Order-Recursive RLS Laguerre Adaptive Filtering

Ricardo Merched, Student Member, IEEE, and Ali H. Sayed, Senior Member, IEEE

Abstract—This paper solves the problem of designing recursive-least-squares (RLS) lattice (or order-recursive) algorithms for adaptive filters that do not involve tapped-delay-line structures. In particular, an RLS–Laguerre lattice filter is obtained.

Index Terms—Laguerre network, lattice filter, order-recursive filter, regularized least-squares, RLS algorithm.

I. INTRODUCTION

This paper solves the problem of designing recursive-least-squares (RLS) lattice algorithms for adaptive filters that do not involve tapped-delay-line structures. As is well-known, all the derivations that are available so far in the literature for RLS order-recursive filters are based on the assumption of regression vectors with shift structure (see, e.g., [1]–[6]). The resulting filters are therefore not applicable to situations that involve other filter structures, such as Laguerre-based networks, where successive regression vectors are not shifted versions of each other.

In recent works [7], [8], it was shown that fast fixed-order RLS algorithms can be derived for certain more general structures in the regression vectors, other than the shift structure. These extended fast array methods turn out to be generalizations of earlier well-known fast transversal schemes for tapped-delay lines known as the fast a posteriori error sequential technique (FAEST) [9] and the fast transversal filter algorithm (FTF) [10]. An example of the usefulness of these extensions, it was recently shown in [11] that the regression vectors that arise in a Laguerre-based network satisfy the structural conditions of [7] and that, therefore, an efficient fixed-order RLS scheme for updating the coefficients of a Laguerre-based adaptive filter can indeed be derived.

These results motivate us to pursue here the development of order-recursive, as opposed to fixed-order, adaptive algorithms for certain general filter structures, other than the conventional FIR structure. A consequence of our derivation will be the first RLS Laguerre-based lattice filter. While the existing RLS-based Laguerre solutions are all $O(M^2)$ algorithms (e.g., [12]), with $M$ being the order of the filter, the lattice filter of this paper offers an $O(M)$ solution for the exact same problem. This result is useful especially since it has been realized for some time that Laguerre networks offer superior modeling capabilities when compared with FIR networks at a reduced number of tap coefficients and with a guaranteed stable performance. This is in contrast with some adaptive IIR filter implementations that require stability monitoring. Excellent accounts of adaptive IIR filters and of the convenience of Laguerre-based filters can be found in [13]–[18]. An example of an application in echo cancellation appears in [19]. In particular, assuming stationary data, [18] proposes an LMS-like Laguerre-based lattice filter that is a generalization of the so-called gradient adaptive lattice (GAL) algorithm (see [5]).

In this paper, we derive an exact RLS Laguerre-based lattice algorithm. One advantage of the RLS-based algorithm, besides optimality, is that least-squares methods offer considerably superior convergence performance and lower misadjustment when compared with stochastic gradient solutions (see the simulation results in Section V and, in particular, Fig. 6).

We start our discussions in Section II with a brief review of the regularized least-squares problem, followed by derivations in Sections III of several order- and time-update relations. Although most of the expressions in this section may look familiar to readers acquainted with the theory of least-squares lattice filters, our presentation actually has three contributions. First, all expressions are derived without assuming any underlying structure in the regression vectors. The derivation of some of the relations derived in this section has been restricted in the literature to the case of shift structured data. Second, the derivation shows that it is possible to derive efficient order-recursive RLS filters, even for cases where the regression vectors do not possess shift structure. This is achieved by pointing out the exact variable whose update is intimately affected by the data structure. The derivation also shows what kinds of data structure lead to fast order-recursive filters. Finally, all order-recursive relations are derived by explicitly solving regularized least-squares problems from the start. In contrast, similar relations have always been derived in the literature without taking into account the need for regularization; this need is usually accounted for by initializing the lattice algorithm with certain nonzero initial conditions. Our arguments will show that these two ways of handling the initialization issue lead to different interpretations of some of the variables in the resulting algorithms.

We end our discussions with a derivation of the RLS Laguerre-lattice filter in Section IV. The algorithm is summarized in Table I, and its schematic representation is shown in Fig. 4. Comparing with the classical lattice filter of Fig. 2, we see that the new RLS Laguerre lattice filter still has a similar cascade structure. The main difference is that the delay blocks of Fig. 2 are replaced by a parallel lattice filter. This essentially amounts to replacing each delay element by a simple time-variant lattice section. We provide simulation results in Section V.
II. Regularized Least-Squares Problem

We first provide a brief review of the regularized least-squares problem. Thus, given a column vector \( y \in \mathbb{C}^{N+1} \) and a data matrix \( H \in \mathbb{C}^{(N+1) \times M} \), the exponentially-weighted least squares problem seeks the column vector \( w \in \mathbb{C}^M \) that solves

\[
\min_w [\mu \lambda^{N+1} ||w||^2 + (y - Hw)^*W(y - Hw)]
\]

where \( \mu \) is a scalar positive regularization parameter (usually small), and \( W = (\lambda^N \oplus \lambda^{N-1} \oplus \cdots \oplus 1) \) is a weighting matrix that is defined in terms of a forgetting factor \( \lambda \) satisfying \( 0 \ll \lambda < 1 \). The symbol \( \oplus \) denotes complex conjugate transposition.

The individual entries of \( y \) will be denoted by \( \{d(i)\} \), and the individual rows of \( H \) will be denoted by \( \{u_i\} \).

\[
y = \begin{bmatrix} d(0) \\ d(1) \\ \vdots \\ d(N) \end{bmatrix}, \quad H = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_N \end{bmatrix}.
\]

Let \( w_N \) denote the optimal solution of (1). It is given by

\[
w_N = (\mu \lambda^{N+1} I + H^*W H)^{-1} H^* W y
\]

where we introduced the coefficient matrix

\[
P_N = (\mu \lambda^{N+1} I + H^*W H)^{-1}.
\]

We further let \( \hat{y} \) denote the vector

\[
\hat{y} \triangleq H w_N.
\]

We will refer to \( \hat{y} \) as the regularized projection (or simply projection) of the observation vector \( y \) onto the range space of \( H \), which is written as \( R(H) \).

