Transactions Briefs

A Cyclic Correlated Structure for the Realization of Discrete Cosine Transform

Yuk-Hee Chan and Wan-Chi Siu

Abstract—**In this paper we propose using the correlated cosine structure (CCS) for the computation of the discrete cosine transform (DCT). This structure has circulant property and is most suitable for the hardware realization. We will show that there exists a close relationship between the CCS and the DCT. In such a case, a 2^m length DCT can be decomposed recursively into shorter length CCS and DCT. This new approach results in very simple and straightforward structure and gives the minimum number of multiplications for its realization.**

I. .

Many algorithms for the computation of the discrete cosine transform (DCT) have been proposed since the introduction of the DCT in 1974 by Ahmed *et al.* [1]. These algorithms can broadly be classified into two groups: 1) indirect computation through the fast discrete Fourier transform and the Walsh Hadamard transform [2], [3], and 2) direct computation of DCT through matrix decomposition or recursive computation [4]–[8]. Among them, Lee's [4], Hou's [5], and Vetterli's [7] algorithms meet the minimum known number of multiplications to implement a length 2^m DCT algorithm.

The convolution structure plays an important role in digital signal processing due to its regularity and simplicity during its hardware implementation [9]. Since a correlation of two sequences can always be converted into a convolution by inverting one of the input sequences, the correlation is also widely used in digital signal processing. In fact, there is a number of DFT algorithms developed depending upon these two structures [10], [11].

In this paper, we first introduce a structure called the correlated cosine structure (CCS), and then present a new algorithm to compute an $N = 2^m$ length DCT by decomposing it into shorter length CCS and DCT recursively. Then we will elaborate the correlation property of the CCS. Finally, we will show that the proposed algorithm enables us to realize the DCT with the least number of multiplications compared with conventional approaches, and it also results in extremely regular structure such that its realization is very simple.

II. Algorithm Derivation

The DCT [1] of a real data sequence $\{x(i): i = 0, 1, \cdots, N-1\}$ is defined as

$$X(k) = \sum_{i=0}^{N-1} x(i) \cos\left[2\pi(2i+1)k/4N\right], \quad \text{for } k = 0, 1, \cdots, N-1. \tag{1}$$

Let us split $X(k)$ into odd and even sequences.

Manuscript received November 20, 1989; revised April 19, 1991 and September 13, 1991. This paper was recommended by Associate Editor E. J. Coyle.

The authors are with the Computer Engineering Section, Department of Electronic Engineering, Hong Kong Polytechnic, Hung Hom, Kowloon, Hong Kong.

IEEE Log Number 9105218.

$$X(2k) = \sum_{i=0}^{N/2-1} \left(x(i) + x(N-1-i)\right) \cos\left[2\pi(2i+1)k/2N\right], \quad \text{for } k = 0, 1, \cdots, N/2-1 \tag{2}$$

and

$$X(2k+1) = \sum_{i=0}^{N-1} x'(i) \cos\left[2\pi(4i+1)(2k+1)/4N\right], \quad \text{for } k = 0, 1, \cdots, N/2-1 \tag{3}$$

where

$$x'(i) = x(2i)$$

$$x'(N-1-i) = x(2i+1), \quad \text{for } i = 0, 1, \cdots, N/2-1. \tag{4}$$

By defining the N-point CCS on $x'(i)$:

$$X'(k) = \sum_{i=0}^{N-1} x'(i) \cos\left[2\pi(4i+1)(4k+1)/4N\right], \quad \text{for } k = 0, 1, \cdots, N-1 \tag{5}$$

we have

$$X(4k+1) = X'(k)$$

$$X(4k+3) = -X'(N/2-1-k), \quad \text{for } k = 0, 1, \cdots, N/4-1. \tag{6}$$

Hence an N-length DCT can be decomposed into an N-length CCS and an $N/2$-length DCT at a cost of $N/2$ additions only. Now we will prove that an N-length CCS can be further decomposed into an $N/2$-length CCS and an $N/4$-length DCT at a cost of $N/2$ multiplications and $5N/4 - 1$ additions.

