

than the data path butterfly calculation hardware, it is advisable to make changes as outlined above only to the address generation logic and leave the data path unchanged if possible.

#### ACKNOWLEDGMENT

The author would like to thank Dr. Yiwan Wong for many stimulating conversations, and Dr. Robert Hewes, director of the VLSI Design Laboratory at Texas Instruments, Inc., for support during his recent sabbatical.

#### REFERENCES

- [1] J. A. Eldon, "Fourier on a chip," *VLSI Signal Processing*, vol. III, R. W. Brodersen, H. S. Moscovitz, Eds. IEEE Press, pp. 15–20, 1988.
- [2] S. Shen *et al.*, "A high performance CMOS chipset for FFT processors," in *Proc. IEEE Conf. Comp., Design, VLSI in Computers*, pp. 578–581, May 1988.
- [3] J. O'Brien, J. Mather, and B. Holland, "A 200 MIPS single-chip 1K FFT processor," in *Proc. ISSCC*, pp. 166–167, Feb. 1989.
- [4] Y. Wong, W. Gass, T. Yoshino, and L. Johnson, "Silicon compilation of fast fourier transforms," Presented at the 1990 IEEE Workshop on VLSI Signal Processing at San Diego, Nov. 1990.
- [5] A. V. Oppenheim and R. W. Schaffer, *Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975, ch. 6.

### Fast Filter Bank (FFB)

Y. C. Lim and B. Farhang-Boroujeny

**Abstract**—The sliding FFT filter bank has an exceedingly low complexity of one multiplication per channel per sample. However, its frequency selectivity and passband response are poor. These limit its suitability to only those applications where good selectivity and flat passband response are not necessary. In this paper, we show that the sliding FFT filter bank is in fact a particular member of a new family of fast filter bank (FFB). In the case of FFT, each cluster of butterflies can in fact be derived from a pair of complementary two-tap (i.e., first-order) prototype FIR filters. The poor selectivity and degraded passband response of the FFT filter bank is a direct consequence of the poor frequency response of the prototype first-order filter. In this paper, we show that by increasing the order of the prototype filters, it is possible to implement a filter bank with arbitrarily good selectivity and flat passband response. The FFB retains the low-complexity feature of the FFT. Because of its very much improved frequency response characteristics, the FFB will be suitable for use in many applications where the FFT filter bank is unsuitable.

#### I. INTRODUCTION

The usefulness of the sliding FFT as a filter bank is well known [1]. Its extremely low complexity of one complex multiplication per channel per sample [2] and its regularity are two of its strongest advantages. The most serious disadvantages of the sliding FFT as a filter bank are its poor passband response and high side-lobes in its stopbands. The peak of the first side-lobe is approximately  $-13$  dB, irrespective of the FFT length. Increasing the FFT length will result in a narrower main-lobe width but has little effect on the side-lobe magnitudes. In this paper, we present a new family of filter bank called the fast filter bank (FFB) whose complexity is only very slightly higher than that of the FFT, but its frequency response

Manuscript received June 16, 1991; revised December 16, 1991. This paper was recommended by Associate Editor Y. F. Huang.

The authors are with the Electrical Engineering Department, National University of Singapore, Singapore 0511.

IEEE Log Number 9108289.

characteristics are much better. The FFB is realized in the form of a system of filters organized in a parallel tree structure.

#### II. THE FFT FILTER BANK

Fig. 1 shows the structure of an eight-point FFT. It can be deduced from the first-level butterflies of Fig. 1 that

$$x^{1,0}(n) = x^{0,0}(n) + W_8^0 x^{0,0}(n-4) \quad (1a)$$

$$x^{1,1}(n) = x^{0,0}(n) - W_8^0 x^{0,0}(n-4) \quad (1b)$$

where

$$W_N^k = e^{-j2\pi k/N}. \quad (1c)$$

When the FFT is used as a filter bank, one complex multiplication is needed to compute both  $x^{1,0}(n)$  and  $x^{1,1}(n)$  every sampling interval. Let  $X^{i,j}(z)$  be the  $z$ -transform of  $x^{i,j}(n)$ . Taking  $z$ -transform, (1) becomes

