$$s_k = W_N^{-[(k \bmod m)^2/2+q(k \bmod m)]} W_N^{k^2/2+qk}$$
$$k = 0,1,\ldots,N-1 \quad (6)$$

We then introduce the following change of variables:

$$k = xm+i \quad i = k \bmod m$$
$$x = 0,1,\ldots,sm-1 \quad i = 0,1,\ldots,m-1 \quad (7)$$

From eqns. 6 and 7 we obtain

$$s_{xm+i} = W_{2s}^{x^2} W_{sm}^{x(i+q)} \quad (8)$$

From eqn. 8 it is obvious that if m is divisible by 2, the elements of sequence $\{s_k\}$ are some powers of W_{sm}, so the alphabet size is $A = sm$. However, if m is not divisible by 2, the alphabet size is $A = 2sm$.

For N odd, we obtain

$$s_{xm+i} = (W_{sm}^t)^{x(xm+1)} W_{sm}^{x(i+q)} \quad (9)$$

where $t = 2^{-1} \bmod sm$. As t is relatively prime with sm, it follows that W^t_{sm} is a primitive smth complex root of unity. Consequently, the sequence $\{s_{xm+i}\}$ given by eqn. 9 has alphabet size equal to $A = sm$.

For $N = m^2$, these GCL sequences will have the alphabet size $A = m$, the same as Frank sequences. For any length $N = sm^2$, it is possible to form $(sm)^m$ totally different strings $\{b_i^{(sm)}\}$, and hence $(sm)^m$ different sm-phase GCL sequences.

Milewski sequences as special case of GCL sequences: Let $\{u_n\}$, $n = 0, 1, \ldots, g-1$, be a Zadoff-Chu sequence of length g. The Milewski sequence of length $N = g^{2h+1}$ is obtaining by concatenation of the rows of matrix $\{z_{x,i}\}$, defined as [3]

$$z_{x,i} = u_{x \bmod s} W_{sm}^{xi} \quad s = g \quad m = g^h$$
$$x = 0,1,2,\ldots,g^{h+1}-1 \quad i = 0,1,2,\ldots,g^h-1 \quad (10)$$

where W_{sm} is a primitive g^{h+1}th complex root of unity.

For g even, Milewski sequence $\{s_k\}$ can be represented as

$$s_{xm+i} = W_s^{x^2/2+q_M x} W_{sm}^{xi} \quad (11)$$

Comparing eqn. 8 and eqn. 11, we see that, for g even, Milewski sequences are equal to the GCL sequences obtained by using the modulating string of eqn. 5, where $W_{sm}^{p(i)} = 1$, $m = g^h$ and $q = mq_M$.

For g odd, Milewski sequence $\{s_k\}$ can be represented as

$$s_{xm+i} = W_{sm}^{mx(x+1)/2+mq_M x} W_{sm}^{xi} \quad (12)$$

On the other hand, the same expression can be obtained from the definition of GCL sequences and eqn. 5, if we take $q = mq_M + (m-1)/2$.

27 October 1993

Electronics Letters Online No: 19940080

B. M. Popović (*IMTEL-Institute of Microwave Techniques and Electronics, B. Lenjina 165b, 11071 Novi Beograd, Yugoslavia*)

References

1 BÖMER, L., and ANTWEILER, M.: 'Perfect N-phase sequences and arrays', *IEEE J. Sel. Areas Commun.*, 1992, **10**, (4), pp. 782–789

2 POPOVIĆ, B.M.: 'Generalised chirp-like polyphase sequences with optimum correlation properties', *IEEE Trans.*, 1992, **IT-38**, (4), pp. 1406–1409

3 MILEWSKI, A.: 'Periodic sequences with optimal properties for channel estimation and fast start-up equalisation', *IBM J. Res. Develop.*, 1983, **27**, (5), pp. 426–431