Consider the structure CCS again. This structure has an interesting property of $X'(N/2+k) = -X'(k)$. Besides, it can be decomposed into

$$X'(k) = \sum_{i=0}^{N/2-1} y(i) \cos\left[2\pi(4i+1)(4k+2)/4N\right] + \sum_{i=0}^{N/2-1} y(i) \cos\left[2\pi(4i+1)(4k)/4N\right] = P(k) + G(k) \tag{7}$$

1057-7130/92$03.00 © 1992 IEEE

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on September 30, 2009 at 02:32 from IEEE Xplore. Restrictions apply.

where

$$y(i) = \big(x'(i) - x'(N/2 + i)\big) \sec\big[\pi(4i + 1)/2N\big]/2, \quad \text{for } i = 0, 1, \cdots, N/2 - 1. \quad (8)$$

The second term $G(k)$ is actually an $N/4$-length DCT of the data sequence $\{g(i)\colon i = 0, 1, \cdots, N/4 - 1\}$ where $g(i)$ is given by

$$g(2i) = y(i) + y(N/4 + i)$$
$$g(2i + 1) = y(N/4 - i - 1) + y(N/2 - i - 1), \quad \text{for } i = 0, 1, \cdots, N/8 - 1. \quad (9)$$

The first term $P(k)$ can be computed indirectly by computing another $N/2$-length CCS, $F(k)$, on $y(i)$:

$$F(k) = \sum_{i=0}^{N/2-1} y(i) \cos\big[2\pi(4i + 1)(4k + 1)/2N\big], \quad \text{for } k = 0, 1, \cdots, N/2 - 1. \quad (10)$$

The relationship between $F(k)$ and $P(k)$ is given by

$$P(2k) = F(k)$$
$$P(N/2 - 2k - 1) = -F(k) = F(N/4 + k), \quad \text{for } k = 0, 1, \cdots, N/4 - 1. \quad (11)$$

Hence an N-length CCS can be decomposed into an $N/2$-length CCS and an $N/4$-length DCT.

III. Properties of CCS

The CCS may be considered as a special case of the discrete symmetric cosine transform (DSCT) [12], and the properties of which were first described by Ersoy in [12]. It is very interesting that the structure CCS can be converted into circular correlation form easily. Specifically, by defining a bijective mapping [8] on the set INDEX = $(i\colon i = 0, 1, \cdots, N - 1)$ to itself, where $N = 2^m$:

$$\langle 5^v \rangle_{4N} = 4u + 1, \text{ where } u, v \in 0, 1, \cdots, N - 1 \quad (12)$$

Equation (5) can be rewritten as

$$X''(k) = \sum_{i=0}^{N-1} x''(i) \cos\big[(2\pi/4N)\langle 5^{i+k}\rangle_{4N}\big] \quad (13)$$

where $X''(k) = X'(((5^k)_{4N} - 1)/4)$ and $x''(i) = x'(((\langle 5^i\rangle_{4N} - 1)/4)$.

We note that this is a circular correlation. Moreover, we can make a further simplification on this equation. Consider the correlation $X''(k)$ in (13) and let $C(n) = \cos[(2\pi/4N)\langle 5^n\rangle_{4N}]$, one may observe that $C(N/2 + n) = -C(n)$ for $n = 0, 1, \cdots, N/2 - 1$. Besides, as $X''(N/2 + n) = -X''(n)$ for $n = 0, 1, \cdots, N/2 - 1$, only $X''(0), \cdots, X''(N/2 - 1)$ are required to be computed. Hence we have

$$\begin{bmatrix} X''(0) \\ X''(1) \\ \vdots \\ X''(N/2 - 1) \end{bmatrix} = \begin{bmatrix} C(0) & C(1) & \cdots & C(N/2 - 1) \\ C(1) & C(2) & \cdots & -C(0) \\ \vdots & \vdots & & \vdots \\ C(N/2 - 1) & \cdots & & -C(N/2 - 2) \end{bmatrix} \cdot \begin{bmatrix} x''(0) - x''(N/2) \\ x''(1) - x''(N/2 + 1) \\ \vdots \\ x''(N/2 - 1) - x''(N - 1) \end{bmatrix}. \quad (14)$$

Note that this is a skew-circular correlation. This structure can be realized on a typical correlation hardware with only a small modification.

