the unknown plant and the adaptive filter after convergence are shown in Fig. 3(b) and (c), respectively. The average squared error between the two responses was 2.4545×10^{-4} over those 200 samples. We can see from Fig. 3 that the adaptive process converged faithfully. Fig. 4 shows the corresponding simulation results obtained by using the LAIIRDF. Fig. 4(a) shows the squared output error (in dB) curve when the summation of squared output error of the last 20 steps fell below 1.0×10^{-11}. The impulse response of the unknown plant is shown in Fig. 3(b) and that of the adaptive filter after convergence is shown in Fig. 4(b). The average squared error between the two responses was 1.6666×10^{-4} over those 200 samples.

V. CONCLUSIONS

We have presented a nonlinear adaptive IIR digital filter, namely, the NAIIRDF. Because of the nonlinear operator involved in this structure, this recursive digital filter is bounded-input/bounded-output stable. Based on this structure, an individual adaptation scheme to improve the convergence speed is incorporated into the adaptive algorithm, which can adjust each different parameter at every iteration so that their values are kept optimum for a new set of input samples. Simulation results for adaptive linear IIR system modeling have been given to demonstrate the performance of the proposed structure and algorithm.

REFERENCES

[1] B. Widrow and S. D. Stearns, *Adaptive Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1985.

[2] H. K. Kwan and Q. P. Li, "New nonlinear adaptive FIR digital filter for broadband noise cancellation," *IEEE Trans. Circuits Syst.–II: Analog and Digital Signal Processing,* vol. 41, no. 5, pp. 355–360, May 1994.

[3] W. B. Mikhael, F. H. Wu, L. G. Kazovsky, G. S. Kang, and L. J. Fransen, "Adaptive filters with individual adaptation of parameters," *IEEE Trans. Circuits Syst.,* vol. CAS-33, no. 7, pp. 677–686, July, 1986.

[4] Q. P. Li, "Linear and nonlinear adaptive digital filters and their applications," Ph.D. dissertation, Dept. of Electrical Engineering, University of Windsor, Windsor, Ontario, Canada, 1994.

Computing the Inverse DFT with the In-Place, In-Order Prime Factor FFT Algorithm

Kin-Man Lam and Hong Yan

***Abstract*—In this correspondence, we present a method for computing the inverse discrete Fourier transform (IDFT) by the in-place, in-order prime factor FFT algorithm (PFA). This is achieved by modifying the input and the output index mapping equations. This approach does not result in any additional cost in terms of program length and computational time.**

I. INTRODUCTION

Fast computation of the discrete Fourier transform (DFT) is of considerable importance in many areas of digital signal processing. However, due to the lack of efficient algorithms for the computation of long DFT's, a long 1-D array is translated into a multidimensional array by using prime factor index mapping.

The forward N-point DFT can be written as follows:

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk} \tag{1}$$

where $k = 0, 1, 2, \cdots, N-1$, and $W = exp(-j\frac{2\pi}{N})$

The inverse N-point DFT has a similar expression as follows:

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-nk} \tag{2}$$

where $n = 0, 1, 2, \cdots, N-1$

As we are often interested in the relative values of the sequence, the normalizing factor $\frac{1}{N}$ can be omitted. Thus

$$\text{IDFT}_n X(k) = x'(n) = \sum_{k=0}^{N-1} X(k) W_N^{-nk}. \tag{3}$$

In the case where the length N of the DFT is not prime, N can be factored as $N = N_1 N_2 \cdots N_m$, where all the N_i's and N_j's are relative prime for $i \neq j$. A linear mapping [1] can be applied to transform this 1-D DFT into an m-dimensional DFT with the following mapping equations:

$$n = < L_1 n_1 + L_2 n_2 + \cdots + L_m n_m >_N \tag{4}$$

$$k = < M_1 k_1 + M_2 k_2 + \cdots + M_m k_m >_N \tag{5}$$

for $n_i, k_i = 0, 1, \cdots N_i - 1$ and where $< t >_N$ means the residue of t.

A simple and more efficient indexing scheme [2], [3] results from considering these equations not as an m-dimensional problem but as a sequence of 2-D problems. The m-factor DFT is calculated by performing $N_2 \cdots N_m$ length-N_1 DFT's and then $N_1 N_3 \cdots N_m$ length-N_2 DFT's, and finally, $N_1 \cdots N_{m-1}$ length-N_m DFT's. Each of the m calculations will require a different two-variable map as shown:

$$n = < L_1 n_1 + L_2 n_2 >_N \tag{6}$$

$$k = < M_1 k_1 + L_2 n_2 >_N \tag{7}$$

where
$$L_1 = N/N_i \qquad i = 1, \cdots, m$$
$$L_2 = N/(N_1 N_2 \cdots N_{i-1} N_{i+1} \cdots N_m) = N_i$$
$$M_1 = L_1 L_1^{-1}$$
$$n_1 = 0, 1, \cdots, N_i - 1$$
$$n_2 = 0, 1, \cdots, (N_1 N_2 \cdots N_{i-1} N_{i+1} \cdots N_m - 1).$$

From the input and output index mappings of (6) and (7), the indices of the other dimensions $(N_1 N_2 \cdots N_{i-1} N_{i+1} \cdots N_m)$ in the mapping equations combine to form one index when one of the dimensions of the DFT's is being calculated.