We also define two estimation error vectors: the a posteriori error vector

\[
e_N = y - H w_N
\]

and the a priori error vector

\[
e_N = y - H w_{N-1}
\]

where \( w_{N-1} \) is the solution to a least-squares problem similar to (1) with data up to time \( N - 1 \) (and with \( \mu \lambda^{N+1} \) replaced by \( \mu \lambda^{N} \)). The minimum cost of (1) will be denoted by \( \xi(N) \), and it is given by

\[
\xi(N) = y^* W e_N.
\]

The last entries of \( e_N \) and \( e_N \) are called the a posteriori and the a priori estimation errors at time \( N \), and they are given by

\[
e(N) = d(N) - u_N w_N, \quad e(N) = d(N) - u_N w_{N-1}.
\]

They are both related by a conversion factor

\[
e(N) = \gamma(N) e(N)
\]

where

\[
\gamma(N) = 1 - u_N P_N u_N^*.
\]

The well-known RLS algorithm allows us to update \( w_N \) recursively as follows:

\[
\gamma^{-1}(N) = 1 + \lambda^{-1} u_N P_{N-1} u_N^* \gamma(N)
\]

\[
g_N = \lambda^{-1} P_{N-1} u_N^* \gamma(N)
\]

\[
w_N = w_{N-1} + g_N e(N)
\]

\[
P_N = \lambda^{-1} P_{N-1} - g_N g_N^* \gamma^{-1}(N) g_N
\]

with \( w_{-1} = 0 \) and \( P_{-1} = \mu^{-1} I \). It also holds that \( g_N = P_{N} u_N^* \).

III. Order-Recursive Relations

We now derive several order-recursive relations. As mentioned in the introduction, we re-emphasize that the presentation in this section has three contributions. First, the arguments do not assume shift structure. Second, the derivation introduces and singles out a variable whose update is affected by data structure. Third, the derivation explicitly incorporates regularization.

Before proceeding, we should remark that since in the remainder of this paper we deal primarily with order-recursive least-squares problems, it becomes important to explicitly indicate the size of all quantities involved (in addition to a time index). For example, we will write \( u_{M,N} \) instead of \( u_N \) to indicate that it is a vector of order \( M \) that is computed by using data up to time \( N \). We will also write \( H_{M,N} \) instead of \( H \) to indicate that it is a matrix with row vectors of size \( M \) and with data up to time \( N \). Similarly, we write \( y_N \) instead of \( y \) and \( W_N \) instead of \( W \) so that problem (1) becomes

\[
\min_w [\mu \lambda^{N+1} ||w||^2 + (y_N - H_{M,N} w_M)^* W_N (y_N - H_{M,N} w_M)]
\]

and its solution is \( w_{M,N} \). In a similar vein, we will write

\[
(\hat{y}_{M,N} , \epsilon_{M,N} , \epsilon_{M,N} , \epsilon_{M,N} , \epsilon_{M,N} , \gamma_{M,N} , \xi_{M,N}).
\]

A. Order Updating

Assume (for simplicity of presentation) that \( M = 3 \), and consider the data matrix

\[
H_{3,N} = \begin{bmatrix}
u(0, 0) & u(0, 1) & u(0, 2) \\
u(1, 0) & u(1, 1) & u(1, 2) \\
u(2, 0) & u(2, 1) & u(2, 2) \\
u(N, 0) & u(N, 1) & u(N, 2) \\
\end{bmatrix}
\]

We may remark that without the factor \( \lambda^{N+1} \) in the cost (1), the above RLS recursions would not be accurate.
The (regularized) projection of $y_N$ onto $\mathcal{R}(H_{3,N})$ is given by [cf. (4)]

$$\hat{b}_{3,N} = H_{3,N} P_{3,N} H_{3,N}^* W_N y_N.$$  

Now, suppose that one more column is appended to $H_{3,N}$, i.e.,

$$H_{4,N} = [H_{3,N} \ x_{3,N}] \quad (10)$$

where

$$x_{3,N} = \begin{bmatrix} u(0, 3) \\ u(1, 3) \\ u(2, 3) \\ \vdots \\ u(N, 3) \end{bmatrix}. $$

The (regularized) projection of $y_N$ onto $\mathcal{R}(H_{4,N})$ is now

$$\hat{b}_{4,N} = H_{4,N} P_{4,N} H_{4,N}^* W_N y_N. \quad (11)$$

In order to relate both projections of the vector $y_N$, we note the following. The coefficient matrices $\{P_{3,N}, P_{4,N}\}$ are $3 \times 3$ and $4 \times 4$, respectively, and they are defined by [cf. (3)]

$$P_{3,N}^{-1} = (\mu \lambda^{N+1} I + H_{3,N} W_N H_{3,N})$$
$$P_{4,N}^{-1} = (\mu \lambda^{N+1} I + H_{4,N} W_N H_{4,N}).$$

They are therefore related via

$$P_{4,N}^{-1} = \begin{bmatrix} P_{3,N}^{-1} & H_{3,N} W_N x_{3,N} \\ x_{3,N}^T W_N H_{3,N} & \mu \lambda^{N+1} + x_{3,N}^T W_N x_{3,N} \end{bmatrix}. $$

Inverting both sides, we get

$$P_{4,N} = \begin{bmatrix} P_{3,N} & 0 \\ 0 & 0 \end{bmatrix} + \frac{1}{\mu \lambda^{N+1} + \xi_3^2(N)} \begin{bmatrix} -x_{3,N}^T W_N & -1 \end{bmatrix}$$

where $w_{3,N}^2$ is the solution to the least-squares problem:

$$\min_{w_3} [\mu \lambda^{N+1} ||w_3^2||^2 + (x_{3,N} - H_{3,N} u_{3,N})^T W_N (x_{3,N} - H_{3,N} u_{3,N})]$$

and $\xi_3^2(N)$ is the corresponding minimum cost. This problem projects $x_{3,N}$ onto $\mathcal{R}(H_{3,N})$. Let

$$b_{3,N} = x_{3,N} - H_{3,N} u_{3,N}$$

denote the resulting (backward) estimation error vector. Substituting (12) into (11), we find that the projections $\{\hat{b}_{4,N}, \hat{b}_{3,N}\}$ are related via

$$\hat{b}_{4,N} = \hat{b}_{3,N} + \frac{b_{3,N}^* W_N y_N}{\mu \lambda^{N+1} + \xi_3^2(N)} b_{3,N}. \quad (13)$$

Subtracting $y_N$ from both sides, we obtain a relation between the corresponding a posteriori estimation error vectors

$$e_{4,N} = e_{3,N} - \epsilon_3(N) b_{3,N} \quad (14)$$

where we define the scalar

$$\epsilon_3(N) \triangleq \frac{b_{3,N}^* W_N y_N}{\mu \lambda^{N+1} + \xi_3^2(N)}.$$  

We therefore derived an order-update relation (14) for the a posteriori error vectors. The recursion however depends on $b_{3,N}$. We are thus motivated to study the propagation of $b_{3,N}$ more closely.