$$X^{1,0}(z) = H_a^{0,0}(z) X^{0,0}(z) \quad (2a)$$

$$X^{1,1}(z) = H_c^{0,0}(z) X^{0,0}(z) \quad (2b)$$

where

$$H_a^{0,0}(z) = 1 + W_8^0 z^{-4} \quad (3a)$$

$$H_c^{0,0}(z) = 1 - W_8^0 z^{-4}. \quad (3b)$$

Note that  $H_a^{0,0}(z)$  and  $H_c^{0,0}(z)$  form a complementary pair such that  $H_a^{0,0}(z) + H_c^{0,0}(z) = 2$ .

Similarly, from the second- and third-level butterflies, we have

$$X^{2,0}(z) = H_a^{1,0}(z) X^{1,0}(z) \quad (4a)$$

$$X^{3,0}(z) = H_a^{2,0}(z) X^{2,0}(z) \quad (4b)$$

where

$$H_a^{1,0}(z) = 1 + W_8^0 z^{-2} \quad (5a)$$

$$H_a^{2,0}(z) = 1 + W_8^0 z^{-1}. \quad (5b)$$

From Fig. 1 it is clear that the transfer function of channel-zero of the sliding FFT filter bank is  $H_a^{0,0}(z)H_a^{1,0}(z)H_a^{2,0}(z)$ . The frequency responses of  $H_a^{0,0}(z)$ ,  $H_a^{1,0}(z)$ ,  $H_a^{2,0}(z)$ , and  $H_a^{0,0}(z)H_a^{1,0}(z)H_a^{2,0}(z)$  are shown in Fig. 2.

From (3) and (5), we note that  $H_a^{0,0}(z)$ ,  $H_a^{1,0}(z)$ , and  $H_a^{2,0}(z)$  are derived from the first-order FIR filter  $1 + W_8^0 z^{-1}$  by replacing  $z$  by  $z^4$ ,  $z^2$ , and  $z$ , respectively. In fact, it can be shown that the sliding FFT filter bank is constructed entirely from subfilters derived from the first-order FIR filters  $1 + z^{-1}$  and  $1 - z^{-1}$  by replacing  $z$  by a suitable integer power of  $W_N^k z$ . The coefficient  $W_N^k$  provides the necessary frequency translation. Note also that in the FFT filter bank  $H_a^{i,j}(z) + H_c^{i,j}(z) = 2$ . As a consequence, for an  $N$ -tap FFT filter bank, the sum of all the channel outputs is equal to  $N$  times the input.

We would like to point out the following. The sliding FFT filter bank implementation technique discussed in [1] involves assigning poles on the unit circle. These poles are the potential cause of hazards. The complexity of one complex multiplication per channel

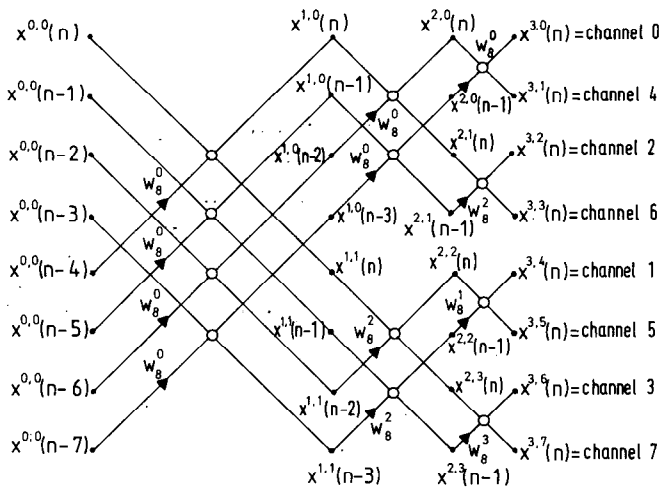
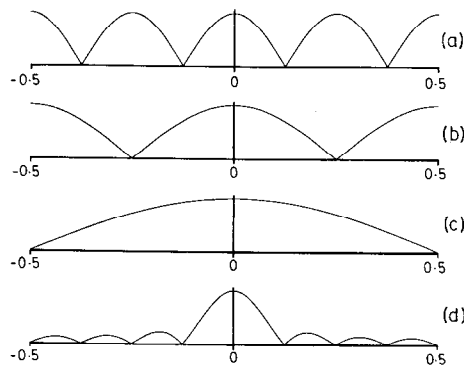


