Overlapped Block Digital Filtering

Ing-Song Lin and Sanjit K. Mitra, *Fellow, IEEE*

***Abstract*—Block digital filtering has been suggested to increase the parallelism of computation and to reduce the computational complexity of digital filtering systems. In this paper the block processing concept is generalized by considering overlapped input and/or output blocks. As an overlapped block digital filter is, in general, a shift-varying system, the conditions for its shift-invariant operation have been developed. These conditions have been exploited to derive computationally efficient shift-invariant block structures. Two types of fast FIR filtering algorithms using the overlapped block filter structures are derived. One is based on the adaptation of fast short-length linear convolution algorithms and the other is based on DFT algorithms. These algorithms not only reduce the computational complexity of filtering operations but also offer modular and parallel structures. Finite wordlength effects of FIR filters implemented using the overlapped block filter structure are also investigated.**

I. Introduction

Block DIGITAL filtering has been suggested to increase the parallelism of computation and to reduce the computational complexity of digital filtering systems [1]–[5]. The basic block diagram of the well-known block digital filter is shown in Fig. 1 in which the input sequence is converted into a series of contiguous blocks of length L by means of a serial-to-parallel converter. Each input block is processed simultaneously by a L-input, L-output block digital filter characterized by a transfer matrix $\boldsymbol{P}(z)$. The output block of which is then converted back into a serial format by means of a parallel-to-serial converter.

In general, a block digital filter is a time-varying system which can be seen from its equivalent multirate representation shown in Fig. 2. It should be noted that in this representation, the samples of the input block are critically down-sampled (i.e., the down-sampling factor is equal to the number of branches) before processing by the block digital filter $\boldsymbol{P}(z)$ whose outputs are again critically up-sampled before being converted into a serial form by the output interleaving structure.

In this paper we generalize the block processing concept by considering overlapped input and output blocks as indicated in Fig. 3 where L represents the input block size, N represents the output block size, and M is the down-sampling (up-sampling) factor. When $L = M$, the input blocks are not overlapped, and when $L > M$, the input blocks are overlapped. Likewise, when $N = M$, the output blocks are not overlapped, and when $N > M$, the output blocks are overlapped. It should be noted that the down-sampling factor need not be the same as the up-sampling factor. However, in this paper we only consider the case when they are equal, and thus the input and the output sampling rates are equal.

As we shall demonstrate later, the overlapped block filter structure will lead to computationally more efficient realizations in many applications. This fact may not be immediately apparent. When the block size is fixed, the down-sampling factor M determines the data rate of each branch. If M is smaller (as in the overlapped blocks case), the number of computations seems increasing instead of decreasing. But when overlapped

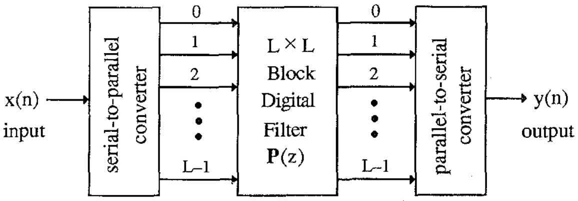

x(n)
input
serial-to-parallel converter
0
1
2
L–1
L × L Block Digital Filter P(z)
0
1
2
L–1
parallel-to-serial converter
y(n)
output

Fig. 1. Schematic representation of a block digital filter.

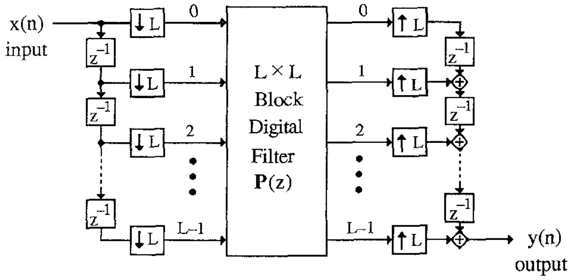


Fig. 2. Multirate representation of a conventional block digital filter.