**B. Backward Estimation Problem**

We start by partitioning $H_{3,N}$ of (10) into

$$H_{3,N} = [x_{0,N} \ H_2 \ x_{3,N}]$$

so that $H_{4,N}$ is now partitioned as

$$H_{4,N} = [x_{0,N} \ H_2 \ x_{3,N}] \quad (16)$$

Using arguments similar to those that led to the update equation (14) for $e_{4,N}$, it is straightforward to verify that $b_{3,N}$ can be obtained as follows:

$$b_{3,N} = b_{3,N} - \epsilon_3^2(N) f_{2,N} $$

where the scalar coefficient $\epsilon_3^2(N)$ is defined as

$$\epsilon_3^2(N) \triangleq \frac{f_{2,N}^* W_N x_{3,N} - \xi_3^2(N)}{\mu \lambda^{N+1} + \xi_2^2(N)} \quad (17)$$

and $f_{2,N}$ is the residual error that results from the solution of the least-squares problem

$$\min_{w_2} [\mu \lambda^{N+1} ||w_2^2||^2 + (x_{3,N} - H_{2,N} u_{3,N})^T W_N (x_{3,N} - H_{2,N} u_{3,N})]$$

whose minimum cost we denote by $\xi_2^2(N)$. This problem projects $x_{3,N}$ onto $\mathcal{R}(H_{2,N})$. Likewise, $b_{3,N}$ is the residual error that results from the solution of the least-squares problem

$$\min_{w_2} [\mu \lambda^{N+1} ||w_2^2||^2 + (x_{3,N} - H_{2,N} u_{3,N})^T W_N (x_{3,N} - H_{2,N} u_{3,N})]$$

whose minimum cost we denote by $\xi_2^2(N)$. This problem projects $x_{3,N}$ onto $\mathcal{R}(H_{2,N})$.

**C. Forward Estimation Problem**

By similar arguments, $f_{2,N}$ can be updated as follows:

$$f_{3,N} = f_{2,N} - \epsilon_3^2(N) \bar{b}_{2,N} $$

where $\epsilon_3^2(N)$ is defined as

$$\epsilon_3^2(N) \triangleq \frac{\bar{b}_{2,N}^* W_N x_{0,N}}{\mu \lambda^{N+1} + \xi_3^2(N)} \quad (18)$$

and

$$\bar{b}_{2,N} = b_{2,N} - \epsilon_3^2(N) f_{2,N} $$

where the scalar coefficient $\epsilon_3^2(N)$ is defined as
Note that we used $\delta_2^o(N)$ in the numerator of $\kappa_2^o(N)$ and $\delta_2(N)$ in the numerator of $\kappa_2^l(N)$ in (17) since it can be easily verified that
\[
[f_{2L}^o W_{N:2L}^o, N]^* = \tilde{b}_{2L}^o W_{N:2L}^o.
\] (19)

Summarizing, we have so far derived the following order-update relations for the error vectors $\{e_{M,N}, b_{M,N}, f_{M,N}\}$ (which are written here for a generic order $M$):
\[
\begin{align*}
\{e_{M+1,N} = e_{M,N} - \kappa_{N}(N)b_{M,N} \\
b_{M+1,N} = b_{M,N} - \kappa_{N}^b(N)f_{M,N} \\
f_{M+1,N} = f_{M,N} - \kappa_{N}^f(N)b_{M,N}.
\end{align*}
\]

We still need to derive a relation for $\tilde{b}_{M,N}$. We postpone this discussion to Section III-G due to its intrinsic dependence on data structure.

If we extract the last entries of the above vectors, we obtain the following relations:
\[
\begin{align*}
e_{M+1}(N) &= e_{M}(N) - \kappa_{M}(N)b_{M}(N) \\
b_{M+1}(N) &= b_{M}(N) - \kappa_{N}^b(M)f_{M}(N) \\
f_{M+1}(N) &= f_{M}(N) - \kappa_{N}^f(M)b_{M}(N)
\end{align*}
\]
where
\[
\begin{align*}
\kappa_{M}(N) &= \frac{\rho_{M}(N)}{\mu \lambda^{N+1} + \xi_{M}^o(N)} \\
\kappa_{N}^b(M) &= \frac{\delta_{M}(N)}{\mu \lambda^{N+1} + \xi_{M}^o(N)} \\
\kappa_{N}^f(M) &= \frac{\delta_{M}(N)}{\mu \lambda^{N+1} + \xi_{M}^o(N)},
\end{align*}
\]
which are needed in the evaluation of the (reflection) coefficients $\{\kappa_{M}(N), \kappa_{N}^b(M), \kappa_{N}^f(M)\}$. To do so, we first derive below a general update result. It is important to re-emphasize that this result will be independent of any data structure in the matrix $H_{M,N}$ (cf. [20]).

D. General Time-Update Result

Consider a generic data matrix of the form
\[
[x \ \overline{H} \ z]
\]
where $x$ and $z$ are column vectors, and $\overline{H}$ is a matrix of appropriate dimensions. Define the weighted inner product
\[
\Delta = x^* W \tilde{z}
\]
where $\tilde{z}$ is the residual vector from a regularized projection of $z$ onto $\mathcal{R}(\overline{H})$, namely, $\tilde{z} = z - \overline{H} w_z$, where $w_z$ is obtained by solving
\[
\min_{w} [\mu \lambda^{N+1} ||w||^2 + (z - \overline{H} w)^* W (z - \overline{H} w)]
\]
(21)
where, as before, $W = \text{diag}\{\lambda^N, \cdots, \lambda, 1\}$.

Now, assume that one more row is appended to the matrix (20), say
\[
\begin{bmatrix}
x \\
\alpha \\
h \\
\beta
\end{bmatrix} \mathcal{H} = \begin{bmatrix} x_1 \ \overline{H}_1 \ z_1 \end{bmatrix}
\]
and introduce the corresponding factor $\Delta_1 = x_1^* W_1 \tilde{z}_1$, where $W_1 = (\lambda \mathcal{W} \otimes 1)$. We would like to relate $\Delta_1$ and $\Delta$ (i.e., we would like to determine an order-update relation for $\Delta$).