Fig. 1. An eight-point FFT used as an eight-channel filter bank.


 Fig. 2. The frequency responses of  $H_a^{0,0}(z)$ ,  $H_a^{1,0}(z)$ ,  $H_a^{2,0}(z)$ , and  $H_a^{3,0}(z)$  ( $H_a^{1,0}(z)H_a^{2,0}(z)$ ) are shown in (a)–(d), respectively.

per sample can be achieved without stability problem by using a true FIR structure [2] the generalization of which is shown in Fig. 3.

### III. FFB—THE FAST FILTER BANK

It is clear from the discussion presented in Section II that the poor frequency response performance of the FFT filter bank is due to the poor frequency response performance of the pair of complementary first-order prototype FIR filters,  $1 + z^{-1}$  and  $1 - z^{-1}$ . An obvious way to improve the performance of the FFT filter bank is to replace the pair of complementary first-order prototype filters by higher order prototype complementary filters. Let  $H_a^i(z)$  and  $H_c^i(z)$  be the pair of higher order complementary prototype filters replacing the  $i$ th level butterflies of the FFT. In order to reduce the hardware complexity of the proposed structure, we shall choose  $H_a^i(z)$  to be an odd length symmetrical impulse response half-band filter. In this case,  $H_a^i(z)$  has the property that its every other coefficient value is zero. Hence, the number of distinct nontrivial multipliers is only about one-quarter of its filter length. In order to simplify discussion, we shall further assume that  $H_a^i(z)$  is noncausal; the center of symmetry of its impulse response occurs at time zero, i.e.,

$$H_a^i(z) = \sum_{n=-\infty}^{\infty} h_a^i(n) z^{-n} \quad (6a)$$

$$h_a^i(n) = h_a^i(-n) \quad (6b)$$

where  $h_a^i(n)$  is the impulse response of  $H_a^i(z)$  at time  $n$ . (Causality can be reinstated if necessary by delaying the noncausal impulse

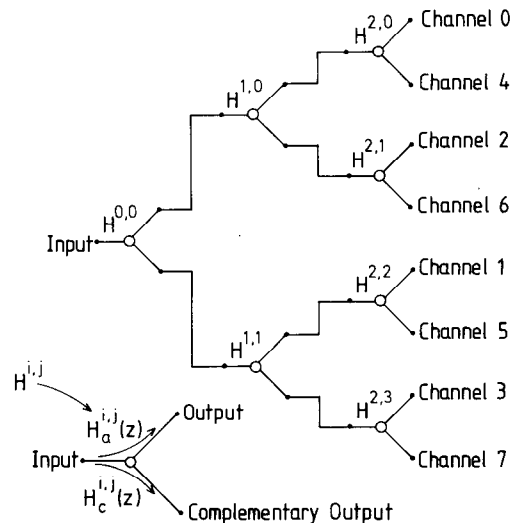


Fig. 3. The structure of an FFB (analysis FFB).

response by an appropriate number of samples.) With the above assumption,  $H_a^i(z)$  and  $H_c^i(z)$  satisfy the relationship

$$H_a^i(z) + H_c^i(z) = 2. \quad (7)$$

Furthermore,  $H_a^i(z)$  is low-pass and  $H_c^i(z)$  is high-pass. Referring to Fig. 3, for an  $N$ -channel filter bank where  $N = 2^L$ ,  $H_a^{i,j}(z)$  and  $H_c^{i,j}(z)$  for all  $i$  and  $j$  are obtained by replacing each  $z$  in  $H_a^i(z)$  and  $H_c^i(z)$ , respectively, by  $W_N^j z^{2^{L-i-j}}$ , where  $\tilde{j}$  is the bit reversed version of  $j$  in  $L - 1$  bits. The output of  $H_c^{i,j}(z)$  is obtained by subtracting the output of  $H_a^{i,j}(z)$  from twice the input.

