An FIR Cascade Structure for Adaptive Linear Prediction

Paolo Prandoni and Martin Vetterli

***Abstract*— An alternative structure for adaptive linear prediction is proposed in which the adaptive filter is replaced by a cascade of independently adapting, low-order stages, and the prediction is generated by means of successive refinements. When the adaptation algorithm for the stages is LMS, the associated short filters are less affected by eigenvalue spread and mode coupling problems and display a faster convergence to their steady-state value. Experimental results show that a cascade of second-order LMS filters is capable of successfully modeling most input signals, with a much smaller MSE than LMS or lattice LMS predictors in the early phase of the adaptation. Other adaptation algorithms can be used for the single stages, whereas the overall computational cost remains linear in the number of stages, and very fast tracking is achieved.**

I. Introduction

Least mean squares (LMS) and recursive least squares (RLS) are the two fundamental algorithms in the area of adaptive filtering; however popular and widespread, these algorithms in their simplest forms suffer from several drawbacks and limitations. Fundamentally, the convergence of LMS filters is marred by two main problems: the eigenvalue spread of the correlation matrix of the input signal and the coupling between modes of convergence. Eigenvalue spread results in nonuniform speed of convergence for the filter values; mode coupling results in nonmonotonic trajectories toward convergence for the filter coefficients and in propagation of the eigenvalue disparity effects among the different modes. RLS filters, on the other hand, provide a decoupled route to convergence; the algorithm, however, relies on the implicit or explicit computation of the inverse of the input signal's autocorrelation matrix. This not only implies a higher computational cost, but more importantly, it can lead to irrecoverable instability problems. Fast RLS algorithms [6]–[8] display a large sensitivity to numerical accuracy, especially when trying to track nonstationary signals or in the presence of noisy inputs, and in the nonstationary case, they can yield reduced performance gains in comparison with LMS [2]. Although some numerically stable RLS algorithms do exist, in practical signal processing applications, the robustness and, above all, the simplicity of LMS structures make the latter the favored choice. In order to improve on the normal LMS algorithm, alternative adaptive structures such as the lattice LMS and the frequency-domain LMS are designed to counteract mode coupling, albeit at the price of a greater misadjustment [3]. Prewhitening filters have also been proposed for applications such as time-delay estimation and system identification [4], [5] to reduce the consequences of eigenvalue spread.

In linear prediction problems, and especially in the case of nonstationary signals, final misadjustment is of lesser importance with respect to the tracking capabilities of the system. The speed of convergence in a linear predictor is, however, reduced by three common circumstances.

1) The eigenvalue spread is usually large due to the high correlation of the input signal.
2) High-order FIR predictors are needed to model ARMA inputs, and it can be shown that eigenvalue spread is a nondecreasing function of the filter length.
3) Long filters require a small step size, which in turn slows the adaptation process.

To counteract these problems, in this correspondence, we present an alternative structure for adaptive linear prediction based on least squares minimization in which the predictor is formed by a cascade of small-order *independently adapting* linear filters. In a cascade of short filters, larger step sizes are allowed, and the single stages are less affected by eigenvalue disparity effects; this results in a faster and more agile route to convergence.

Adaptive filtering in cascade form has been addressed previously for both FIR and IIR filters [9]–[11]; application to speech prediction has been detailed in [12]. In these references, the cascade realization is introduced mainly to obtain a predictor in factored form, which allows an easier stability check for the inverse filter. The focus of our contribution is, however, on the *independent adaptation* of the single stages (even though the advantages of the factored form remain). Independent adaptation results in a predictor that does *not* converge to the usual Wiener solution; nevertheless, the performance improvements over conventional LMS filters are such that we can justify its consideration in linear prediction problems.

II. The Cascade Structure

The general structure of the cascade for the one-step-ahead linear prediction problem is shown in Fig. 1. In the cascade, each of the M sections is an independently adapting FIR predictor of order $l_k, k = 1 \dots M$. Let $x_k(n)$ be the input to stage k, and let $e_k(n)$ be the corresponding prediction error; it is

$$e_k(n) = x_k(n) - \sum_{m=1}^{l_k} f_k^{(m)}(n) x_k(n-m) \tag{1}$$

where $f_k^{(1)}(n) \dots f_k^{(l_k)}(n)$ are the time-varying taps of the kth predictor. The cascade structure is such that $x_{k+1}(n) = e_k(n)$; $x_1(n) = x(n)$, where $x(n)$ is the signal to model. The global prediction error of the structure is the error of the last stage, $e_M(n)$. After convergence, $f_k^{(m)}(n) \to f_k^{(m)}$, and the global predictor transfer function can be expressed as

$$F(z) = \prod_{k=1}^{M} F_k(z) \tag{2}$$

where

$$F_k(z) = 1 - \sum_{m=1}^{l_k} f_k^{(m)} z^{-m}. \tag{3}$$

Among the several architectural variables of the system (number of stages, number of taps per stage, update algorithms), the most successful experiments have taken into account cascade structures of minimally sized stages using LMS or RLS-like adaptation.