4 SUEHIRO, N., and HATORI, M.: 'Modulatable orthogonal sequences and their applications to SSMA systems', *IEEE Trans.*, 1988, **IT-34**, (1), pp. 93–100

5 POPVIĆ, B.M.: 'Efficient matched filter for the generalized chirp-like polyphase sequences', to be published in *IEEE Trans.*, Aerospace and Electronics Systems, 1994

Generalised lapped orthogonal transforms

R.L. de Queiroz, T.Q. Nguyen and K.R. Rao

Indexing terms: Transforms, Signal processing

A class of linear-phase paraunitary filter banks is developed, possessing fast implementation algorithms based on the discrete cosine transform. In this formulation, the lapped orthogonal transform is a particular case which was extended to accommodate the overlap of any number of blocks. Optimised design examples are presented.

Introduction: The discrete cosine transform (DCT) [1] is used in most of the international standards for image compression and recently the lapped orthogonal transform (LOT) [2] was developed as a competitive alternative because of its extended basis functions which overlap across traditional block boundaries, thus eliminating the blocking effect. It is well known that these two forms are particular choices of FIR linear-phase paraunitary filter banks (LPPUFBs) [3,4]. Linear-phase filter banks have been studied intensively and several design approaches can be found in the literature. However, fast implementation algorithms were usually ignored. Very recently, a minimal structure to implement all LPPUFBs (where the filter lengths are the same) was developed [5]. Based on this work [5], we introduce a particular simplification leading to a class of LPPUFB which we call generalised linear-phase lapped orthogonal transforms (GENLOTs). The GENLOTs have a fast implementation algorithm based on the DCT, and both the DCT and LOT can be regarded as special cases. One of the reasons for the growing popularity of the LOT is the fact that it possesses a fast implementation algorithm and good performance. Also, its algorithm is based on the DCT which is highly popular in image coding and for which an uncountable number of algorithms, chips, and computer programs have been developed for its implementation. We follow here the same principle, i.e. to include the DCT as a basic transform and add stages to it.

Simplified factorisation of LPPUFB: Let L be an upper limit for the length of the filters and M be the number of channels. From the lapped transform viewpoint (time domain), L is the length of the basis functions of the transform and M is the block size. Recall that $L = M$ and $L = 2M$ for the DCT and LOT, respectively. The overlap amount N is expressed as $L = NM$ and is related to the degree n of the polyphase transfer matrix $\mathbf{E}(z)$ [3] by $n = N - 1$ [3,4]. The structure for implementing all LPPUFBs of M channels (M even) depends on N. From the work by Soman *et al.* [5], we know that a complete parameterisation of any paraunitary $\mathbf{E}(z)$ of degree $N-1$ characterising a linear-phase filter bank is given by

$$\mathbf{E}(z) = \mathbf{SPT}_{N-1}\Lambda(z)\mathbf{T}_{N-2}\Lambda(z)\cdots\Lambda(z)\mathbf{T}_0\mathbf{P} \quad (1)$$

where

$$\mathbf{P} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{J} \end{bmatrix} \qquad \Lambda(z) = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & z^{-1}\mathbf{I} \end{bmatrix} \quad (2)$$

$$\mathbf{S} = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{S}_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_1 \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{J} \\ \mathbf{I} & -\mathbf{J} \end{bmatrix} \quad (3)$$

and

$$\mathbf{T}_i = \frac{1}{2} \begin{bmatrix} \mathbf{I} & \mathbf{I} \\ \mathbf{I} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{U}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_i \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{I} \\ \mathbf{I} & -\mathbf{I} \end{bmatrix} \quad (4)$$