IV. DCT Realization

We now further prove the possibility of decomposing an N-length DCT into two N-length CCS's. Recall in Section II that the odd sequence of an N-length DCT can be converted into an N-length CCS. Let us also deal with the even sequence. From (1) and (4), we have the even sequence $X(2k)$:

$$X(2k) = \sum_{i=0}^{N-1} x'(i) \cos\big[2\pi(4i + 1)(2k)/4N\big], \quad \text{for } k = 0, 1, \cdots, N/2 - 1. \quad (15)$$

If we let

$$y'(i) = 2x'(i) \cos\big[(2\pi/4N)(4i + 1)\big], \quad \text{for } i = 0, 1, \cdots, N - 1 \quad (16)$$

then

$$X(2k) + X(2k + 2) = Y(k), \text{ for } k = 0, 1, \cdots, N/2 - 1 \quad (17)$$

where

$$Y(k) = \sum_{i=0}^{N-1} y'(i) \cos\big[2\pi(4i + 1)(2k + 1)/4N\big], \quad \text{for } k = 0, 1, \cdots, N/2 - 1. \quad (18)$$

As $X(N) = 0$, we can find all $X(2k)$'s by solving the equation set (17), which involves $N/2 - 1$ additions only. Note that $Y(k)$ is exactly in the form of (3). Hence it can be converted into an N-length CCS by similar techniques.

Hence it is obvious that for any length-2^n DCT, we can decompose it into a number of given length, say 2^m, CCS's such that we can have a suitable size of CCS's for any particular hardware realization.

V. An Example

Consider the case where we have a modified correlation hardware that can realize (14) with $N = 8$. Then let us clarify our proposal using a length 16 DCT with input sequence $\{x(i)\colon i = 0, 1, \cdots, 15\}$ by making use of this 4-point module. We first rearrange the data as shown in (4), (8), and (9):

$$y(0) = \big(x(0) - x(15)\big) \sec a/2$$
$$y(1) = \big(x(2) - x(13)\big) \sec 5a/2$$
$$y(2) = \big(x(4) - x(11)\big) \sec 9a/2$$
$$y(3) = \big(x(6) - x(9)\big) \sec 13a/2$$
$$y(4) = \big(x(8) - x(7)\big) \sec 17a/2$$
$$y(5) = \big(x(10) - x(5)\big) \sec 21a/2$$
$$y(6) = \big(x(12) - x(3)\big) \sec 25a/2$$
$$y(7) = \big(x(14) - x(1)\big) \sec 29a/2$$

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on September 30, 2009 at 02:32 from IEEE Xplore. Restrictions apply.

and where $a = \pi/32$,

$$g(0) = y(0) + y(4)$$

$$g(1) = y(3) + y(7)$$

$$g(2) = y(1) + y(5)$$

$$g(3) = y(2) + y(6).$$

Then, from (6), (7), and (11), we have

$$\begin{bmatrix} X(1) \\ X(3) \\ X(5) \\ X(7) \\ X(9) \\ X(11) \\ X(13) \\ X(15) \end{bmatrix} = \begin{bmatrix} X'(0) \\ -X'(7) \\ X'(1) \\ -X'(6) \\ X'(2) \\ -X'(5) \\ X'(3) \\ -X'(4) \end{bmatrix} = \begin{bmatrix} P(0) + G(0) \\ -P(7) - G(7) \\ P(1) + G(1) \\ -P(6) - G(6) \\ P(2) + G(2) \\ -P(5) - G(5) \\ P(3) + G(3) \\ -P(4) - G(4) \end{bmatrix}$$

$$= \begin{bmatrix} F(0) + G(0) \\ F(0) + G(1) \\ -F(3) + G(1) \\ -F(3) + G(2) \\ F(1) + G(2) \\ F(1) + G(3) \\ -F(2) + G(3) \\ -F(2) \end{bmatrix}$$

where $\{F(k)\colon k = 0, 1, \cdots, 7\}$ and $\{G(k)\colon k = 0, 1, \cdots, 3\}$ are given by

$$\begin{bmatrix} G(0) \\ G(1) \\ G(2) \\ G(3) \end{bmatrix} = \begin{bmatrix} \cos 0 & \cos 0 & \cos 0 & \cos 0 \\ \cos 2e & \cos 6e & \cos 10e & \cos 14e \\ \cos 4e & \cos 12e & \cos 20e & \cos 28e \\ \cos 6e & \cos 18e & \cos 30e & \cos 42e \end{bmatrix} \begin{bmatrix} g(0) \\ g(1) \\ g(2) \\ g(3) \end{bmatrix}$$