Theoretical analysis of the cascade is difficult and will not be attempted here except for a low-order example. Computer simulations have, however, shown that the cascade is particularly effective during

Manuscript received October 15, 1996; revised February 27, 1998. The associate editor coordinating the review of this paper and approving it for publication was Dr. Phillip A. Regalia.

P. Prandoni is with the Laboratoire de Communication Audio Visuelle, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland (e-mail: prandoni@de.epfl.ch).

M. Vetterli is with the Electrical Engineering and Computer Science Department, University of California at Berkeley, Berkeley CA 94720 USA. He is also with the École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland (e-mail: vetterli@de.epfl.ch).

Publisher Item Identifier S 1053-587X(98)06114-5.

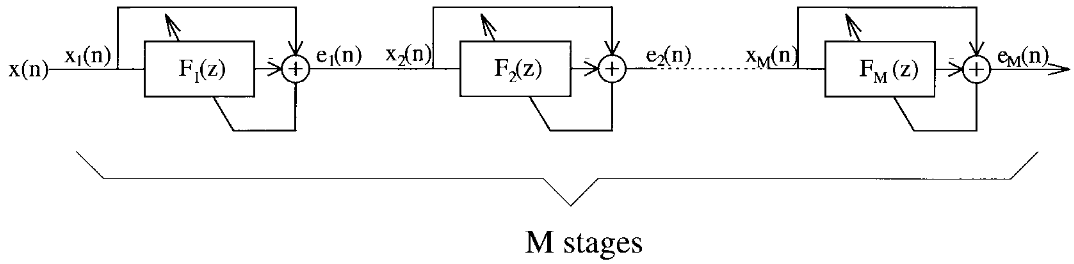


Fig. 1. General structure of the cascade predictor.

the initial, transient phase of the adaptation and displays a higher final misadjustment. We conjecture here (without formal proof) that the cascade operates a linear prediction in terms of *successive refinements*. Each stage is a short filter that "sees" its input signal through an autocorrelation matrix of small size; the linear prediction structure will therefore attempt to cancel the dominant mode of its input signal. If the signal to be modeled contains widely separate modes, the low-order approximation to the signal's autocorrelation will be sufficiently accurate; the first stage will therefore place its zeros close to the dominant poles of the AR model, the input to the second stage will identify the second dominant mode, and so on, with a successful progressive refinement strategy.

Care has to be exercised, however, for certain classes of signals such as signals with frequency spectra symmetric around $\pi/2$ (whose autocorrelation is zero at odd lags) or signals whose generating AR model contains clustered poles. In these cases, the small autocorrelation matrix implicit in a low-order filter would not allow the first stage to resolve the separate modes of the signal, leading to a predictor stuck "in balance" between different modes. Since there is no feedback path between stages, the first stage would pass a fundamentally unchanged signal to the second stage, and so on, thus propagating the problem across the whole cascade. In addition, due to the small order of the autocorrelation matrix, the cascade might not prove effective in noise-reduction applications such as line enhancement, in which the SNR conditions are markedly adverse. This suggests that the most useful application for the cascade is that of a "startup engine" for a global gradient predictor as in [9] or in a master–slave configuration as in [13].

A. Cascade of One-Tap Filters

The simplest form of the cascade predictor filter is composed of one-tap stages; of course, this structure will generally be inadequate for general input signals since the resulting prediction filter has only strictly real zeros. Surprisingly, however, the speed of convergence of the cascade filter with LMS adaptation is such that its initial MSE is usually smaller than those of equivalent-order LMS and lattice LMS predictors. In addition, it is instructive to analyze the behavior of this structure, which is partly tractable in analytical form.

In the cascade of one-tap filters, for each stage, $l_k = 1$; if we assume convergence toward the Wiener solution *for the single stages*, it is going to be

$$f_k(n) \to \frac{r_k(1)}{r_k(0)} = f_k \tag{4}$$

where $r_k(m)$ is the autocorrelation function for $x_k(n)$. Since $x_{k+1}(n) = e_k(n)$, the chain relation

$$r_{k+1}(m) = \left(1 + f_k^2\right) r_k(m) - f_k(r_k(m-1) + r_k(m+1)) \tag{5}$$

holds. The overall prediction error coincides with $e_M(n)$; defining $J_k = E[e_k^2(n)] = r_{k+1}(0)$ and using (5) for $m = 0$, we have $J_k = J_{k-1} - (r_k^2(1)/J_{k-1}) \leq J_{k-1}$ so that the final MSE is a nonincreasing function of the number of stages.