The matrices $\mathbf{0}$, $\mathbf{I}$, $\mathbf{S}_0$, $\mathbf{S}_1$, $\mathbf{U}_i$, and $\mathbf{W}_i$ are square orthogonal matrices, each of size $M/2 \times M/2$. $\mathbf{0}$ and $\mathbf{I}$ are the null and identity matrices, respectively, and $\mathbf{J}$ is a counter-diagonal identity matrix. $\mathbf{S}_0$, $\mathbf{S}_1$, $\mathbf{U}_i$, and $\mathbf{W}_i$ are general orthogonal matrices. However, it is easy to see that $\mathbf{SPT}_{N-1}$ can be simplified to

$$\mathbf{SPT}_{N-1} = \frac{1}{\sqrt{2}} \begin{bmatrix} \tilde{\mathbf{U}}_{N-1} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{W}}_{N-1} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{I} \\ \mathbf{I} & -\mathbf{I} \end{bmatrix} \quad (5)$$

where $\tilde{\mathbf{U}}_{N-1} = \mathbf{S}_0\mathbf{U}_{N-1}$ and $\tilde{\mathbf{W}}_{N-1} = \mathbf{S}_1\mathbf{W}_{N-1}$. Thus, revisiting eqn. 1, we can state that we obtain a polyphase matrix of LPPUFB $\mathbf{E}_{k+1}(z)$, with degree $k+1$, from one with degree k, denoted as $\mathbf{E}_k(z)$, by

$$\mathbf{E}_{k+1}(z) = \mathbf{K}_{k+1}(z)\mathbf{E}_k(z) \quad (6)$$

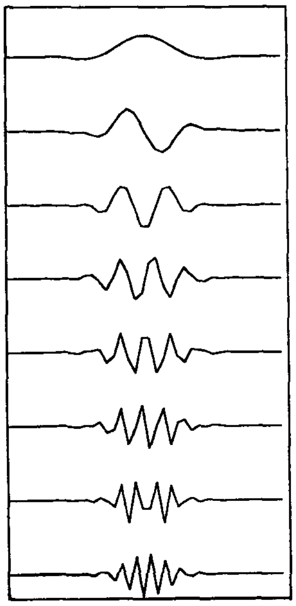
where

$$\mathbf{K}_i(z) = \frac{1}{2}\begin{bmatrix}\mathbf{U}_i & \mathbf{0}\\ \mathbf{0} & \mathbf{W}_i\end{bmatrix}\begin{bmatrix}\mathbf{I} & \mathbf{I}\\ \mathbf{I} & -\mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & z^{-1}\mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{I} & \mathbf{I}\\ \mathbf{I} & -\mathbf{I}\end{bmatrix} \quad (7)$$

GENLOTs: Note that to obtain the LOT from the DCT, a stage identical to $\mathbf{K}_1(z)$ was added to the output of the DCT [2]. Then, the overlapping can be viewed as a postprocessing operation over the output of the DCT. If $L = NM$, the basis functions will extend over N signal blocks. Accordingly, to construct a three-block overlap beginning with the traditional LOT ($N = 2$), it would be a mere process of repeating the postprocessing to the output of the LOT. The GENLOTs use this principle and use the starting stage in this recursion as the DCT, i.e. the polyphase transfer matrix of degree zero is always the DCT matrix, as

$$\mathbf{E}(z) = \mathbf{K}_{N-1}(z)\mathbf{K}_{N-2}(z)\ldots\mathbf{K}_1(z)(\mathbf{DCT}) \quad (8)$$

Of course, the parameters involved in each stage are different and each orthogonal matrix $\mathbf{U}_i$ and $\mathbf{W}_i$ can be parameterised into a set of $M(M-2)/8$ plane rotations, each with one degree of freedom.