and

$$\begin{bmatrix} F(0) \\ F(1) \\ -F(2) \\ -F(3) \end{bmatrix} = \begin{bmatrix} \cos e & \cos 5e & \cos 25e & \cos 29e \\ \cos 5e & \cos 25e & \cos 29e & -\cos e \\ \cos 25e & \cos 29e & -\cos e & -\cos 5e \\ \cos 29e & -\cos e & -\cos 5e & -\cos 25e \end{bmatrix} \cdot \begin{bmatrix} (y(0) - y(4)) \\ (y(1) - y(5)) \\ (y(6) - y(2)) \\ (y(7) - y(3)) \end{bmatrix} \text{ where } e = \pi/16.$$

Note that this is in cyclic form, which is very suitable for distributed arithmetic [13] and VLSI realization.

From (2), we know that the even sequence $X(2k)$ is actually an 8-point DCT on sequence $\{x(i) + x(15 - i)\colon i = 0, 1, \cdots, 7\}$, which can be decomposed into two 8-point CCS's. We first permute the input sequence using (4) and (16):

$$\begin{bmatrix} y_0(0) \\ y_0(1) \\ y_0(2) \\ y_0(3) \\ y_0(4) \\ y_0(5) \\ y_0(6) \\ y_0(7) \end{bmatrix} = \begin{bmatrix} x(0) + x(15) \\ x(2) + x(13) \\ x(4) + x(11) \\ x(6) + x(9) \\ x(7) + x(8) \\ x(5) + x(10) \\ x(3) + x(12) \\ x(1) + x(14) \end{bmatrix},$$

$$\begin{bmatrix} y_0'(0) \\ y_0'(1) \\ y_0'(2) \\ y_0'(3) \\ y_0'(4) \\ y_0'(5) \\ y_0'(6) \\ y_0'(7) \end{bmatrix} = \begin{bmatrix} 2(x(0) + x(15))\cos e \\ 2(x(2) + x(13))\cos 5e \\ 2(x(4) + x(11))\cos 9e \\ 2(x(6) + x(9))\cos 13e \\ -2(x(7) + x(8))\cos e \\ -2(x(5) + x(10))\cos 5e \\ -2(x(3) + x(12))\cos 9e \\ -2(x(1) + x(14))\cos 13e \end{bmatrix}.$$

From (3), (5), and (6), using also the bijective mapping (12) and the simplification given in (14), the odd sequence of this 8-point DCT can be written as

$$\begin{bmatrix} X(2) \\ X(10) \\ X(14) \\ X(6) \end{bmatrix} = \begin{bmatrix} \cos e & \cos 5e & \cos 25e & \cos 29e \\ \cos 5e & \cos 25e & \cos 29e & -\cos e \\ \cos 25e & \cos 29e & -\cos e & -\cos 5e \\ \cos 29e & -\cos e & -\cos 5e & -\cos 25e \end{bmatrix} \cdot \begin{bmatrix} y_0(0) - y_0(4) \\ y_0(1) - y_0(5) \\ y_0(6) - y_0(2) \\ y_0(7) - y_0(3) \end{bmatrix}.$$

Similarly, from (15), (17), (18), and (5), the mapping (12) and the simplification given in (14), the even sequence of this 8-point DCT can be computed by

$$\begin{bmatrix} X(0) + X(4) \\ X(8) + X(12) \\ X(12) \\ X(4) + X(8) \end{bmatrix} = \begin{bmatrix} \cos e & \cos 5e & \cos 25e & \cos 29e \\ \cos 5e & \cos 25e & \cos 29e & -\cos e \\ \cos 25e & \cos 29e & -\cos e & -\cos 5e \\ \cos 29e & -\cos e & -\cos 5e & -\cos 25e \end{bmatrix} \cdot \begin{bmatrix} 2(y_0(0) + y_0(4))\cos e \\ 2(y_0(1) + y_0(5))\cos 5e \\ 2(y_0(6) + y_0(2))\cos 25e \\ 2(y_0(7) + y_0(3))\cos 29e \end{bmatrix}.$$

Note that these two matrices are also in cyclic form now and hence can be ported into the 4-point module. Furthermore, the sequence $G(k)$ is actually a 4-point DCT on the sequence $\{g(i)$:

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on September 30, 2009 at 02:32 from IEEE Xplore. Restrictions apply.