For a two-stage cascade filter, exact expressions for the filter values can be derived. This is, of course, a very simple example, but it is one in which direct comparison of the cascade filter and the Wiener solution is manageable. Let $F^* = [1 \ -c_1^* \ -c_2^*]^t$ be the prediction error filter as given by the Wiener solution of the linear prediction problem for a two-tap filter. The normal equations yield

$$c_1^* = \frac{r_x(1)[r_x(0) - r_x(2)]}{r_x^2(0) - r_x^2(1)} \tag{6}$$

$$c_2^* = \frac{r_x(0)r_x(2) - r_x^2(1)}{r_x^2(0) - r_x^2(1)}. \tag{7}$$

Let $F = [1 \ -c_1 \ -c_2]^t$ be the tap values of the equivalent filter for the cascade predictor after convergence, that is, $c_1 = f_0 + f_1$ and $c_2 = -f_0 f_1$. Using (4), we obtain

$$c_1 = c_1^*, \tag{8}$$

$$c_2 = (r_x(1)/r_x(0))^2 c_2^* = \beta^2 c_2^*. \tag{9}$$

The filter taps thus converge to a biased version of the optimal filter; in this case, the bias affects the second coefficient as a function of the correlation of the signal; it is always $\beta \leq 1$ with larger values for signals with a narrow power spectrum.

These results can be used to verify experimentally the performance of the cascade structure. Consider a stationary process $x(n)$ obtained by filtering unit variance Gaussian white noise through a two-pole filter $H(z) = 1/[(1 - \rho e^{j\theta} z^{-1})(1 - \rho e^{-j\theta} z^{-1})]$; the eigenvalue spread for the 2×2 correlation matrix is, in this case, $\lambda = (1 + \rho^2 + 2\rho\cos\theta)/(1 + \rho^2 - 2\rho\cos\theta)$, yielding increasing values for $\theta \to 0$. The bias factor for the second coefficient of the cascade predictor is $\beta = 2\rho\cos\theta/(1 + \rho^2)$; ideally, when $\rho \to 1$ and $\theta \to 0$, the performance of the cascade should prove superior and largely immune from effects deriving from either the large eigenvalue spread and the mode coupling. Indeed, the simulations shown in Fig. 2 confirm these predictions. In the experiments, $\rho = 0.95$, and $\theta = \pi/20$; the filter coefficients are updated using the standard LMS equations [1] with the step size chosen in both normal and cascaded filters to maximize the speed of convergence.[1] The graphs show the results of an ensemble average over 100 trials for 2000 iterations; Fig. 2(a) displays the trajectories of $c_1(n)$ and $c_2(n)$ for a two-tap LMS filter, a lattice LMS, and the cascade LMS; the Wiener values are shown by the dotted lines. Fig. 2(b) shows the mean-square errors for the same three filters. The superior performance of the cascade predictor confirms the expected results.

B. Cascade of Two-Tap Filters

A less-constrained building block than the one-tap filter is needed to approximate complex poles (and zeros) in the generating AR(MA)

[1] This is achieved by increasing the step-size value until the stability limit is reached.