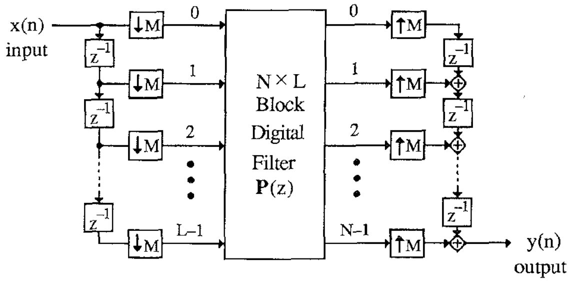


Fig. 3. Multirate representation of an overlapped block digital filter.

Manuscript received December 1, 1994; revised October 2, 1995. This work was supported in part by a University of California MICRO Grant with matching support from Alcatel-Telettra, Inc. This paper was recommended by Associate Editor G. Rajan.

I.-S. Lin was with the Center for Information Processing Research, Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106 USA. He is now with the Chung San Institute of Science and Technology, Taipei, Taiwan.

S. K. Mitra is with the Center for Information Processing Research, Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106 USA.

Publisher Item Identifier S 1057-7130(96)04577-6.

block structure is used, the correlation between signal blocks can be utilized more efficiently resulting in more efficient structures.

II. Shift-Invariant Conditions for Block Digital Filters

2.1 *Input-Output Relation*

Because of the shift-varying property, the output of a overlapped block digital filter contains not only a (linear shift-invariant) filtered version of the input signal, but also aliasing components. A block digital filter can be made shift-invariant by ensuring the elimination of all aliasing components.

From basic multirate theory [6] we can show that the input-output relation of the structure of Fig. 3 in the z-domain is given by

$$Y(z) = \frac{1}{M}[z^{-(N-1)} \quad \cdots \quad z^{-1} \quad 1]\boldsymbol{P}(z^M) \cdot \sum_{k=0}^{M-1} \begin{bmatrix} 1 \\ (zW_M^k)^{-1} \\ \vdots \\ (zW_M^k)^{-(L-1)} \end{bmatrix} X(zW_M^k). \tag{1}$$

The $k = 0$ term is the shift-invariant component and all others are aliasing components. When these aliasing components are canceled, this system becomes shift-invariant.

The input-output relation of a linear shift-varying system can also be described in the sample-domain by a superposition

$$y(n) = \sum_{i=-\infty}^{\infty} h(n,i)x(i) \tag{2}$$

where $x(n)$ and $y(n)$ are, respectively, the input and the output sequences, and $h(n,i)$ is the response of the system at time n to a unit sample sequence applied at time i. A frequency-domain version of (2) is given by

$$Y(e^{j\Omega_1}) = \int_{-\pi}^{\pi} H(e^{j\Omega_1}, e^{j\Omega_2})X(e^{j\Omega_2})\, d\Omega_2 \tag{3}$$

where

$$X(e^{j\Omega_2}) = \sum_{i=-\infty}^{\infty} x(i)e^{-j\Omega_2 i} \tag{4}$$

$$Y(e^{j\Omega_1}) = \sum_{n=-\infty}^{\infty} y(n)e^{-j\Omega_1 n} \tag{5}$$

and the bi-frequency response $H(e^{j\Omega_1}, e^{j\Omega_2})$ is defined as

$$H(e^{j\Omega_1}, e^{j\Omega_2}) = \frac{1}{2\pi}\sum_{i=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} h(n,i)e^{-j\Omega_1 n}e^{j\Omega_2 i}. \tag{6}$$

For an overlapped block digital filter, $h(n + M, i + M) = h(n,i)$ where M is the down-sampling factor. It has been shown [7] that $H(e^{j\Omega_1}, e^{j\Omega_2})$ is nonzero only if $\Omega_2 = \Omega_1 - 2\pi\nu/M, \nu \in 0, 1, \cdots, M-1$, which are parallel lines in the (Ω_1, Ω_2)-plane and corresponding to different aliasing components. Fig. 4 shows a typical bi-frequency response of an overlapped block digital filter with an up/down-sampling factor of 4. The center line is the LSI component and all others are aliasing components.