II. COMPUTING THE IDFT

The expressions for the forward and inverse DFT are very similar. It is preferred that the forward DFT can be used in the computation of the inverse DFT. One obvious way [4] to compute the inverse DFT is by noting that

$$x'(n) = [\sum_{k=0}^{N-1} X^*(k) W_N^{nk}]^*. \tag{8}$$

The inverse DFT is obtained by computing the DFT of the conjugate of the input sequence and then by conjugating the output sequence.

Manuscript received December 31, 1993; revised January 13, 1995. The associate editor coordinating the review of this paper and approving it for publication was Prof. Ali N. Akansu.

The authors are with the Department of Electrical Engineering, University of Sydney, Syndey, Australia.

IEEE Log Number 9413318.

This approach requires additional steps to conjugate both the input and output sequences. Another method [5] is based on the fact that

$$x'(n) = j[\sum_{k=0}^{N-1}(jX^*(k))W_N^{nk}]^* \quad (9)$$

and that

$$x(n) = a(n) + j \cdot b(n) \ \Rightarrow \ j \cdot x^*(n) = b(n) + j \cdot a(n). \quad (10)$$

Therefore, the IDFT of a sequence can be obtained by exchanging the real and imaginary parts of the initial sequence, then performing a forward DFT, and finally exchanging the real and imaginary parts of the result.

In case of the in-place, in-order prime factor algorithm (PFA), this permutation can be absorbed into the output index mapping (7). From (3), it can be shown that

$$\begin{aligned}\text{IDFT}_n X(k) &= \sum_{k=0}^{N-1} X(k) W_N^{-nk} \\ &= DFT_{-n}X(k). \end{aligned} \quad (11)$$

This means that the output of the inverse DFT is equivalent to that of a forward DFT followed by a permuted output. This result means that the same forward DFT program can be used for the inverse DFT by modifying the coefficients of the mapping equations only. No extra computational cost is required in calculating the inverse DFT.

For permuting the output index mapping, (7) becomes

$$\begin{aligned} k' &= N - < M_1k_1 + L_2n_2 >_N \\ &=< M_1'k_1 + L_2'n_2 >_N \end{aligned} \quad (12)$$

where $M_1' =< -M_1 >_N$, $L_2' =< -L_2 >_N$. The corresponding input index mapping must also be modified so that the mappings are in place and in order. Therefore, (6) is written as

$$n' =< L_1n_1 + L_2'n_2 >_N . \quad (13)$$

Equations (13) and (12) form the input and output index mappings, respectively. These two equations can be written as follows:

$$n' =< (\alpha_1 \frac{N}{N_i})n_1 + (\alpha_2 N_i)n_2 >_N \quad (14)$$

$$k' =< (\gamma_1 \frac{N}{N_i})k_1 + (\gamma_2 N_i)n_2 >_N \quad (15)$$

where $\alpha_1 = 1$, $\gamma_1 = N_i - (\frac{N}{N_i})^{-1}$, and $\alpha_2 = \gamma_2 = (\frac{N}{N_i}) - 1$. The necessary conditions [3] for the two mapping functions to be in place and in order are

$$(\alpha_1, N_i) = (\alpha_2, \frac{N}{N_i}) = (\gamma_1, N_i) = (\gamma_2, \frac{N}{N_i}) = 1 \quad (16)$$

where (a, b) is the greatest common divisor of a and b. It is obvious that $(\alpha_1, N_i) = (\alpha_2, \frac{N}{N_i}) = (\gamma_2, \frac{N}{N_i}) = 1$. For $(\gamma_1, N_i) = 1$, it is proved as follows.