676/1

Fig. 1 *Basis functions (filter impulse responses) of a GENLOT with L = 40, M = 8, N = 5, designed for maximum transform coding gain, assuming an AR(1) signal with intersample correlation 0.95*

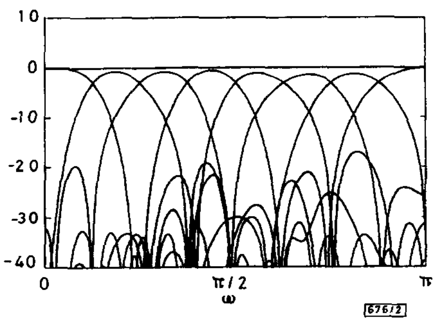


Fig. 2 *Filter frequency responses (in dB) of a GENLOT whose basis functions are shown in Fig. 1*

This results in a total of $M(N-1)(M-2)/4$ degrees of freedom. As an example. Fig. 1 shows the basis functions of an $N = 5$ LPPUFB for $M = 8$, and Fig. 2 shows the frequency response of the respective filters. As in the LOT [2], the transform coding gain was optimised. However, the parameters were found by nonlinear optimisation routines, searching the space of all $M(N-1)(M-2)/4$ degrees of freedom.

Remarks: The increase in computation when the length of the basis function goes from NM to $(N+1)M$ is given by two sets of trivial butterflies and one stage involving the orthogonal matrices. This is constant for any overlap N. However, the implementational cost can be further decreased by simplifying the inatrices $\mathbf{U}_i$ and $\mathbf{W}_i$ as was done for the LOT, i.e. choosing a small set of plane rotations which will approximate the orthogonal matrix. Furthermore, improvements to the optimisation methods are currently being studied.

Acknowledgments: This work was supported in part by the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Brazil, under Grant 200.804-90-1.

5 October 1993
Electronics Letters Online No: 19940193

R. L. de Queiroz and K. R. Rao (*Electrical Eng. Department, University of Texas at Arlington, Box 19016, Arlington, TX 76019-0016, USA*)

T. Q. Nguyen (*MIT Lincoln Laboratory, Rm. D-361, 244 Wood St., Lexington, MA 02173-9108, USA*)

References

1 RAO, K.R., and YIP, P.: 'Discrete cosine transform: Algorithms, advantages, applications' (San Diego, CA: Academic press, 1990)
2 MALVAR, H.S., and STAELIN, D.H.: 'The LOT: Transform coding without blocking effects', *IEEE Trans.*, 1989, **ASSP-37**, pp. 553–559
3 VAIDYANATHAN, P.P.: 'Multirate systems and filter banks (Englewood Cliffs, NJ: Prentice-Hall, 1993)
4 MALVAR, H.S.: 'Signal processing with lapped transforms' (Norwood, MA: Artech House, 1992)
5 SOMAN, A.K., VAIDYANATHAN, P.P., and NGUYEN, T.Q.: 'Linear-phase orthonormal filter banks'. Proc. IEEE Int. Conf. on Acoust., Speech, Signal Processing, April 1993, (Minneapolis, MN), Vol. III, pp. 209–212

Improved estimator for a discretised learning routing algorithm

T.C. Wan and C. Douligeris

Indexing terms: Learning systems, Computer networks

The use of learning automata in dynamic routing algorithms has shown promise. An improved estimator for the new scheme flow control with discretised learning (NSDL) routing algorithm is presented which provides a better profile of the network environment, to improve the performance of the NSDL scheme.

Introduction: Existing distributed routing schemes, which operate under the assumption that network traffic is quasistatic and varies slowly with time, are inadequate for handling the rapidly fluctuating conditions in current gigabit packet networks. The need for a robust dynamic algorithm which is computationally efficient and which is able to adapt effectively to rapidly changing traffic profiles to maximise overall throughput is therefore essential.

Narendra *et al.* [1,2] have investigated the use of learning automata for making routing decisions. Extensions to the above scheme have been proposed by Vasilakos *et al.* [3,4] which used a different training algorithm for updating the action probabilities.

Under current schemes, the inputs are taken as being direct feedbacks of the environment. As such, they are susceptible to changes in a dynamic nonstationary environment. In addition, it is not possible to derive quantitative measurements from the learning automaton regarding a particular action under current schemes. Owing to the nonstationary nature of the environment, the model must track changes in the real environment and reflect it in the model as the environment changes. Adaptive algorithms are well