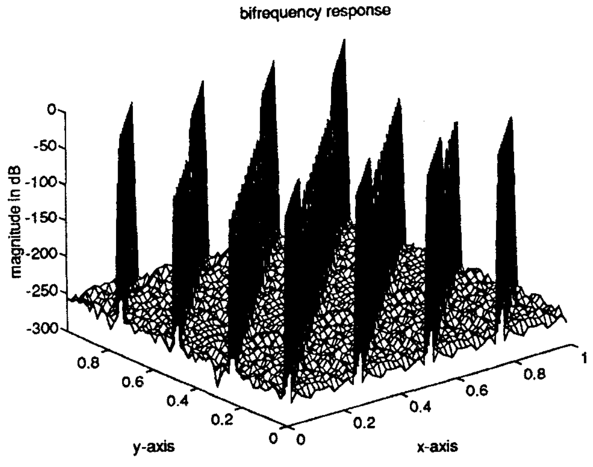


Fig. 4. The bi-frequency response of a typical overlapped block digital filter. The x-axis is $\Omega_1/2\pi$, and the y-axis is $\Omega_2/2\pi$.

2.2 *Shift-Invariant Conditions*

Vaidyanathan and Mitra [8] have studied the shift-invariant conditions for nonoverlapped block digital filters and have proved that for $L = M = N$ (nonoverlapped block processing), the over-all system is shift-invariant if and only if the block transfer matrix $\boldsymbol{P}(z)$ has the following form:

$$\boldsymbol{P}(z) = \begin{bmatrix} H_0(z) & H_1(z) & \cdots & H_{M-1}(z) \\ z^{-1}H_{M-1}(z) & H_0(z) & \cdots & H_{M-2}(z) \\ \vdots & \vdots & \ddots & \vdots \\ z^{-1}H_1(z) & z^{-1}H_2(z) & \cdots & H_0(z) \end{bmatrix}. \tag{7}$$

Note that $\boldsymbol{P}(z)$ is almost like a circulant matrix except that all elements below the main diagonal are multiplied by an additional z^{-1} term. Such a matrix has been called a *pseudocirculant matrix* [8]. The transfer function of the shift-invariant system is then given by

$$H(z) = H_0(z^M) + z^{-1}H_1(z^M) + \cdots + z^{-(M-1)} \cdot H_{M-1}(z^M). \tag{8}$$

Our objective here is to extend the above result to the overlapped ($L \geq M$ and $N \geq M$) case. For simplicity, we consider first a three-branch system ($L = N = 3$) with a down/up-sampling factor of 2 ($M = 2$) as shown in Fig. 5. The block transfer matrix $\boldsymbol{P}(z)$ here is a 3×3 matrix given by:

$$\boldsymbol{P}(z) = \begin{bmatrix} P_{00}(z) & P_{01}(z) & P_{02}(z) \\ P_{10}(z) & P_{11}(z) & P_{12}(z) \\ P_{20}(z) & P_{21}(z) & P_{22}(z) \end{bmatrix}. \tag{9}$$

We can change the input delay-chain structure of above figure to a critically down-sampled form as shown in Fig. 6(a). Likewise, we can change the output interleaving structure of above figure to a nonoverlapped form as shown in Fig. 6(b). After these modifications, an equivalent two-branch structure

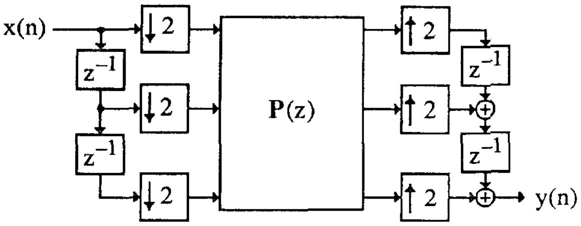


Fig. 5. A simple overlapped block digital filter.