$i = 0, 1, 2, 3\}$. *By using similar technique as on the 8-point DCT shown above, we have*

$$\begin{bmatrix} G(0) + G(2) \\ -G(2) \end{bmatrix} = \begin{bmatrix} \cos\theta & \cos 5\theta \\ \cos 5\theta & -\cos\theta \end{bmatrix} \begin{bmatrix} 2(g(0) + g(3))\cos\theta \\ 2(g(2) + g(1))\cos 5\theta \end{bmatrix}$$

and

$$\begin{bmatrix} G(1) \\ -G(3) \end{bmatrix} = \begin{bmatrix} \cos\theta & \cos 5\theta \\ \cos 5\theta & -\cos\theta \end{bmatrix} \begin{bmatrix} g(0) - g(3) \\ g(2) - g(1) \end{bmatrix} \text{ where } \theta = \pi/8.$$

VI. Results and Comparisons

This new algorithm has three advantages.

First, in Section II, we have proved that any $N = 2^m$ length DCT can be decomposed into shorter length CCS's and DCT's. If we ignore the correlation property of the CCS and recursively use the algorithm in Section II, we can have the following equations that give the multiplicative complexity of the algorithm in Section II:

$$\begin{aligned} M(N - \text{DCT}) &= M(N - \text{CCS}) + M(N/2 - \text{DCT}) \\ A(N - \text{DCT}) &= A(N - \text{CCS}) + A(N/2 - \text{DCT}) + N/2, \quad \text{for } N \geq 4 \\ M(2 - \text{DCT}) &= 1 \\ A(2 - \text{DCT}) &= 2 \\ M(N - \text{CCS}) &= M(N/2 - \text{CCS}) + M(N/4 - \text{DCT}) + N/2 \\ A(N - \text{CCS}) &= A(N/2 - \text{CCS}) + A(N/4 - \text{DCT}) \\ &\quad + 5N/4 - 1, \quad \text{for } N > 4 \\ M(4 - \text{CCS}) &= 3 \\ A(4 - \text{CCS}) &= 5 \\ M(2 - \text{CCS}) &= 1 \\ A(2 - \text{CCS}) &= 1. \end{aligned} \tag{19}$$

In a nonrecursive form, we have

$$\begin{aligned} M(N - DCT) &= (N/2)\log_2 N \\ A(N - DCT) &= (3N/2)\log_2 N - N + 1 \\ M(N - CCS) &= (N/4)\log_2 2N \\ A(N - CCS) &= (3N/4)\log_2 N - N/4. \end{aligned} \tag{20}$$

The multiplicative complexity of this algorithm is comparable to that of Lee's [2] and Vetterli's [3] approaches. Table I shows the computational complexity to realize a CCS and a DCT by using the algorithm in Section II. Table II shows the comparison of computational complexity among selected algorithms.

TABLE I
Computational Complexity of CCS and DCT by Using New Algorithm in Section II

	CCS		DCT	
N	M	A	M	A
2	1	1	1	2
4	3	5	4	9
8	8	16	12	29
16	20	44	32	81
32	48	112	80	209
64	112	272	192	513
128	256	640	448	1217
256	576	1472	1024	2817

TABLE II
Comparison of the Computational Complexity among Selected Algorithms for the Realization of an N-Point DCT

	Present Method		Lee's Method		Vetterli's Method	
N	M	A	M	A	M	A
8	12	29	12	29	12	29
16	32	81	32	81	32	81
32	80	209	80	209	80	209
64	192	513	192	513	192	513
128	448	1217	448	1217	448	1217
256	1024	2817	1024	2817	1024	2817
512	2304	6401	2304	6401	2304	6401

Secondly, we can see that the CCS is most suitable for hardware realization due to its correlation property. Hence we may decompose a 2^m-length DCT into two 2^m-length CCS's by using the decomposition algorithm shown in Section IV instead of that shown in Section II such that it can be easily realized using the dedicated correlation hardware.