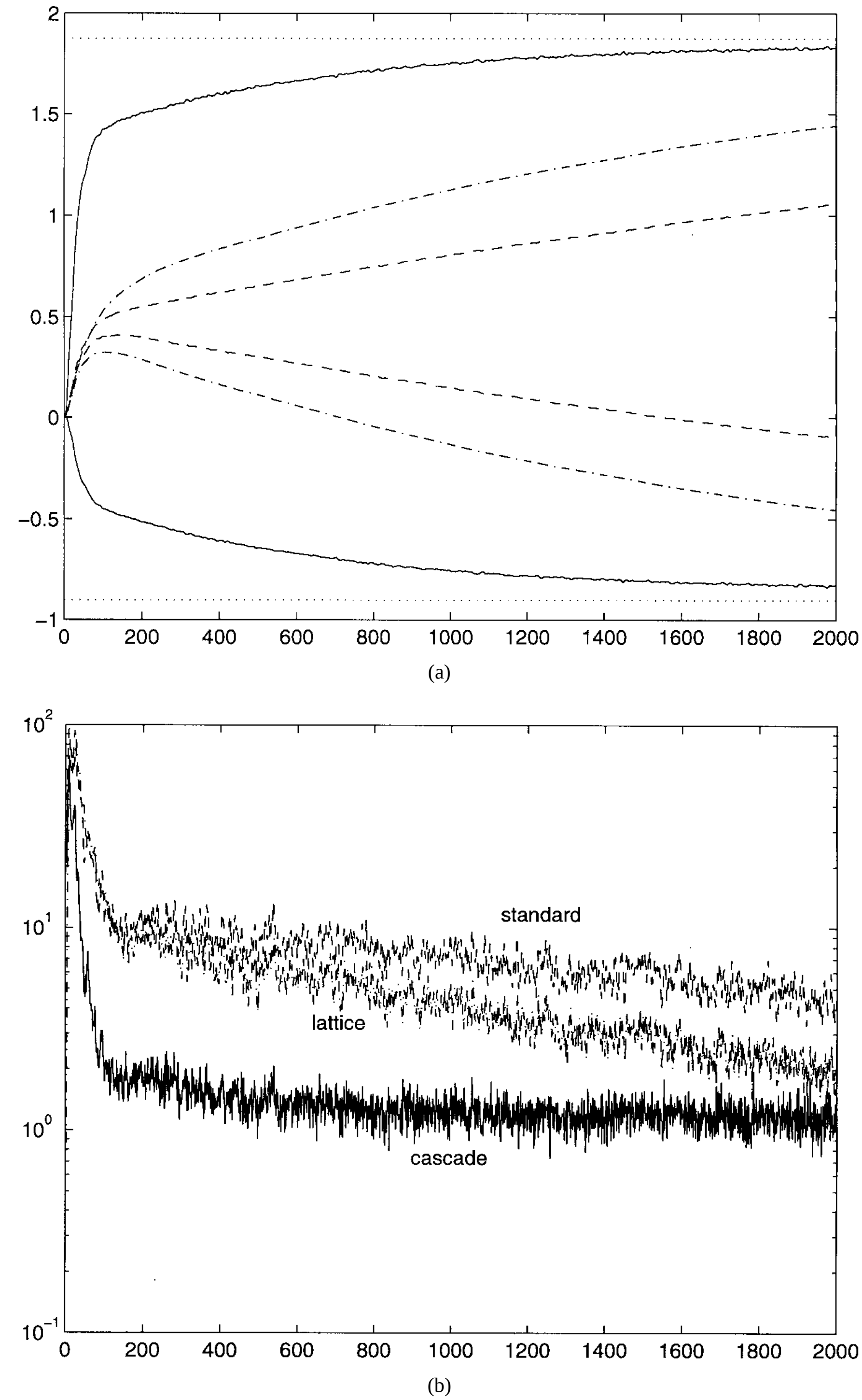
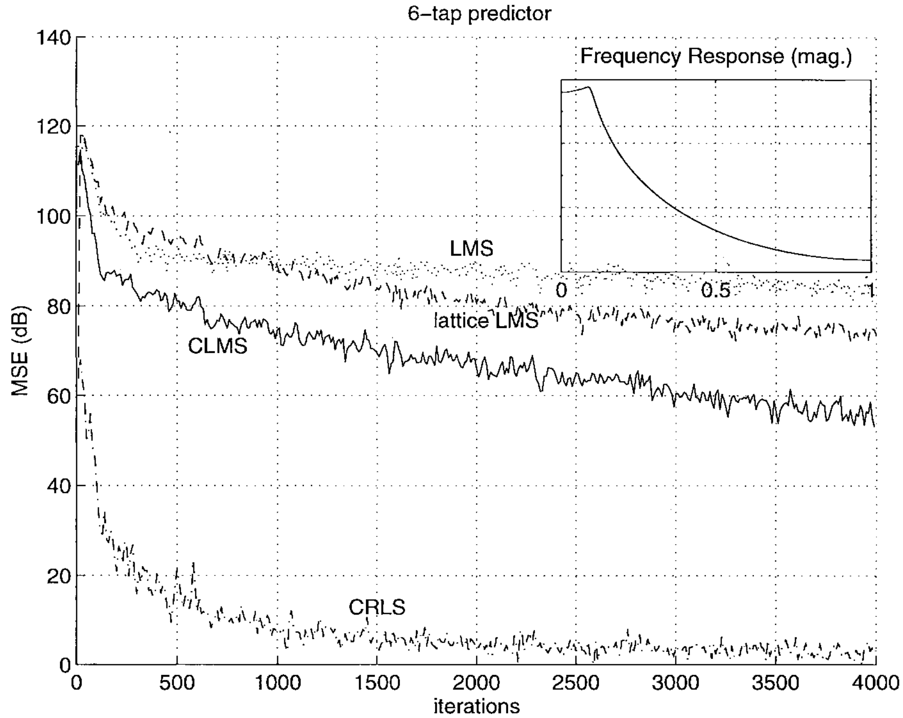


Fig. 2. Performance comparison between LMS (dashed), lattice LMS (dash-dot), and one-tap cascade LMS (solid) in predicting an AR(2) signal. (a) Trajectories of the filter taps. (b) MSE.