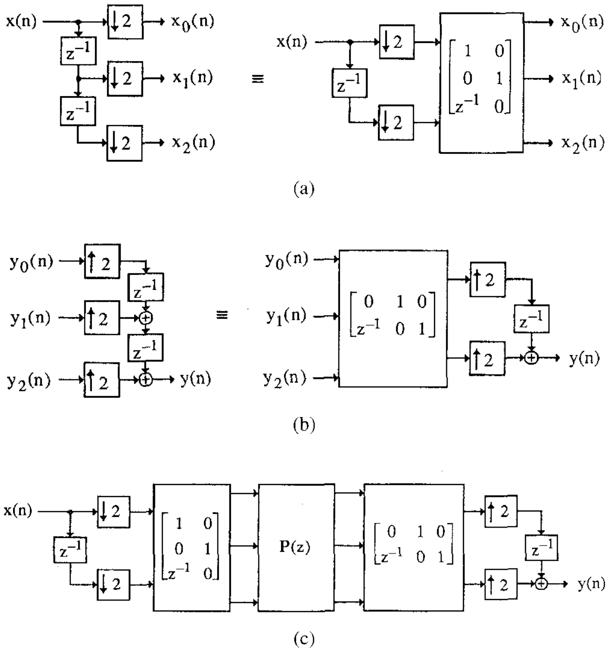


Fig. 6. (a) Equivalent input structure. (b) Equivalent output structure. (c) Equivalent maximally decimated system.

of Fig. 5 is obtained as indicated in Fig. 6(c). From this figure it is evident that if

$$Q(z) = \begin{bmatrix} 0 & 1 & 0 \\ z^{-1} & 0 & 1 \end{bmatrix} P(z) \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ z^{-1} & 0 \end{bmatrix} \tag{10}$$

is a pseudocirculant matrix, the system of Fig. 6(c) and hence that of Fig. 5 becomes shift-in-variant.

For a general overlapped block digital filter with L input branches, N output branches, an up/down-sampling factor of M, and a block transfer function matrix $P(z)$, by using a similar procedure, we can show that the condition for shift-invariance is that the $M \times M$ matrix $Q(z)$ given by

$$Q(z) = R(z)P(z)S(z) \tag{11}$$

be a pseudocirculant matrix where $R(z)$ is a $M \times N$ matrix of the following form:

$$R(z) = \begin{bmatrix} \cdots & 0 & 0 & 0 & 1 & \cdots & 0 & 0 & 0 \\ \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \cdots & z^{-1} & 0 & 0 & 0 & \cdots & 1 & 0 & 0 \\ \cdots & 0 & z^{-1} & 0 & 0 & \cdots & 0 & 1 & 0 \\ \cdots & 0 & 0 & z^{-1} & 0 & \cdots & 0 & 0 & 1 \end{bmatrix} \tag{12}$$

and $S(z)$ is a $L \times M$ matrix of the form:

$$S(z) = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ z^{-1} & 0 & 0 & \cdots & 0 \\ 0 & z^{-1} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \end{bmatrix}. \tag{13}$$

We shall call a block transfer matrix $P(z)$ satisfying the above condition as an *extended pseudocirculant matrix.*

2.3 Implementation of LSI Systems Using Overlapped Block Structure

For a given up/down-sampling factor M, unlike the nonoverlapped case, there are many different choices of the input block length L, the output block length N, and the block transfer matrix $P(z)$, all leading to the same pseudocirculant matrix $Q(z)$ and thus the same transfer function $H(z)$. For instance, to implement in overlapped block structure form a LSI system with a transfer function $H(z) = H_0(z^3) + z^{-1}H_1(z^3) + z^{-2}H_2(z^3)$, we can choose $M = 3, L = 3, N = 5$, and

$$P(z) = \begin{bmatrix} H_2(z) & 0 & 0 \\ H_1(z) & H_2(z) & 0 \\ H_0(z) & H_1(z) & H_2(z) \\ 0 & H_0(z) & H_1(z) \\ 0 & 0 & H_0(z) \end{bmatrix}. \tag{14}$$

As indicated below, the matrix $Q(z)$:

$$Q(z) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ z^{-1} & 0 & 0 & 1 & 0 \\ 0 & z^{-1} & 0 & 0 & 1 \end{bmatrix}$$

$$P(z) = \begin{bmatrix} H_0(z) & H_1(z) & H_2(z) \\ z^{-1}H_2(z) & H_0(z) & H_1(z) \\ z^{-1}H_1(z) & z^{-1}H_2(z) & H_0(z) \end{bmatrix} \tag{15}$$

is a pseudocirculant matrix.