Proof: Let $(N_i - (\frac{N}{N_i})^{-1}, N_i) = c$, where c is an integer. Then

$$N_i - (\frac{N}{N_i})^{-1} = pc \quad (17)$$

$$N_i = qc \quad (18)$$

where p and q are integers. Multiply (17) by $\frac{N}{N_i}$, and then, take residue modulo N_i, and we have

$$\begin{aligned} &< N_i \frac{N}{N_i} - (\frac{N}{N_i})^{-1}\frac{N}{N_i} >_{N_i} =< pc\frac{N}{N_i} >_{N_i} \\ \Rightarrow \quad & N_i - 1 =< pc\frac{N}{N_i} >_{qc} \\ \Rightarrow \quad & -1 = pc\frac{N}{N_i} - kqc \\ \Rightarrow \quad & -1 = c(p\frac{N}{N_i} - kq) \end{aligned} \quad (19)$$

where k is an integer. Since all the quantities in (19) are integers and c is a positive integer, c must be equal to 1. This completes the proof.

Under the conditions in (16), the input index mapping results in all the $\frac{N}{N_i}$ row permutations for $n_2' = 0, 1, \cdots, \frac{N}{N_i} - 1$ being structurally identical. Hence, although the positions of the columns in the 2-D array may change, the elements within each column and their order remain unchanged. This is an essential property because the column DFT's are computed in the next stage. Similarly, the column DFT's are also computed in place and in order.

In fact, a PFA program can be modified slightly for performing both the forward and inverse DFT. A variable may be included to indicate the type of operation to be performed. The PFA subroutine is generally of the type such as the following:

```
PFA(XR, XI, N, FLAG)
float *XR, *XI;
int N;
int FLAG;
```

XR and XI are the real and imaginary parts of the input sequence, respectively, and of the result on output. If FLAG $= 1(-1)$, forward (inverse) DFT is to be performed. Inside the subroutine, the coefficients L_2 and M_1 are calculated using the following equations:

$$L_2 =< \text{FLAG} \cdot N_i >_N \quad (20)$$

$$M_1 =< \text{FLAG} \cdot L_1 L_1^{-1} >_N . \quad (21)$$

In fact, the coefficients for the mapping equations can be precalculated for specific applications.

III. Conclusion

The computation of the inverse DFT by means of PFA has been examined. This is achieved by modifying the input and the output index mappings only. This approach does not require any extra computation. By modifying the PFA subroutine slightly, one subroutine can be used for both the forward and inverse transforms.

References

[1] C. S. Burrus, "Index mappings for multidimensional formulation of DFT and convolution," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-25, pp. 239–242, June 1977.

[2] C. S. Burrus and P. W. Eschenbacher, "An in-place, in-order prime factor FFT algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 806–817, Aug. 1981.

[3] B. Arambepola, "Discrete Fourier transform processor based on the prime-factor algorithm," *Proc. Inst. Elec. Eng.*, vol. 130, pt. G, no. 4, pp. 138–144, Aug. 1983.

[4] L. R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975.

[5] P. Duhamel, B. Piron, and J. M. Etcheto, "On computing the inverse DFT," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, no. 2, pp. 285-286, Feb. 1988.

A New Transform with Symmetrical Coding Performance for Markov(1) Signals

Feng Zou and Richard R. Gallagher

Abstract—**Based on the DCT and DST-II, we present a new transform that provides a symmetrical performance for positively and negatively correlated Markov(1) signals. A fast computational algorithm is presented. A sufficient condition of symmetrical transforms is also provided, and the idea of constructing the DCST is then generalized to the KLT's of any Markov(1) signal.**

I. Introduction

The Karhunen-Lòeve transform (KLT) is the optimal transform in the sense of achieving minimum mean square error for the reconstructed signals. The source model widely used for the measurement of coding performances of various transform techniques is the first-order stationary Markov process (Markov(1)). The basis vectors of the KLT for Markov(1) signals consist of a set of sinusoidal sequences. For an autocorrelation coefficient $|\rho| < 1$, the frequencies of these eigenvectors (referred to as eigenfrequencies [1]) are not evenly distributed along the frequency axis. Therefore, an FFT-type fast algorithm is not available for a general case. But for highly correlated signals with $\rho > 0$, the KLT can be well approximated with discrete cosine transform (DCT) [2], [3]. Because of the availability of many fast computational algorithms [4], the DCT has been widely used in practice. However, for negatively correlated signals ($-1 < \rho < -0.5$), its performance is greatly inferior to the KLT [5]. On the contrary, the discrete sine transform-II (referred to as DST for simplicity) renders the best performance in this range of ρ, though its performance is not satisfactory for $\rho > 0$. We define a symmetrical transform as the one that provides the same coding performance for both positively and negatively correlated Markov(1) signals. Neither the DCT nor the DST is symmetrical. However, based on the DCT and DST, we introduce a new transform that provides symmetrical performance for the Markov(1) signals.