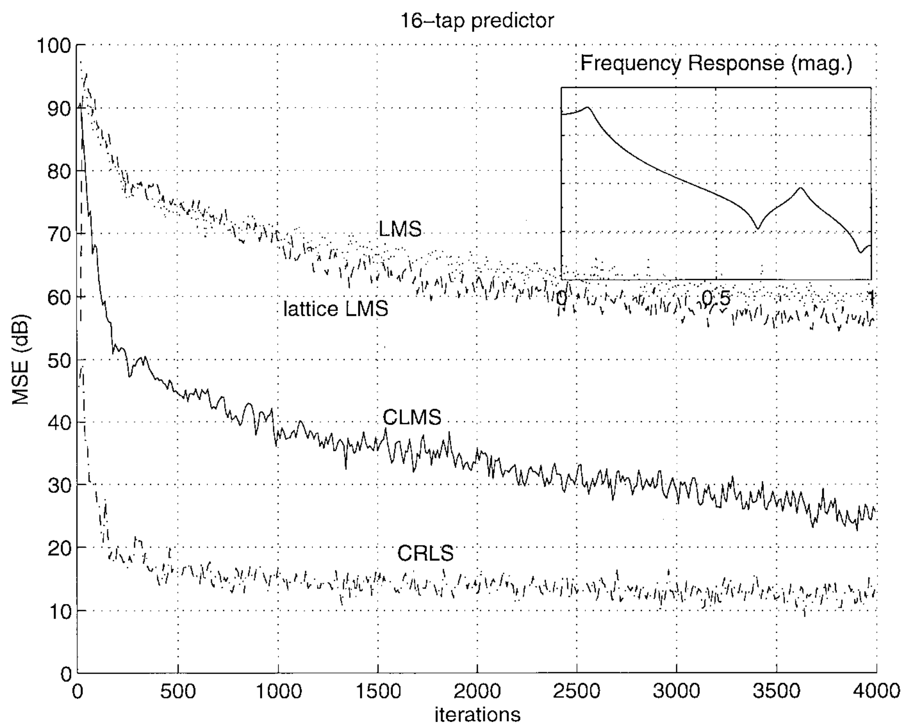
model. The simplest such element for real input signals is a two-tap filter (or, equivalently, a one-tap complex filter for complex inputs). Unfortunately, it is hard to obtain manageable analytical results for the resulting predictor because of the involved recursive formula for the autocorrelation of $e_k(n)$ [equation (5) in this case depends also on $r_k(m \pm 2)$]. However, as stated earlier, the main advantages of the cascade appear in the transient phase of the adaptation rather than in the steady-state limit. Theoretical analysis of the transient is always extremely complex; in this correspondence, we prefer to give some typical simulation results to illustrate the behavior of the cascade, deferring a more detailed analysis of its convergence properties to future research.

The update mechanism for the single sections can be the LMS algorithm as before, and in this case, we will label the cascade predictor "CLMS." For a two-tap filter, however, the normal equations are simple enough to be solved directly; for each stage k, the estimates of the input autocorrelation at time n and lag m $r_k^{(n)}(m)$ can be computed as a running estimate

$$r_k^{(n)}(m) = \lambda r_k^{(n-1)}(m) + x_k(n)x_k(n-m) \tag{10}$$



(a)



(b)

Fig. 3. MSE in linear prediction problems for LMS (dotted line), lattice LMS (dashed), cascade of two-tap filters with LMS (solid) and RLS (dash-dot) adaptation. The predicted signal is an (a) AR(6) lowpass signal and (b) ARMA signal (six poles, four zeros).

where λ is a forgetting factor (close to unity). These values, for $m = 0, 1, 2$, can be used with (6) and (7) to compute the filter coefficients for all n. This adaptation algorithm is simply a direct-form implementation of the RLS update for the single stages; it provides an extremely fast convergence and immediate stability monitoring but at a linear computational cost. We will label this realization of the cascade predictor "CRLS." The computational complexity of the cascade structure is always $O(M)$; in this case, the computational costs for a structure equivalent to a $2M$-tap filter are $5M$ multiplies for CLMS and $10M$ multiplies and M divisions for CRLS.

Fig. 3 displays, as an example, the performances of LMS, lattice LMS, CLMS, and CRLS filters for the one-step-ahead linear prediction problem; the input signals are obtained by filtering unit-variance