An alternate choice, for example, is $L = 5. N = 3$, and

$$P(z) = \begin{bmatrix} H_0(z) & H_1(z) & H_2(z) & 0 & 0 \\ 0 & H_0(z) & H_1(z) & H_2(z) & 0 \\ 0 & 0 & H_0(z) & H_1(z) & H_2(z) \end{bmatrix}. \tag{16}$$

In this case

$$Q(z) = P(z) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ z^{-1} & 0 & 0 \\ 0 & z^{-1} & 0 \end{bmatrix}$$

$$= \begin{bmatrix} H_0(z) & H_1(z) & H_2(z) \\ z^{-1}H_2(z) & H_0(z) & H_1(z) \\ z^{-1}H_1(z) & z^{-1}H_2(z) & H_0(z) \end{bmatrix} \tag{17}$$

is again seen to be the same pseudocirculant matrix as in (15).

We can also choose $L = 4, N = 4$, and

$$\boldsymbol{P}(z) = \begin{bmatrix} A(z) & H_2(z) & 0 & 0 \\ H_0(z) & H_1(z) & H_2(z) & 0 \\ 0 & H_0(z) & H_1(z) & H_2(z) \\ 0 & 0 & H_0(z) & B(z) \end{bmatrix}. \tag{18}$$

If $A(z) + B(z) = H_1(z)$, we arrive at

$$\begin{aligned} \boldsymbol{Q}(z) &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ z^{-1} & 0 & 0 & 1 \end{bmatrix} \boldsymbol{P}(z) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ z^{-1} & 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} H_0(z) & H_1(z) & H_2(z) \\ z^{-1}H_2(z) & H_0(z) & H_1(z) \\ z^{-1}H_1(z) & z^{-1}H_2(z) & H_0(z) \end{bmatrix} \end{aligned} \tag{19}$$

which is again the same pseudocirculant matrix as in (15).

Now consider the general case. Assume that the LSI transfer function $H(z)$ to be realized is of the form of (8). We can choose an input block size of $L = M$, and an output block size of $N = 2M - 1$, and a block transfer function matrix

$$\boldsymbol{P}_1(z) = \begin{bmatrix} H_{M-1}(z) & 0 & \cdots & 0 \\ H_{M-2}(z) & H_{M-1}(z) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ H_0(z) & H_1(z) & \cdots & H_{M-1}(z) \\ 0 & H_0(z) & \cdots & H_{M-2}(z) \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & H_0(z) \end{bmatrix} \tag{20}$$

Alternately, we can choose a structure with $N = M, L = 2M - 1$, and a block transfer function matrix (see (21) at the bottom of the page).

$$\boldsymbol{P}_2(z) = \begin{bmatrix} H_0(z) & H_1(z) & \cdots & H_{M-1}(z) & 0 & \cdots & 0 \\ 0 & H_0(z) & \cdots & H_{M-2}(z) & H_{M-1}(z) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & H_0(z) & H_1(z) & \cdots & H_{M-1}(z) \end{bmatrix}. \tag{21}$$

It can be easily shown that $\boldsymbol{P}_1(z)$ and $\boldsymbol{P}_2(z)$ are both extended pseudocirculant matrices for the desired transfer function of (8).

We shall call $\boldsymbol{P}_1(z)$ the *Type A extended pseudocirculant matrix*, and $\boldsymbol{P}_2(z)$ the *Type S extended pseudocirculant matrix*. For a Type A structure, the size of the input blocks are equal to the down-sampling factor, and the size of the output blocks are larger than the up-sampling factor. As a result, in this case, the input blocks are not overlapped, whereas, the output blocks are overlapped. These properties are very similar to the conventional overlap-add algorithms for linear convolution. For a Type S structure, it is exactly opposite, i.e., the input blocks are overlapped, while, the output blocks are not overlapped. These properties are very similar to the standard overlap-save algorithms for linear convolution.

