid to enable the implementation in parallel architectures like systolic arrays.

There is a large body of research on the systolic implementation of 2-D wavelet transforms (cf., for example, [1], [7], and [10]), but to our knowledge, very little work has been done in the area of the parallel implementation of *nonseparable* wavelet transforms.

II. MULTIRESOLUTION ANALYSIS

Wavelet bases can be constructed through multiresolution analysis (MRA, cf., for example, [3]). To define an MRA, we will use the notion of *dilation*. A linear transformation A is a dilation if A leaves Z^n invariant and has eigenvalues λ_i such that $|\lambda_i| > 1$. These properties imply that $|\det A| = q \geq 2$. We will also use the notion of *translation*. The translator operator τ_y is defined by $\tau_y f(x) = f(x - y)$.

An MRA associated with (Z^n, A) is an increasing family $\ldots \subset V_{j-1} \subset V_j \subset V_{j+1} \ldots (j \subset Z)$ of closed subspaces of $L^2(R^n)$ with the following properties:

- $\bigcup_{j \in Z} V_j$ is dense in $L^2(R^n)$, and $\bigcap_j V_j = \{0\}$.
- $f(x) \in V_j$ if and only if $f(Ax) \in V_{j+1}$.
- V_0 is invariant under τ_γ for all γ in Z^n.
- There exists a function $\phi \in V_0$ (the *scaling* function) such that $\{\tau_\gamma \phi, \gamma \in Z^n\}$ is a complete orthonormal basis for V_0.

Let $q = |\det(A)|$. It can be shown that there exist $(q - 1)$ wavelet functions $\psi_i \in L^2(R^n)$ such that their translates $\{\tau_\gamma \psi_i \mid i \in [1, 2, \ldots, q - 1], \gamma \in Z^n\}$ form an orthonormal basis in W_0. Gröchenig and Madych [3] present a method to build a set of $(q - 1)$ wavelet functions from the scaling function.

We will present a generalization of the Haar wavelet in two dimensions. In this case, the scaling function is the characteristic function of a measurable set Q. The characteristic function of a bounded measurable set Q is the scaling function of an MRA associated with (Z^n, A) if and only if the set Q is such that (cf. [3]) we have the following.

1) $Q \cap (Q + k) \simeq 0, \forall k \neq 0,$ and $k \in Z^n,$
2) There is a collection of q lattice points k_i that are representatives of distinct cosets in Z^n/AZ^n such that $AQ \simeq \bigcup_{i=1}^q (k_i + Q)$.
3) $\bigcup_{k \in Z^n} (Q + k) \simeq R^n.$

III. LOCALIZATION

Let us assume that ϕ is a scaling function for an MRA associated with (Z^n, A). We are interested in the case where the scaling function is the characteristic function of a measurable set Q. We will show how we can calculate the wavelet transform of a function $f(x)$ with a cascade of lowpass and highpass filters. Our algorithm is a generalization of Mallat's pyramid algorithm. If $f(x)$ is in subspace V_r, we have $f(x) = \sum_{k \in Z^n} a_{r,k} \phi(A^r x - k)$. We multiply by $\phi(A^r x - k)$ and integrate to obtain

$$a_{r,k} = q^r \int f(x) \phi(A^r x - k) \, dx = q^r \int_{A^r x - k \in Q} f(x) \, dx = f_r(k)$$

where we define $f_r(k)$ as the average value of $f(x)$ when $x \in A^{-r}(Q + k)$. We are interested in the 2-D case ($n = 2$). At the finest level of detail (for example, $r = 0$), a computer image is composed of pixels. Following [2], we define a grey-scale image of finite resolution as a map $\mu : P \to [0, 1]$, where P is the set of pixels, and $\mu(k)$ is interpreted as the grey tone of the pixel k. We postulate that the function $\mu(k)$ is the function $f_0(k)$, which is the average of the function $f(x)$ (the image) for $x \in Q + k$. In other words, the shape of the pixel approximates the set Q. The wavelet transform of a function $f(x)$ is the set of values $b_{i,j,k}$ such that

$$f(x) = \sum_{i,j,k} b_{i,j,k} \psi_j(A^i x - k).$$