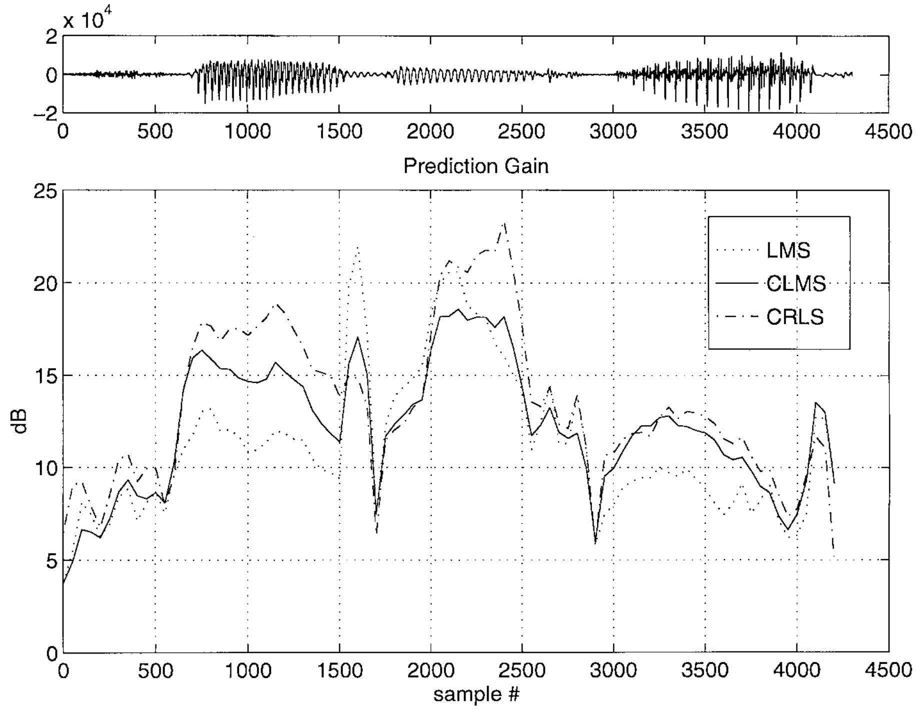


Fig. 4. Speech prediction. Prediction gain for a 12-tap LMS filter (dotted line), cascade LMS (solid), and cascade RLS (dash-dot).

white Gaussian noise by the AR an the ARMA transfer function displayed in the upper-right box. The cascade structure contains $N/2$ two-tap stages, where N is the number of taps of the standard LMS filter of reference; the step sizes are chosen as in Section II-A. The MSE is plotted against the number of iterations, averaged over an ensemble of 100 trials. Clearly, the CRLS results are not meant to be compared with the LMS directly; they are displayed in the same figure only by reason of convenience.

C. An Application: Linear Prediction of Speech

The agile behavior of the cascade structure at the onset of the adaptation process suggests its potential in tracking nonstationary signals such as speech. The two-tap cascade structure has indeed been tested against a reference speech linear prediction scheme [14]. The update equation for the 12-tap $c^{(m)}(n)$ of the proposed FIR predictor is

$$c^{(m)}(n+1) = a\bar{c}^{(m)} + (1-a)[c^{(m)}(n) + \alpha(n)e(n)x(n-k)] \quad (11)$$

where

- $e(n)$ usual prediction error;
- a leakage term;
- $\bar{c}^{(m)}$ quiescent values for the filter taps.

The step size α is time varying and inversely proportional to a running estimate of the input variance; details on the suggested values for the prediction parameters can be found in the cited reference. The competing cascade predictor uses the same parameters and is composed of six stages of two-tap filters. To obtain a quantitative measure of the differences in performance between the two systems, the index

$$\Delta = \frac{\sum(P_C(n) - P_L(n))}{\sum P_L(n)} \times 100 \quad (12)$$

is defined, where $P_C(n)$ and $P_L(n)$ are the time-varying segmental prediction gains for the cascade and the system of (11), respectively. In Fig. 4, a short speech segment is shown along with the prediction gains over time for the original system and for the cascade, with both LMS and RLS-like adaptation schemes. The index values are $\Delta = 8.45$ and $\Delta = 19.1$ for the CLMS and for the CRLS, respectively.

III. Conclusion

Even though the theoretical analysis of the cascade structure is difficult, experimental simulations show that in many cases, the cascade predictor displays an interesting ability to converge rapidly to a good approximation of the optimal predictor, surpassing even computationally more expensive structures. It circumvents many of the fundamental problems in the convergence of LMS filters, albeit at the price of a number of "weak spots" toward certain classes of signals. The interest of the cascade structure is eminently practical, for instance, as a start-up method for other kinds of global gradient updates. In the absence of a more conclusive set of comparisons with established applications, it is premature to judge the general applicability of the scheme; the results presented here, however, indicate an interesting potential of the FIR cascade structure in terms of speed of convergence and computational efficiency.