III. Fast FIR Filtering Algorithms

FIR filters are often used in digital signal processing applications as they can be designed with exact linear phase and do not have any stability problems. However, an FIR filter structure is computationally more expensive than an IIR equivalent meeting the same filter specifications. Hence, it is important to find computationally efficient FIR filtering algorithms. In this section, we derive a set of such algorithms using the overlapped block structure. Some of these algorithms are similar to those proposed by Vetterli [9], and Mou and Duhamel [10]. However, our approach is more general providing better insight into the problem and leading to a new set of fast filtering algorithms.

3.1 Structures Based on Fast Short-Length Linear Convolution Algorithms

We now derive a set of fast block filtering algorithms that can be considered as direct extensions of fast short-length linear convolution algorithms. The computational complexities of these algorithms are analyzed, and some computer experiments are carried out to verify the analysis. Finite word-length effects are also studied.

3.1.1 Algorithms: It is well known that the linear convolution is equivalent to polynomial multiplication [11] and many fast short-length linear convolution algorithms are derived from this view point [11], [12]. First we establish the link between the polynomial multiplication problem and the Type A extended pseudocirculant matrix. Consider the product $s(x)$ of two first-order polynomials $f(x) = f_1x + f_0$ and $g(x) = g_1x + g_0$:

$$\begin{aligned} s(x) &= f(x)g(x) = (f_1x + f_0)(g_1x + g_0) \\ &= s_2x^2 + s_1x + s_0. \end{aligned} \tag{22}$$

We can rewrite the relation between the coefficients of various polynomials involved as

$$\begin{bmatrix} s_2 \\ s_1 \\ s_0 \end{bmatrix} = \begin{bmatrix} f_1 & 0 \\ f_0 & f_1 \\ 0 & f_0 \end{bmatrix} \begin{bmatrix} g_1 \\ g_0 \end{bmatrix}. \tag{23}$$

Note that the 3 × 2 matrix in the above equation has exactly the same form as the Type A extended pseudocirculant matrix with $M = 2$.

There are a number of fast algorithms which can be used to implement (23) efficiently [12], [13]. For example, using the Winograd algorithm [12] we arrive at

$$\begin{bmatrix} f_1 & 0 \\ f_0 & f_1 \\ 0 & f_0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_1 & 0 & 0 \\ 0 & f_0 + f_1 & 0 \\ 0 & 0 & f_0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} \tag{24}$$

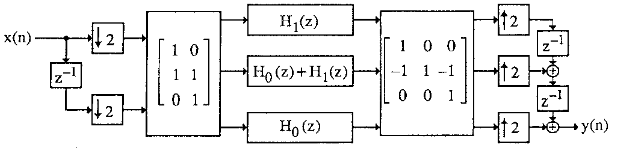


Fig. 7. Fast FIR filtering algorithm based on Type A overlapped block structure.

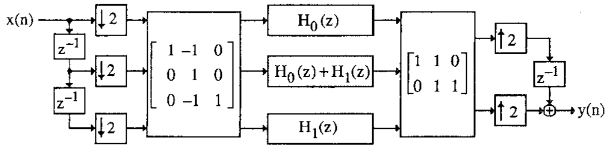


Fig. 8. Fast FIR filtering algorithm based on Type S overlapped block structure.

which requires one less multiplication at the expense two more additions than that are needed in a direct implementation of (23).

Extending (24) to the polynomial case we obtain

$$\begin{bmatrix} H_1(z) & 0 \\ H_0(z) & H_1(z) \\ 0 & H_0(z) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} H_1(z) & 0 & 0 \\ 0 & H_0(z)+H_1(z) & 0 \\ 0 & 0 & H_0(z) \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix}. \qquad (25)$$

The matrix on the left-hand side of (25) is a Type A extended pseudocirculant matrix with $L = M = 2$, and $N = 3$. In implementing a transfer function $H(z) = H_0(z^2)+z^{-1}H_1(z^2)$ we can use the decomposition of (25) and realize $H(z)$ using an overlapped block structure with $L = M = 2, N = 3$, as shown in Fig. 7. Each polynomial $H_i(z), i = 0, 1$, in (25) corresponds to a filtering operation with an FIR filter of about half the length of the original filter $H(z)$ being its polyphase component. Through this process the number of subfiltering operations has been reduced to 3 from 4, thus decreasing the computational complexity.