Finally, we can decompose a long-length DCT into acceptable length CCS's such that we can deal with it with a fixed-problem-size machine. We propose two methods to do this. Method one uses the decomposition algorithm shown in Sections II and IV recursively until the problem can be ported into the hardware. An example of this approach is shown in Section V. Method two is to decompose an N-length DCT into two $N/2$-length DCT's recursively and then change those DCT's after decomposition into CCS's by using the algorithm in Section IV. However, method one is more efficient as it requires fewer multiplications.

Let us illustrate the point with a simple example. Consider that we have a 16-point DCT and a 4-point modified correlation hardware mentioned in Section V again:

Method 1:			
step 1: 16-DCT	→	8-DCT(1) + 16-CCS(1)	with no cost of multiplication
step 2: 8-DCT(1)	→	8-CCS(2) + 8-CCS(3)	with a cost of 4 multiplications
step 3: 16-CCS(1)	→	8-CCS(4) + 4-DCT(2)	with a cost of 8 multiplications
step 4: 4-DCT(2)	→	4-CCS(5) + 4-CCS(6)	with a cost of 2 multiplications
Total: 16-DCT	→	3*8-CCS + 2*4-CCS	with a cost of 14 multiplications
Method 2:			
step 1: 16-DCT	→	8-DCT(1) + 8-DCT(2)	with a cost of 8 multiplications
step 2: 8-DCT(1)	→	8-CCS(1) + 8-CCS(2)	with a cost of 4 multiplications
step 3: 8-DCT(2)	→	8-CCS(3) + 8-CCS(4)	with a cost of 4 multiplications
Total: 16-DCT	→	4*8-CCS	with a cost of 16 multiplications.

Obviously, to realize an 8-point CCS needs more multiplications than to realize two 4-point CCS's. Besides, the multiplication cost

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on September 30, 2009 at 02:32 from IEEE Xplore. Restrictions apply.

of method one is less than that of method two. Hence, decomposition method one is much better, especially when N is large.

VIII. Conclusion

In this paper, the relationship between the DCT and the CCS is discussed. Based on this relationship, a new algorithm is proposed to compute a 2^m-length DCT via the CCS. This algorithm is one of the possible approaches that require the least multiplicative complexity for the realization, and its regularity makes it very suitable for VLSI realization as well. Besides, we have proposed two methods to decompose a 2^m-length DCT into any desirable length CCS's such that we can port them into a machine with a fixed-problem-size.

References

[1] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. C-23, pp. 90–94, Jan. 1974.

[2] J. Makoul, "A fast cosine transform in one and two dimensions," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 27–34, Feb. 1980.

[3] M. J. Narasimha and A. M. Peterson, "On the computation of the discrete cosine transform," *IEEE Trans. Commun.*, vol. COM-26, pp. 934–946, June 1978.

[4] B. G. Lee, "A new algorithm to compute the discrete cosine transform," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 1243–1245, Dec. 1984.

[5] H. S. Hou, "A fast recursive algorithm for computing the discrete cosine transform," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, pp. 1455–1461, Oct. 1987.

[6] Z. Wang, "On computing the discrete Fourier and cosine transforms," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, pp. 1341–1344, Oct. 1985.

[7] M. Vetterli and H. Nussbaumer, "A simple FFT and DCT algorithms with reduced number of operation," *Signal Processing*, vol. 6, pp. 267–278.

[8] P. Duhamel and H. H'mida, "New 2^n DCT algorithms suitable for VLSI implementation," in *Proc. ICASSP 85*, pp. 780–783, Mar. 1985.

[9] O. K. Ersoy, "Semisystolic array implementation of circular, skew circular, and linear convolutions," *IEEE Trans. Comput.*, vol. C-34, pp. 190–196, 1985.

[10] C. M. Rader, "Discrete Fourier transforms when the number of data samples is prime," *Proc. IEEE*, vol. 56, pp. 1107–1108, 1968.

[11] O. K. Ersoy, "A two-stage representation of DFT and its applications," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, pp. 825–831, June 1987.

[12] ——, "On relating discrete Fourier, sine, and symmetric cosine transforms," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, pp. 219–222, Feb. 1985.