References

[1] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice Hall, 1985.
[2] P. Clarkson, *Optimal and Adaptive Signal Processing*. Boca Raton, FL: CRC, 1993.
[3] D. Mansour and A. Gray, "Unconstrained frequency-domain adaptive filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, pp. 726–734, Oct. 1982.
[4] M. Mboup *et al.*, "LMS coupled adaptive prediction and system identification," *IEEE Trans. Signal Processing*, vol. 42, pp. 2607–2615, Oct. 1994.
[5] M. V. Dokic and P. M. Clarkson, "Real-time adaptive filters for time-delay estimation," *Mech. Syst. Signal Process.*, vol. 6, pp. 403–418, 1992.
[6] C. F. N. Cowan, "Performance comparison of finite linear adaptive filters," *Proc. Inst. Elect. Eng.*, pt. F, vol. 134, pp. 211–216, 1987.

[7] G. Carayannis *et al.*, "A fast sequential algorithm for least-squares filtering and prediction," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, Dec. 1983.
[8] J. M. Cioffi and T. Kailath, "Fast recursive least squares transversal filters for adaptive processing," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, 1984.
[9] L. B. Jackson and S. L. Wood, "Linear prediction in cascade form," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-26, pp. 518–528, Dec. 1978.
[10] M. Nayeri and K. Jenkins, "Alternate realizations to adaptive IIR filters," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 485–496, Apr. 1989.
[11] U. Forssen, "Analysis of adaptive FIR filters in cascade form," *IEEE Trans. Circuits Syst. II*, vol. 41, pp. 392–401, June 1994.
[12] J. M. Raulin *et al.*, "A 60 channel PCM-ADPCM converter," *IEEE Trans. Commun.*, vol. 30, pp. 567–573, Apr. 1982.
[13] W. J. Song and M. S. Park, "A complementary pair LMS algorithm for adaptive filtering," in *Proc. ICASSP*, 1997, vol. 3, pp. 2261–2264.
[14] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*. Englewood Cliffs, NJ: Prentice-Hall, 1984.

DCT/DST and Gauss–Markov Fields: Conditions for Equivalence

José M. F. Moura and Marcelo G. S. Bruno

***Abstract*— The correspondence addresses the intriguing question of which random models are equivalent to the discrete cosine transform (DCT) and discrete sine transform (DST). Common knowledge states that these transforms are *asymptotically* equivalent to first-order Gauss *causal* Markov random processes. We establish that the DCT and the DST are *exactly* equivalent to homogeneous one-dimensional (1-D) and two-dimensional (2–D) Gauss *noncausal* Markov random fields defined on finite lattices with appropriate boundary conditions.**

I. Introduction

In this correspondence, we establish the second-order equivalence between the discrete sine transform (DST) and the discrete cosine transform (DCT) and arbitrary order noncausal Gauss–Markov random fields (GMrf's) defined on a finite lattice. We prove this by showing that the DST and the DCT diagonalize the covariance matrix associated with these fields. Following [1], we work with the inverse of the covariance matrix, which is called the potential matrix, that is highly structured; for homogeneous noncausal GMrf's of arbitrary order, it is given by a Toeplitz *canonical* matrix plus a boundary matrix.

Section II expresses the Toeplitz component of the potential matrix as matrix polynomials that are diagonalizable by either the DST or the DCT plus a perturbation matrix. Section III shows that for a given arbitrary order one-dimensional (1-D) GMrf, particular choices of boundary conditions (bc's) lead to a boundary matrix that cancels the perturbation term in the expansion of the Toeplitz canonical matrix. The final result is then an overall potential matrix that is diagonalizable by either the DST or the DCT. Results are extended to two-dimensional (2-D) GMrf's in Section IV using the Kronecker product. The same techniques can be used to show similar results for the other sinusoidal transforms introduced in [2].

Manuscript received September 24, 1996; revised March 6, 1998. This work was supported in part by DARPA under Grant DABT 63-98-1-0004. The work of M. G. S. Bruno was also supported in part by CNPq-Brazil. The associate editor coordinating the review of this paper and approving it for publication was Dr. Jitendra K. Tugnait.

The authors are with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213-3890 USA.

Publisher Item Identifier S 1053-587X(98)05966-2.

II. Notation and Preliminaries

Let $\mathbf{e}_1 = [1\ 0\ \cdots\ 0]^T$ and, for $1 \le i \le N$, $\mathbf{e}_i$ be a zero vector except for entry i, which is a 1. We define the following $N \times N$ matrices whose entries are zero, except as indicated:

- reflection matrix $\mathbf{J}$, with counter diagonal of ones;
- forward shift $\mathbf{K}_1$, with first upper diagonal of ones;
- backward shift $\mathbf{K}_2 = \mathbf{K}_1^T$, with lower diagonal of ones;
- the powers of the shift operators $\mathbf{K}_1^i$ and $\mathbf{K}_2^i$, $1 \le i \le N-1$, with the ith-upper, or, respectively, i-lower, diagonal of ones;
- $\mathbf{F}_0 = \mathbf{0}$, and symmetric matrices $\mathbf{F}_i = \mathbf{J}(\mathbf{K}_1^{N-i} + \mathbf{K}_2^{N-i})$, $0 \le i \le N-1$, with the $(i-1)$th lower and upper *counter* diagonals of ones;
- $\mathbf{H} = \mathbf{K}_1 + \mathbf{K}_2$, with the first upper and lower diagonal of ones;
- $\overline{\mathbf{H}}$ is like $\mathbf{H}$, with, in addition, the entries (1, 1) and (N, N), which are also ones.