As above, let $w_{21}$ denote the solution of a problem similar to (21) with $\{z, \overline{H}_1, W_1, \lambda^{N+2}\}$ replaced by $\{z_1, \overline{H}_1, W_1, \lambda^{N+2}\}$. Likewise, let $w_{21}$ denote the solution of a problem similar to (21) with $\{z, \overline{H}_1, W_1, \lambda^{N+2}\}$ replaced by $\{x_1, \overline{H}_1, W_1, \lambda^{N+2}\}$. Now, define the a posteriori errors
\[
\tilde{\alpha} = \alpha - h w_{21}, \quad \tilde{\beta} = \beta - h w_{21},
\]
and the conversion factor
\[
\gamma = 1 - h [\mu \lambda^{N+2} I + \overline{H}_1^* W_1 \overline{H}_1]^{-1} h^* \hat{\Delta} = 1 - h \overline{P}_1 h^*.
\]
From the definition of $\Delta_1$, we have
\[
\Delta_1 = x_1^* \alpha^* \begin{bmatrix} \lambda W & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{z} \\ \overline{H}_1 \end{bmatrix} - x_1^* \alpha^* \beta = (\lambda x^* W \tilde{z} + \alpha^* \beta - (\lambda x^* W \overline{H}_1 + \alpha^* h) w_{21}.
\]
(22)

The RLS recursion (6) allows us to relate $w_2$ and $w_z$ as
\[
w_{21} = w_z + \gamma^{-1} \overline{P}_1 h^* \tilde{\beta}.
\]
Substituting into (22), we obtain, after grouping terms
\[
\Delta_1 = \lambda \Delta + \tilde{\alpha}^* \tilde{\beta} + \gamma \Delta_2.
\]
(23)

E. Time-Update Relations

We can now use the general result (23) to derive updates for
\[
\{\delta_{M}(N), \rho_{M}(N), \xi_{M}^o(N), \xi_{M}^b(N), \xi_{M}^f(N)\}.
\]
Consider, for example, the quantity
\[
\delta_{M}(N) = x_{M+1}^* W_{N} \tilde{b}_{M,N}
\]
which appears in the numerator of $\kappa_{f}^f(N)$ [see (18) for $M = 2$]. We can partition $H_{M+2,N}$ as
\[
H_{M+2,N} = \begin{bmatrix} x_{0,N} & \overline{H}_{M,N} & x_{M+1,N} \\ u(N,0) & u(N,M+1) \end{bmatrix}
\]
and make the identifications $x_1 \rightarrow x_{0,N}, \ z_1 \rightarrow \tilde{b}_{M,N}, \ \overline{H}_1 = \overline{H}_{M,N}, \ \text{and} \ W_1 \rightarrow W_{N}$. It then follows that
\[
\delta_{M}(N) = \lambda \delta_{M}(N - 1) + \frac{f_{M}(N) \overline{H}_{M,N}}{\gamma_{M}(N)}
\]
(24)
where the conversion factor $\gamma_M(N)$ in this case is defined by
\[
\gamma_M(N) = 1 - u_{M,N}P_{M,N}\pi_M(N).
\]

$P_{M,N} = [\mu \lambda^{M+1} + H_{M,N}W_NH_{M,N}]^{-1}$.

In a similar vein, we can justify the following time-update equations:
\[
\begin{align*}
\rho_M(N) &= \lambda \rho_M(N-1) + \frac{\tilde{e}_M(N)\tilde{e}_M(N)}{\gamma_M(N)} \\
\xi_M(N) &= \lambda \xi_M(N-1) + \frac{[\tilde{b}_M(N)]^2}{\gamma_M(N)} \\
\delta_M(N) &= \lambda \delta_M(N-1) + \frac{[\tilde{f}_M(N)]^2}{\gamma_M(N)} \\
\eta_M(N) &= \lambda \eta_M(N-1) + \frac{[\tilde{b}_M(N)]^2}{\gamma_M(N)}
\end{align*}
\]

where the conversion factor $\gamma_M(N)$ is defined by
\[
\gamma_M(N) = 1 - u_{M,N}P_{M,N}\pi_M(N),
\]

$P_{M,N} = [\mu \lambda^{M+1} + H_{M,N}W_NH_{M,N}]^{-1}$.

\section*{F. Order-Update Relations}

We can order update the conversion factors as follows. Note that the last row of $H_{M+1,N}$ can be partitioned as
\[
[u_{M+1,N} u(N, M)]
\]
so that by multiplying a relation of the form (12) for $P_{M+1,N}$ by $u_{M+1,N}$ from the left and by its conjugate transpose from the right, we get
\[
\gamma_{M+1}(N) = \gamma_M(N) - \frac{[b_M(N)]^2}{\mu \lambda^{N+1} + \xi_M^b(N)}
\]

Similarly, we get
\[
\gamma_{M+1}(N) = \gamma_M(N) - \frac{[\tilde{b}_M(N)]^2}{\mu \lambda^{N+1} + \xi_M^b(N)}
\]

In addition, using the order-update relations for the vectors $b_{M+1,N}$ and $f_{M+1,N}$ and the defining relations
\[
\begin{align*}
\xi_M^f(N) &= x_0,W_N f_{M+1,N} \\
\xi_M^b(N) &= x_{M+1,N} W_N b_{M+1,N}
\end{align*}
\]

we obtain the following order recursions for the minimum costs $\xi_M^f(N)$ and $\xi_M^b(N)$:
\[
\begin{align*}
\xi_{M+1}^{f}(N) &= \xi_M^{f}(N) - \frac{[\delta_M(N)]^2}{\mu \lambda^{N+1} + \xi_M^f(N)} \\
\xi_{M+1}^{b}(N) &= \xi_M^{b}(N) - \frac{[\delta_M(N)]^2}{\mu \lambda^{N+1} + \xi_M^b(N)}
\end{align*}
\]

\section*{G. Significance of Data Structure}

Thus far, we have derived almost all the necessary recursions for the development of an adaptive lattice filter. All the results hold for arbitrary data structures. The only update missing is the one for the error sequence $\{\tilde{b}_M(N)\}$. As is shown schematically in Fig. 1 by the boxes with question marks, we need to know how to generate the errors $\{\tilde{b}_M(N)\}$. It is the update of these variables that is directly affected by data structure and it is the key to achieving a fast algorithm [by fast, we mean $O(M)$ operations per iteration for a filter of order $M$].