### IV. THE SYNTHESIS FFB

The FFB structure presented in Fig. 3 is a single-input multiple-output analysis FFB. If  $H_i(z)$  is the transfer function of the  $i$ th channel, then the  $z$ -transform of the  $i$ th channel output,  $Y_i(z)$ , is given by

$$Y_i(z) = H_i(z) X(z) \quad (8)$$

where  $X(z)$  is the  $z$ -transform of the input. In some applications, there are many channels of input signals to be filtered and summed to produce one output. If  $Y(z)$  and  $X_i(z)$  are the  $z$ -transforms of the output signal and the  $i$ th channel input signal, respectively, then

$$Y(z) = \sum_{i=0}^{N-1} H_i(z) X_i(z). \quad (9)$$

This kind of filtering operation can be performed very efficiently using the multiple-input single-output synthesis FFB. Fig. 4 shows the structure of an eight-band synthesis FFB. This is clearly obtained by changing the direction of the flow of data in the structure of Fig. 3.

### V. AN EXAMPLE

We choose a 64-channel FFB to illustrate our technique. There are six levels of butterflies in the FFT. The coefficient values of the prototype filters,  $H_a^i(z)$ , replacing these butterflies are shown in Table I. Only nonzero coefficients are listed in Table I. The coefficient values of  $H_c^i(z)$  can be derived from those of  $H_a^i(z)$  using (7). Fig. 5 shows the frequency response plot of channel-8 of the foregoing example. The peak side-lobe magnitude is  $-56$  dB. The multiplication complexity of this example is 1.34 complex multiplications per channel per sample.

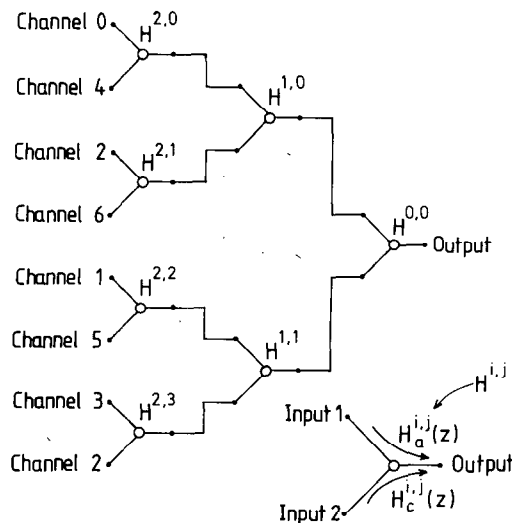


Fig. 4. A synthesis FFB.

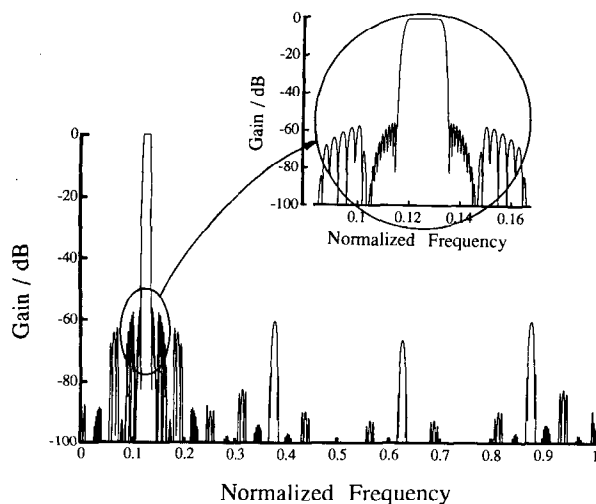


Fig. 5. The frequency response of channel 8 of a 64-channel FFB.