[13] S. A. White, "Applications of distributed arithmetic to digital signal processing: A tutorial review," *IEEE ASSP Mag.*, vol. 6, pp. 4–19, July 1989.

Convergence Analysis and Design of an Adaptive Filter with Finite-Bit Power-of-Two Quantized Error

Eweda Eweda

Abstract—**This paper is concerned with the analysis and design of an adaptive filter governed by an LMS algorithm with finite-bit power-of-two quantization of the error signal. Both the input data and the optimal filter weights are assumed stationary. An expression of the steady-state error power is derived. According to this expression, the error power is linearly increasing in the step size μ and exponentially decreasing in the number of quantizer bits B. A practically interesting result is the derivation of a threshold value of B above which the error power is constant versus B. The threshold is a decreasing function of the noise power. Expressions of B and μ that achieve a given tolerable value of the error power with the fastest convergence and the minimum hardware complexity are provided.**

Manuscript received July 23, 1990; revised October 7, 1991. This paper was recommended by Associate Editor D. Graupe.

The author is with Military Technical College, Kobry El-Kobba, Cairo, Egypt.

IEEE Log Number 9105219.

I. Introduction

Quantization of input signals of the correlation multiplier in adaptive filtering considerably reduces the hardware requirements of the adaptation algorithm [4]. One of the algorithms that uses this fact is the power-of-two quantizer algorithm [1], which will be the subject of this paper. To be specific, consider the case of adaptive identification of an unknown plant whose output a_k is related to its input x_k by

$$a_k = \sum_{i=0}^{N-1} g_i x_{k-i} + b_k \tag{1.1}$$

where g_i; $i = 0, 1, \cdots, N-1$ are the parameters of the plant and b_k is the plant noise. Define the parameter vector $\boldsymbol{G}$ and the observation vector $\boldsymbol{X}_k$ as

$$\boldsymbol{G}^T \triangleq (g_0, g_1, \cdots, g_{N-1}) \tag{1.2}$$

$$\boldsymbol{X}_k^T \triangleq (x_k, x_{k-1}, \cdots, x_{k-N+1}) \tag{1.3}$$

where the superscript T denotes the transpose. Then (1.1) becomes

$$a_k = \boldsymbol{G}^T \boldsymbol{X}_k + b_k. \tag{1.4}$$

Throughout the paper, it will be assumed that the plant input $\{x_k\}$ *and the plant noise* $\{b_k\}$ *are mutually independent zero-mean stationary Gaussian random processes.* Due to (1.4), this assumption implies that the sequences $\{a_k\}$ and $\{\boldsymbol{X}_k\}$ are zero mean, jointly stationary, and Gaussian. Identification of the plant is made by an adaptive filter whose weight vector, $\boldsymbol{H}_k$, is updated according to the well-known least mean square (LMS) algorithm [5]

$$\boldsymbol{H}_{k+1} = \boldsymbol{H}_k + \mu e_k \boldsymbol{X}_k; \qquad \mu > 0 \tag{1.5}$$

$$e_k \triangleq a_k - \boldsymbol{H}_k^T \boldsymbol{X}_k. \tag{1.6}$$

Quantized LMS algorithms can use quantization of the estimation error e_k, the regressor $\boldsymbol{X}_k$, or both of them. However, it has been shown in [7] that the quantization of $\boldsymbol{X}_k$ may lead to the instability of the algorithm. Therefore, in this paper we shall consider the algorithm with power-of-two quantization of e_k only. This algorithm is given by

$$\boldsymbol{H}_{k+1} = \boldsymbol{H}_k + \mu \boldsymbol{X}_k q(e_k) \tag{1.7}$$

where the power-of-two quantizer $q(\cdot)$ is defined by [4]

$$q(u) = 2^{[\log_2 |u|]} \operatorname{sgn}(u) \tag{1.8}$$

where $\operatorname{sgn}(u)$ is the sign of u, and $[b]$ is the largest integer less than b. The quantizer defined by (1.8) is an infinite bit quantizer. In applications, however, a finite bit quantizer is used. The first analysis of the finite bit power-of-two quantizer (FBQ) algorithm is

1057-7130/92$03.00 © 1992 IEEE

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on September 30, 2009 at 02:32 from IEEE Xplore. Restrictions apply.