To illustrate this point, consider, as an example, the case of prewindowed input data with shift structure, e.g., for $M = 3$

\[
H_{4,N} = \begin{bmatrix}
u(0) & 0 & 0 & 0 \\
u(1) & u(0) & 0 & 0 \\
u(2) & u(1) & u(0) & 0 \\
u(3) & u(2) & u(1) & u(0) \\
\vdots & \vdots & \vdots & \vdots \\
u(N) & u(N-1) & u(N-2) & u(N-3)
\end{bmatrix}
\]

Then, any two successive columns $\{x_{i+1,N}, x_{i+2,N}\}$ of $H_{M,N}$ are related by the lower triangular shift matrix $Z$, i.e.,
\[
x_{i+1,N} = Zx_{i,N}
\]

so that the following always holds for all $M$:
\[
H_{M,N} = ZH_{M,N}, \quad ZH_{M,N} = \begin{bmatrix} 0 \\
H_{M,N-1} \end{bmatrix}.
\]

Using these relations in the definitions of $\tilde{b}_{M,N}$ and $b_{M,N}$, viz., in
\[
\begin{align*}
\tilde{b}_{M,N} &= x_{M+1,N} - H_{M,N}u_{M,N} \\
b_{M,N} &= x_{M,N} - H_{M,N}u_{M,N}
\end{align*}
\]

we can easily verify that
\[
\tilde{b}_{M,N} = [0]
\]

and hence
\[
\tilde{b}_{M,N} = b_{M,N}(N-1)
\]

which is a widely known result. In a similar vein, it will also hold that
\[
\xi_M^f(N) = \xi_M^f(N-1) \quad \text{and} \quad \gamma_M(N) = \gamma_M(N-1).
\]
These equalities eliminate the need for recursions (26) and (30), and the general lattice recursions of this paper collapse to the well-known tapped-delay-line lattice network depicted in Fig. 2. The corresponding regularized lattice equations are listed in Table I.

We may note that we have redefined the minimum cost variables in order to save addition operations. For example, we defined

\[ \zeta^f_M(N) \triangleq \mu \lambda^{N+1} + \xi^f_M(N), \]
\[ \zeta^b_M(N) \triangleq \mu \lambda^{N+1} + \xi^b_M(N). \]

Then, it is easy to verify that these new quantities satisfy recursions similar to those of \( \{ \xi^f_M(N), \xi^b_M(N) \} \), namely

\[ \zeta^f_M(N) = \lambda \zeta^f_M(N-1) + \frac{|f_M(N)|^2}{\gamma^f_M(N-1)} \]
\[ \zeta^b_M(N) = \lambda \zeta^b_M(N-1) + \frac{|b_M(N)|^2}{\gamma^b_M(N)} \]
\[ \zeta^f_{M+1}(N) = \zeta^f_M(N) - \frac{|d_M(N)|^2}{\zeta^f_M(N)} \]
\[ \zeta^b_{M+1}(N) = \zeta^b_M(N) - \frac{|d_M(N)|^2}{\zeta^b_M(N)} \]

but with the initial conditions \( \zeta^f_0(-1) = \zeta^b_0(-1) = \mu. \)

Observe that the variables \( \{ \zeta^f_M(N), \zeta^b_M(N) \} \) do not correspond to the exact values of the minimum costs for the backward and forward prediction problems. Only when \( \lambda < 1 \) and \( N \to \infty \), they tend to coincide with the actual values \( \{ \xi^f_M(N), \xi^b_M(N) \}. \)

Now, what if two successive columns of the input data matrix \( H_{M,N} \) are not shifted versions of each other as in (31) but are instead related by a more general matrix \( \Phi \)? Would it still be possible to derive a fast lattice algorithm? Interesting enough, the answer is positive. We demonstrate this fact in the next section by considering an important example. The result will show that it is possible to move beyond what has been developed so far in the literature for shift-structured data and to develop exact RLS lattice algorithms for more general data.

Table I

**CLASSICAL RLS-FIR ADAPTIVE LATTICE ALGORITHM**

**Initialization**

For \( M = 0 \) to \( M - 1 \) set:
\[ \mu \] is a small positive number.
\[ \delta^b_M(-1) = \rho^b_M(-1) = 0 \]
\[ \zeta^f_M(-1) = \zeta^b_M(-1) = \mu \]

For \( N \geq 0 \), repeat:
\[ \gamma^b_0(N) = 1 \]
\[ b_0(N) = f_0(N) = u(N) \]

For \( M = 0 \) to \( M - 1 \), repeat:
\[ \zeta^f_M(N) = \lambda \zeta^f_M(N-1) + \frac{|f_M(N)|^2}{\gamma^f_M(N-1)} \]
\[ \zeta^b_M(N) = \lambda \zeta^b_M(N-1) + \frac{|b_M(N)|^2}{\gamma^b_M(N)} \]
\[ \delta^b_M(N) = \lambda \delta^b_M(N-1) + \frac{|b_M(N)|^2}{\gamma^b_M(N)} \]
\[ \rho^b_M(N) = \lambda \rho^b_M(N-1) + \frac{|b_M(N)|^2}{\gamma^b_M(N)} \]
\[ \gamma^b_{M+1}(N) = \gamma^b_M(N) - \frac{|b_M(N)|^2}{\gamma^b_M(N)} \]
\[ \zeta^f_{M+1}(N) = \zeta^f_M(N) - \frac{|d_M(N)|^2}{\zeta^f_M(N)} \]
\[ \zeta^b_{M+1}(N) = \zeta^b_M(N) - \frac{|d_M(N)|^2}{\zeta^b_M(N)} \]

IV. RLS LAGUERRE ADAPTIVE FILTERING

Consider the Laguerre-based model of Fig. 3, where
\[ L_0(z) = \frac{\sqrt{1-a^2}}{1-\alpha^2 \cdot z^{-1}} \quad \text{and} \quad L(z) = \frac{z^{-1} - a}{1-\alpha^2 \cdot z^{-1}}, \quad 0 < |a| < 1. \]

Note that \( L(z) \) is a first-order all-pass system and that, unlike a general IIR structure, the poles of the Laguerre-based model are fixed at \( \alpha. \) (The choice of \( \alpha \) can be optimized offline according to some criterion; see, e.g., [21].) The input to the Laguerre filter at time \( N \) is denoted by \( s(N) \), and the coefficients that combine the outputs of the successive sections \( \{ L_0(z), L(z) \} \) are denoted by \( \{ u_k \} \).