II. Discrete Cosine-Sine Transform

A. Discrete Cosine-Sine Transform

The autocorrelation function of a Markov(1) process has the following normalized format:

$$r_{|m-n|} = \rho^{|m-n|}, -1 < \rho < 1. \tag{1}$$

The basis vectors of its KLT for N even are given by [3], [6]

$$\phi(k,m) = \sqrt{\frac{2}{N+\lambda_k}} \sin\left[\omega_k\left\{m - \left(\frac{N-1}{2}\right)\right\} + (k+1)\frac{\pi}{2}\right] \quad 0 \le m, k \le N-1 \tag{2}$$

where m and k are time and spectral indexes, respectively; $\{\omega_k\}$ are the positive roots of the transcendental equation

$$\tan(N\omega_k) = -\frac{(1-\rho^2)\sin\omega_k}{\cos\omega_k - 2\rho + \rho^2\cos\omega_k} \tag{3}$$

and the eigenvalues λ_k are given by

$$\lambda_k = \frac{1-\rho^2}{1-2\rho\cos\omega_k + \rho^2} \quad 0 \le k \le N-1. \tag{4}$$

Manuscript received July 27, 1994; revised February 20, 1995. The associate editor coordinating the review of this paper and approving it for publication was Dr. K. J. Ray Liu.

The authors are with the Department of Electrical and Computer Engineering, Kansas State University, Manhattan, KS 66506 USA.

IEEE Log Number 9413328.

Assume for $\rho = \rho_p > 0$, the roots of (3) are ω_{kp}. When $\rho = -\rho_p$, we can write (3) as

$$\tan(N\omega_k) = -\frac{(1-\rho_p^2)\sin\omega_k}{\cos\omega_k + 2\rho_p + \rho_p^2\cos\omega_k}. \tag{5}$$

It is easy to verify that $\{\pi - \omega_{kp}\}$ are the roots of (5), i.e., the eigenfrequencies of ρ_p and $-\rho_p$ are symmetrical with respect to $\pi/2$. As $\rho \to 1$, KLT $\to$ DCT, with basis vectors defined as

$$\phi_c(k,m) = \begin{cases} \sqrt{\frac{2}{N}}\cos\frac{2m+1}{2N}k\pi & 1 \le k \le N-1 \\ \sqrt{\frac{1}{N}} & k=0 \end{cases} \quad 0 \le m \le N-1. \tag{6}$$

The N eigenfrequencies are given by

$$\omega_{kp} = \frac{k\pi}{N} \quad 0 \le k \le N-1. \tag{7}$$

Thus $(\pi - \frac{k\pi}{N})$ or

$$\omega_k = \frac{k+1}{N}\pi \quad 0 \le k \le N-1 \tag{8}$$

are the roots of (3) as $\rho \to -1$. By the procedures similar to [3] or [6] we obtain

$$\phi_s(k,m) = \begin{cases} \sqrt{\frac{2}{N}}\sin\left(\frac{2m+1}{2N}(k+1)\pi\right) & 0 \le k \le N-2 \\ (-1)^m\sqrt{\frac{1}{N}} & k=N-1 \end{cases} \quad 0 \le m \le N-1. \tag{9}$$

which is the definition of the DST (i.e., DST-II).

Taking the $N/2$ even basis vectors (0, 2, $\cdots$) from DCT and the $N/2$ odd basis vectors (1, 3, $\cdots$) from DST, we define the discrete cosine-sine transform (DCST) matrix as follows:

$$\phi(k,m) = \begin{cases} \sqrt{\frac{1}{N}} & k=0 \\ \sqrt{\frac{2}{N}}\cos\frac{2m+1}{2N}k\pi & k=2,4,\cdots,N-2 \\ \sqrt{\frac{2}{N}}\sin\frac{2m+1}{2N}(k+1)\pi & k=1,3,\cdots,N-3 \\ (-1)^m\sqrt{\frac{1}{N}} & k=N-1. \end{cases} \tag{10}$$

Also, we can write

$$\begin{aligned} \phi(2k,m) &= a_k\sqrt{\frac{2}{N}}\cos\frac{2m+1}{N}k\pi \\ \phi(2k+1,m) &= a_{k+1}\sqrt{\frac{2}{N}}\sin\frac{2m+1}{N}(k+1)\pi \\ k = 0,1,2,\cdots,\frac{N}{2}-1 \quad a_l &= \begin{cases} \frac{1}{\sqrt{2}} & l=0,N/2 \\ 1 & l=1,2,\cdots,N/2-1. \end{cases} \end{aligned} \tag{11}$$

The forward and inverse DCST's are defined as

$$F(n) = \sum_{m=0}^{N-1}\phi(n,m)x(m) \quad 0 \le n \le N-1 \tag{12}$$

and

$$x(m) = \sum_{n=0}^{N-1}\phi(n,m)F(n) \quad 0 \le m \le N-1 \tag{13}$$