Eigenstructure of $\mathbf{H}$ *and* $\overline{\mathbf{H}}$*:* Matrices $\mathbf{H}$ and $\overline{\mathbf{H}}$ are symmetric tridiagonal matrices. Their eigenstructure is well known, e.g., [3]. The eigenvectors of $\mathbf{H}$ are the basis vectors of the DST, and the eigenvectors of $\overline{\mathbf{H}}$ are the basis vectors of the DCT. In other words, their eigenvectors are the rows of the orthogonal transform matrices $\mathbf{S}$ and $\mathbf{C}$ for the DST and DCT, respectively, which are defined as

$$S_{kn} = \left[\sqrt{\frac{2}{N+1}}\ \sin\ \frac{\pi(k+1)(n+1)}{N+1}\right]_{0\le k,\, n \le N-1} \tag{1}$$

$$C_{kn} = \begin{cases} \sqrt{\dfrac{1}{N}}, & k = 0,\ 0 \le n \le N-1 \\ \sqrt{\dfrac{2}{N}}\ \cos\ \dfrac{\pi(2n+1)k}{2N}, & 1 \le k \le N-1 \\ & 0 \le n \le N-1. \end{cases} \tag{2}$$

Toeplitz Matrices: We decompose a banded symmetric $N \times N$ Toeplitz matrix as

$$\forall\, m \le N: \quad \mathbf{T} = b_1\mathbf{I} + b_2(\mathbf{K}_1 + \mathbf{K}_2) + b_3(\mathbf{K}_1^2 + \mathbf{K}_2^2) + \cdots + b_m(\mathbf{K}_1^{m-1} + \mathbf{K}_2^{m-1}). \tag{3}$$

Powers $[\mathbf{K}_1^{i-1} + \mathbf{K}_2^{i-1}]$*:* The following lemma relates $(\mathbf{K}_1^i + \mathbf{K}_2^i)$ to the powers of $\mathbf{H}^i$ and of $\overline{\mathbf{H}}^i$.

Lemma II.1: Let $\mathbf{F}_{-1} = \mathbf{F}_0 = \mathbf{0}$ and $\mathbf{F}_i$, $i \ge 1$ be defined as before. Then

$$0 \le i \le N-1: \quad \mathbf{K}_1^i + \mathbf{K}_2^i = P_i(\mathbf{H}) + \mathbf{F}_{i-1} \tag{4}$$

where $P_0(\mathbf{H}) = 2\mathbf{I}$, and $P_i(\mathbf{H})$, $1 \le i \le N-1$ are matrix polynomials in $\mathbf{H}$

$$i \text{ even:} \quad P_i(\mathbf{H}) = \mathbf{H}^i - i\mathbf{H}^{i-2} - \alpha_1^i\mathbf{H}^{i-4} - \cdots - \alpha_{(i-2)/2}^i\mathbf{I} \tag{5}$$

$$i \text{ odd:} \quad P_i(\mathbf{H}) = \mathbf{H}^i - i\mathbf{H}^{i-2} - \alpha_1^i\mathbf{H}^{i-4} - \cdots - \alpha_{(i-3)/2}^i\mathbf{H}. \tag{6}$$

Similarly

$$0 \le i \le N-1: \quad \mathbf{K}_1^i + \mathbf{K}_2^i = Q_i(\overline{\mathbf{H}}) - \mathbf{F}_i \tag{7}$$

where $Q_0(\overline{\mathbf{H}}) = 2\mathbf{I}$ and $Q_i(\overline{\mathbf{H}})$, $1 \le i \le N-1$ are matrix polynomials in $\overline{\mathbf{H}}$

$$i \text{ even:} \quad Q_i(\overline{\mathbf{H}}) = \overline{\mathbf{H}}^i - i\overline{\mathbf{H}}^{i-2} - \alpha_1^i\overline{\mathbf{H}}^{i-4} - \cdots - \alpha_{(i-2)/2}^i\mathbf{I} \tag{8}$$

$$i \text{ odd:} \quad Q_i(\overline{\mathbf{H}}) = \overline{\mathbf{H}}^i - i\overline{\mathbf{H}}^{i-2} - \alpha_1^i\overline{\mathbf{H}}^{i-4} - \cdots - \alpha_{(i-3)/2}^i\overline{\mathbf{H}}. \tag{9}$$