Now, consider the case of prewindowed input data (i.e., \( s(i) = 0 \) for \( i \leq 0 \) and zero initial conditions). Using the difference equations that define \( \{ L_0(z), L(z) \} \), it is possible to relate two successive columns of the data matrix in this case as

\[ x_{i+1,N} = \Phi_N x_i, N \]

This point serves as a good example of one of the issues we raised earlier, namely, by solving regularized problems from the start, we can clarify the significance and the meaning of the different variables that appear in the lattice recursions.
where $\Phi_N$ is an $(N + 1) \times (N + 1)$ lower triangular Toeplitz matrix of the form

$$
\Phi_N = \begin{bmatrix}
-a & -a & -a & \cdots \\
1 - a^2 & 1 - a^2 & -a & \cdots \\
a(1 - a^2) & 1 - a^2 & -a & \cdots \\
& \ddots & \ddots & \ddots \\
& & \ddots & \ddots \end{bmatrix}.
$$

(34)

Of course, it also holds that

$$
\mathbf{H}_{M,N} = \Phi_N \mathbf{H}_{M,N}.
$$

(35)

A. Exploiting Data Structure

Now, referring back to the definitions of the error vectors $\{b_{M,N}, \tilde{b}_{M,N}\}$ in (32), we see that we need to relate $\{u_{M,N}^e, u_{M,N}^f\}$. These vectors are given by

$$
u_{M,N}^e = P_{M,N} H_{M,N} W_x x_{M,N}
$$

$$
u_{M,N}^f = P_{M,N} H_{M,N} W_x x_{M,N+1}
$$

$$P_{M,N} = [\mu \lambda^{N+1} I + H_{M,N}^* W_x H_{M,N}]^{-1}
$$

$$P_{M,N} = [\mu \lambda^{N+1} I + H_{M,N}^* W_x H_{M,N}]^{-1}.
$$

(36)

For simplicity of notation, we are going to write $\{H, P, \Phi, W\}$ instead of $\{H_{M,N}, P_{M,N}, \Phi_N, W_N\}$ in the following derivation. Later, we will state the results with the correct subscripts. We will also assume in the sequel that $\lambda = 1$ so that $W = I$.

Using (35), we get

$$
u_{M,N}^e = \mu I + H^* \Phi_N^* \Phi H
$$

(36)

Now note that $\Phi_N^* \Phi$ is a rank-one modification of the identity matrix, namely, it satisfies

$$
\Phi_N^* \Phi_N = I - c_N c_N^*
$$

(37)

where

$$
c_N = \sqrt{1 - a^2} \begin{bmatrix}
a_N & a_{N-1} & \cdots & a
\end{bmatrix}^T.
$$

[We will also write $c$ instead of $c_N$ for simplicity.]

Substituting (37) into (36), we obtain

$$
u_{M,N}^e = [\mu I + H^* (I - c_N^* c_N) H]^{-1} H^* (I - c_N^* c_N) x_{M,N}
$$

$$= (P^{-1} - H^* c_N^* c_N H)^{-1} H^* (I - c_N^* c_N) x_{M,N}.
$$

(38)

For simplicity, define $v = c^* H$. Then, using the matrix inversion lemma, we get

$$
(P^{-1} - v^* v)^{-1} = P - \frac{P v^* v P}{v^* v - 1}.
$$

Substituting this expression into (38) leads to a desired relation between $\{u_{M,N}^e, u_{M,N}^f\}$:

$$
u_{M,N}^e = \Phi_N \mathbf{H}_{M,N}.
$$

(35)

If we now multiply by $\mathbf{H}_{M,N}$ from the left and subtract $x_{M+1,N}$, then using (32) and (35), we obtain

$$
\nu_{M,N} = \Phi b_{M,N} + c_N^* b_{M,N} - \Phi \hat{c}_{M,N}.
$$

(39)

where we defined the vector

$$
\hat{c}_{M,N} = \Phi_N P_{M,N} H_{M,N}^* c_{M,N}.
$$

This vector has the interpretation of being the regularized projection of $c_{M,N}$ onto $\mathcal{R}(H_{M,N})$.

In view of the above, we find that the last entry of $\nu_{M,N}$ is given by

$$
\nu_{M,N} = \mu \lambda^{N+1} I + H_{M,N}^* W_x H_{M,N}
$$

(40)

where $\phi_N$ is the last row of $\Phi_N$. The above relation involves four growing-length inner products on the right-hand side:

$$
\{\phi_N b_{M,N}, \phi_N c_{M,N}, \phi_N \hat{c}_{M,N}, \phi_N \hat{c}_{M,N}^*\}.
$$

We will show that the first two are related to each other, whereas the last two are also related to each other. This will follow as a result of the fact that $\phi_N$ and $c_N$ have similar forms. In this way, we shall need only to develop order-recursive updates for two of these inner-product terms.

To this end, first note that we can simplify (40) by exploiting the similarity between the vectors $\phi_N$ and $c_N$, viz.

$$
\phi_N = \sqrt{1 - a^2} \begin{bmatrix}
a_N & a_{N-1} & \cdots & a
\end{bmatrix}^T.
$$

to write

$$
\phi_N b_{M,N} = \frac{\sqrt{1 - a^2}}{a} c_N^* b_{M,N} - \frac{b_{M,N}}{a}
$$

$$
\phi_N \hat{c}_{M,N} = \frac{\sqrt{1 - a^2}}{a} c_N^* \hat{c}_{M,N} - \frac{\hat{c}_{M,N}}{a}.
$$

Substituting these expressions into (40), we obtain, after some manipulations

$$
\nu_{M,N} = \frac{1}{a} \left( c_N^* b_{M,N} \hat{c}_{M,N} - b_{M,N} \right).
$$

(38)
which contains only two inner products

\[
\tau_M(N) \triangleq c_N^k b_{M,N}, \quad \zeta_M(N) \triangleq 1 - c_N^k \hat{c}_{M,N}.
\]

With these definitions, we can write

\[
\hat{e}_M(N) = \frac{1}{a}(b_M(N) - \kappa_M^k(N) \hat{c}_M(N))
\]

where we defined

\[
\kappa_M^k(N) = \frac{\tau_M(N)}{\zeta_M(N)}
\]

and \( \hat{c}_M(N) = c_M(N) - \hat{c}_M(N). \) [Here, \( c_M(N) \) and \( \hat{c}_M(N) \) denote the last entries of \( c_{M,N} \) and \( \hat{c}_{M,N} \), respectively.]