TABLE I  
THE IMPULSE RESPONSES OF  $H_a^i(z)$  FOR THE EXAMPLE IN FIG. 5

$n$	$h_a^1(n)$	$h_a^2(n)$	$h_a^3(n)$	$h_a^4(n)$	$h_a^5(n)$	$h_a^6(n)$
0	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
1, -1	0.62764	0.61750	0.57374	0.56535	0.50191	0.50048
3, -3	-0.18648	-0.16013	-0.07526	-0.06543		
5, -5	0.08816	0.05558				
7, -7	-0.04299	-0.01493				
9, -9	0.01895					
11, -11	-0.00695					

## VI. CONCLUSION

In this paper, it was shown that the sliding FFT consists entirely of subfilters derived from the first-order prototype FIR filter  $1 + z^{-1}$ . If the first-order prototype filter of the FFT is replaced by a higher order prototype filter, a fast filter bank with an arbitrarily sharp frequency response performance and extremely low complexity is produced. The FFB can be further subdivided into two categories, namely, the analysis FFB and the synthesis FFB.

## REFERENCES

- [1] L. R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975, p. 382.
- [2] R. Stasinski, "Adaptive filters in domains of adaptive transform," in *Proc. 1990 Singapore Int. Conf. Comm. Sys.*, pp. 729-733, 1990.

## On the Implementation of the Frequency-Domain LMS Adaptive Filter

A. O. Ogunfunmi and A. M. Peterson

**Abstract**—This paper presents alternative implementations for the frequency-domain LMS adaptive filter. A common implementation method is as a bank of bandpass filters implemented by the discrete Fourier transform (DFT) before the input to the adaptive filter.

One of the new implementations uses the LMS steepest descent algorithm of Widrow and Hoff to compute the DFT. Therefore, the frequency-domain LMS adaptive filter is implemented as a cascade of two parts, with each one running the LMS adaptive algorithm (called the LMS-DFT). This requires fewer computations than the FFT for large filter lengths because the transform part requires  $O(2N)$  computations per input sample where  $N$  is the filter length while the conventional method of taking FFT's requires  $O(N \log N)$  computations per input sample. Simulations of practical applications of frequency-domain LMS adaptive filters are carried out to determine its suitability and comparisons are made with simulations using the FFT. For real input data, another frequency-domain LMS adaptive filter is suggested based on the discrete Hartley transform (DHT). This new frequency-sampling implementation uses the recursive computation of the DHT. This also requires fewer computations for large filter lengths because the transform part requires  $O(2N)$  computations per input sample where  $N$  is the filter length.

## I. INTRODUCTION

It is well known that the convergence speed of time-domain LMS adaptive filters depends on the ratio of the maximum to the minimum eigenvalues of the autocorrelation matrix of the input [16]. Filters having inputs with wide eigenvalue spread often take longer to converge than filters with white noise inputs. To solve this problem, orthogonalization of the input signals was suggested [1], [3], [9], [10], [14]. This led to the so-called frequency-domain adaptive filter. The frequency-domain adaptive filter (or more generally transform-domain adaptive filter) can be implemented using many different transforms to do the orthogonalization. In this paper, we will use the terms transform-domain/frequency-domain interchangeably. We do not discuss the frequency-domain *block* adaptive filter.

In the following sections, we propose a new structure for implementing the frequency-domain LMS adaptive digital filter for real or complex data. It is based on a method of obtaining the "continuous-flow" DFT of a stream of data using the LMS adaptive

Manuscript received February 9, 1990; revised September 3, 1991, and January 31, 1992. This paper was recommended by Associate Editor Y. F. Huang.

A. O. Ogunfunmi is with the Department of Electrical Engineering, Santa Clara University, Santa Clara, CA 95053.

A. M. Peterson is with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305.

IEEE Log Number 9200143.