Hence, all we really need to know is how to update the quantity \( \hat{c}_M(N) \) and the inner products \( \tau_M(N) \) and \( \zeta_M(N) \).

First, note that \( \hat{c}_{M,N} \) and \( \hat{c}_{M,N} \) can be order updated, in the same fashion as in (13) and (14), namely

\[
\hat{e}_{M+1,N} = \hat{e}_{M,N} + \frac{b_M(N) \tau_M(N)}{\mu + \zeta_M(N)}
\]

\[
\hat{e}_{M+1}(N) = \hat{e}_M(N) - \frac{b_M(N) \tau_M(N)}{\mu + \zeta_M(N)}
\]

\[
\hat{e}_M(N) = -\kappa_M(N) b_M(N)
\]

where we defined

\[
\kappa_M(N) \triangleq \frac{\tau_M(N)}{\mu + \zeta_M(N)}.
\]

In addition, multiplying the above recursion for \( \hat{e}_{M,N} \) by \( c_N^k \) from the left, we obtain

\[
c_N^k \hat{e}_{M+1,N} = c_N^k \hat{e}_{M,N} + \frac{c_N^k b_{M,N} \nu(N)}{\mu + \zeta_M(N)}
\]

and subtracting one from both sides, we get the following order-update recursion for \( \zeta_M(N) \):

\[
\zeta_{M+1}(N) = \zeta_M(N) - \frac{[\tau_M(N)]^2}{\zeta_M(N)}.
\]

From (39), we can derive an alternative recursion for \( \xi_M(N) \).

To see this, we multiply (39) by \( x_{M+1,N}^* \) from the left and get

\[
\xi_{M+1}(N) = x_{M+1,N}^* \Phi \dot{b}_{M,N} + \kappa_M(N) x_{M,N}^* \Phi \dot{c}_{M+N}.
\]

Now, using (37) and the substitutions \( x_{M,N}^* \dot{b}_{M,N} = \xi_M(N) \), \( \dot{c}_{M,N} = \tau_M(N) \), and \( \dot{c}_{M,N} = 1 - \dot{c}_M(N) \), we get

\[
\xi_{M+1}(N) = \xi_M(N) - \frac{\tau_M(N)}{\zeta_M(N)} x_{M,N}^* \dot{c}_{M,N}.
\]

However, similar to (19), it holds that \( x_{M,N}^* \dot{c}_{M,N} = \dot{b}_{M,N}^\mu \cdot \dot{c}_{M+N} \)

so that we arrive at the recursion

\[
\xi_{M+1}(N) = \xi_M(N) - \frac{[\tau_M(N)]^2}{\zeta_M(N)}
\]

B. Final Time-Update Recursions

Finally, it only remains to determine a recursive relation for \( \tau_M(N) \). This step requires more effort. We start by noting that we can write \( \tau_M(N) \) in the form

\[
\tau_M(N) = \sqrt{1 - a^2} [1 \ 1 \ \cdots \ 1] A_N \dot{b}_{M,N}
\]

where \( A_N = \text{diag}(a^N, a^{N-1}, \ldots, a, 1) \). Observe that in so doing, we expressed \( \tau_M(N) \) as the product of a constant vector by a diagonal weighting matrix \( A_N \). Now, in order to update \( \tau_M(N) \) to \( \tau_{M+1}(N) \), we need to generalize our earlier result on the update of \( \Delta \) to \( \Delta_1 \) in Section III-D.

Thus, consider again a generic data matrix of the form

\[
\begin{bmatrix} x & H & z \end{bmatrix}
\]
and introduce the weighted inner product $\Delta = x^T \Lambda x$, for some positive-definite diagonal matrix $\Lambda$. Here, $\bar{z}$ denotes the residual vector from a regularized projection of $z$ onto $\bar{H}$, namely, $\bar{z} = z - \bar{H}w$, where $w$ is obtained by solving

$$\min_w [\mu \lambda^{N+1} |w|^2 + (z - \bar{H}w)^* W(z - \bar{H}w)]$$

(41)

where, as before, $W = \text{diag}\{ \lambda^N, \cdots, \lambda, 1 \}$. Note that now, however, the definition of $\Delta$ involves a center matrix $\Lambda$ that is distinct from $W$. Now, consider the extended matrix

$$\begin{bmatrix} x & \bar{H} & \bar{z} \\ \alpha & h & \beta \end{bmatrix} \rightarrow \begin{bmatrix} x_1 & \bar{H}_1 & \bar{z}_1 \end{bmatrix}$$

and introduce the corresponding factor $\Delta_1 = x_1^T \Lambda_1 \bar{z}_1$, where $\Lambda_1$ is related to $\Lambda$ via $\Lambda_1 = (\eta \Lambda \oplus 1)$, for some $\eta$. We again would like to relate $\Delta_1$ and $\Delta$ (i.e., we would like to determine an order-update relation for $\Delta$).

As above, let $w_2$ denote the solution of a problem similar to (41) with $\{ z, \bar{H}, W, \lambda^{N+1} \}$ replaced by $\{ z_2, \bar{H}_1, (\lambda W \oplus 1), \lambda^{N+2} \}$. Likewise, let $w_2$ denote the solution of a problem similar to (41) with $\{ z_2, \bar{H}_1, W, \lambda^{N+1} \}$ replaced by $\{ x_2, \bar{H}_1, (\lambda W \oplus 1), \lambda^{N+2} \}$. In addition, define the a posteriori error

$$\hat{\beta} = \beta - hw_2.$$

Then, an argument similar to that in Section III-D will show that

$$\Delta_1 = \eta \Delta + \left[ \eta - \frac{h}{\lambda} \bar{H} \Lambda \bar{H}^T \Lambda x \right]^* \hat{\beta}$$

(42)

where $P$ is defined by

$$P = \left( \mu \lambda^{N+1} + \bar{H}^T W \bar{H} \right)^{-1}.$$

Returning to the update of $\tau_M(N)$, we can now make the following identifications:

$$\begin{align*}
\tau_1 & = \sqrt{1 - \delta^2} \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}^T \\
\alpha & = \sqrt{1 - \delta^2}, \quad \eta = a, \quad \lambda = 1 \\
\bar{z}_1 & = b_{M,N}, \quad W_1 = I, \quad \Lambda_1 = \Lambda_N \\
\eta \lambda x & = \sqrt{1 - \delta^2} \begin{bmatrix} a^N & a^{N-1} & \cdots & a \end{bmatrix}^T
\end{align*}$$

that is, $\eta \lambda x$ corresponds to the top entries of $C_N$. In this case, the term $\eta h \bar{H} \Lambda \bar{H}^T \Lambda x$ would correspond to the (regularized) estimate of $\alpha$ that is based on the prior data in $\bar{H}$. Then, the difference $\alpha - \eta h \bar{H} \Lambda \bar{H}^T \Lambda x$ becomes equal to the a priori error in estimating $\alpha$, which can be transformed to the a posteriori error $\hat{\alpha}$ by means of the conversion factor $\gamma_M(N)$. This leads to the desired update equation

$$\tau_M(N + 1) = \tau_M(N) + \frac{\hat{\alpha}_M(N) b_{M,N}}{\gamma_M(N)}$$

Using similar arguments and the fact that

$$\begin{align*}
\zeta_M(N) & = [1 - \zeta_M(N)] - e_N^* \hat{z}_{M,N} = a^{N+1} - e_N^* \hat{z}_{M,N} \\
\zeta^*_M(N) & = \zeta_M(N) - \frac{(\hat{\alpha}_M(N))^2}{\gamma_M(N)}
\end{align*}$$
we can also obtain the following time-update recursion for $\xi_M(N)$:

$$
\xi_M(N) = a^2 \xi_M(N-1) + \frac{|c_M(N)|^2}{\gamma_M(N)}
$$

Table III compares the computational cost of a Laguerre lattice filter of order $M$ with a shift-structured (FIR) lattice filter of order $M'$ with $\lambda = 1$ (recall that, in general, $M \ll M'$).

Comparing Fig. 4 with the conventional lattice structure of Fig. 2, we see that the new lattice filter is still fundamentally simple; the major modification is in the substitution of the delay blocks of Fig. 2 by a second lattice filter that runs in parallel. This, in effect, corresponds to replacing the delay blocks by simple time-variant lattice sections. We may also mention that the algorithm of Table II is based on propagating the a posteriori estimation errors. Alternative implementations that are based on a priori errors, or even on normalized errors, can be derived and will be pursued elsewhere [22]. This is in addition to array forms. Note also that in the listing of Table II, we employed time-updates for the variables $\{\xi_M(N), c_M(N), r_M(N)\}$; order-update relations are also possible and can be used.

V. SIMULATIONS

In order to illustrate the advantages of using a Laguerre-based adaptive lattice structure, we compare the performance of a sixth-order Laguerre lattice filter with a shift-structured RLS lattice implementation of order 500. For this purpose, we consider the same IIR system used in [18], viz.

$$
G(z) = \frac{0.0017z^{-1}(1+0.673z^{-1})}{(1-0.308z^{-1})(1-0.819z^{-1})(1-0.903z^{-1})}.
$$

The input signal is a first-order AR process, and the SNR at the output is 50 dB. The Laguerre pole is located at $a = 0.0978$, as in [18]. Fig. 5 shows the resulting learning curves that are obtained by averaging over 20 experiments. It is clear that the mean-square error of the Laguerre structure is considerably better during the training phase. As expected, the Laguerre network is better suited to modeling the IIR system, which is accomplished at significantly less computational burden.

We also compare the performance of the RLS-Laguerre algorithm with the corresponding gradient-Laguerre lattice algorithm proposed in [18]. We consider a system identification scenario where the unknown system to be identified is itself a Laguerre network of the same order $M = 6$ as the Laguerre adaptive filters. The input signal is simply white noise, and the Laguerre pole is fixed at $a = 0.5$. Fig. 6 shows the learning curves of both algorithms averaged over 1000 experiments. We see that the RLS-Laguerre lattice offers significant improvement in both convergence performance and misadjustment, as is expected for least-squares designs.

VI. CONCLUSIONS

We developed a framework for efficient order-recursive RLS adaptive filtering for input data that do not necessarily arise from tapped-delay lines. A special important case occurs in Laguerre-
The approach of this paper can be extended to other filter networks, other than the Laguerre structure, especially when differences of the form $W - \Phi W \Phi$ have low rank. In addition, we can also develop normalized versions, array versions, and lattice schemes with feedback. These extensions will be published elsewhere [22].

REFERENCES


Ricardo Merched (S’97) was born in Rio de Janeiro, Brazil. He received the B.S. and M.S. degrees in electrical engineering from Universidade Federal do Rio de Janeiro in 1995 and 1997, respectively. Since September 1997, he has been pursuing the Ph.D. degree with the Electrical Engineering Department, University of California, Los Angeles, working on different aspects of adaptive filter design, especially multirate adaptive filters and fast RLS Laguerre adaptive filters.

Ali H. Sayed (SM’99) received the Ph.D. degree in electrical engineering in 1992 from Stanford University, Stanford, CA.

He is Associate Professor of electrical engineering at the University of California, Los Angeles (UCLA). He has over 135 journal and conference publications, is co-author of the research monograph Indefinite Quadratic Estimation and Control (Philadelphia, PA: SIAM, 1999) and of the graduate-level textbook Linear Estimation (Englewood Cliffs, NJ: Prentice-Hall, 2000). He is also co-editor of the volume Fast Reliable Algorithms for Matrices with Structure (Philadelphia, PA: SIAM, 1999). He is a member of the editorial boards of the SIAM Journal on Matrix Analysis and Its Applications and the International Journal of Adaptive Control and Signal Processing and is Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING. He has served as Co-Editor of special issues of the journal Linear Algebra and Its Applications. He has contributed several articles to engineering and mathematical encyclopedias and handbooks. His research interests span several areas including adaptive and statistical signal processing, filtering and estimation theories, equalization techniques for communications, interplays between signal processing and control methodologies, and fast algorithms for large-scale problems.

Dr. Sayed has served on the program committees of several international meetings. He is also a recipient of the 1996 IEEE Donald G. Fink Award.