It can be seen that the Type S extended pseudocirculant matrix is precisely the transpose of the Type A extended pseudocirculant matrix. Hence, a simple transpose operation of a Type A overlapped block structure realization yields an equivalent realization using a Type S overlapped block structure. Fig. 8 indicates the transpose of Fig. 7.

In general, the linear convolution of two length-M sequences $x(n)$ and $h(n)$ can be written in matrix form as

$$\begin{bmatrix} y_{2M-2} \\ y_{2M-3} \\ \vdots \\ y_{M-1} \\ y_{M-2} \\ \vdots \\ y_0 \end{bmatrix} = \begin{bmatrix} h_{M-1} & 0 & \cdots & 0 \\ h_{M-1} & h_{M-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ h_0 & h_1 & \cdots & h_{M-1} \\ 0 & h_0 & \cdots & h_{M-2} \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & h_0 \end{bmatrix} \cdot \begin{bmatrix} x_{M-1} \\ x_{M-2} \\ \vdots \\ x_0 \end{bmatrix}. \qquad (26)$$

Most fast short-length convolution algorithms can be thought of ways to decompose the $(2M-1)\times M$ matrix in the above equation. To be more specific, this matrix can be decomposed into the form $\boldsymbol{UGV}$. Matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ consist of integers only and can be implemented without multiplications, and matrix $\boldsymbol{G}$ is a diagonal matrix whose diagonal elements depend on $h_0, h_1, \cdots, h_{M-1}$ only. We can simply apply the same procedure to the above example to adapt fast short-length linear convolution algorithms into fast FIR filtering algorithms.

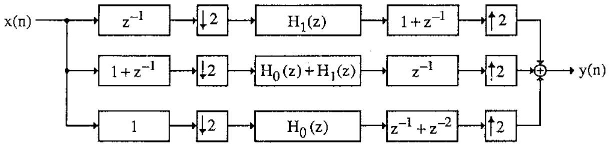


Fig. 9. Efficient FIR filtering based on filter bank structure.

It should be noted that the structure of Fig. 7 is very similar to the structure of Fig. 9 proposed by Vetterli [9]. When the three filters before the down-samplers and the three filters after the up-samplers in Vetterli's structure are implemented in polyphase forms, we arrive at the same algorithm as in Fig. 7. There is not much difference in computational complexity of both his and our algorithms, but the relationship between the multirate implementation and fast linear convolution algorithms are much clearer by using the concept of overlapped block structure. Also, the transpose structures proposed in this paper (using the Type S extended pseudocirculant matrices) are not covered in Vetterli's work.

Mou and Duhamel [10] also proposed similar fast FIR filtering algorithms using the pseudocirculant property. The computational complexity of their structure is exactly the same as our algorithm but the positions of the delay elements are nicely placed in our algorithms and thus our structure is more regular than theirs.

3.2 Computational Complexity

First consider the computational complexity of the structure of Fig. 7. Because the subfilters $H_i(z)$ operate at half the speed of the original filter $H(z)$ and are of half the length of $H(z)$, each subfilter needs a quarter of the number of multplications of the original filter per output sample. The total number of multiplications is therefore only about 3/4 of the direct form implementation of that of $H(z)$. To be precise, assume that $H(z)$ is of length K. Each subfilter is then of length $K/2$ and requires $K/2$ multiplications and $(K-2)/2$ additions. To compute 2 output samples we also need 1 pre-addition, 2 post-additions, and 1 addition for the output interleaving structure. The total number of multiplications per output sample is therefore

$$\frac{1}{2}(3)\frac{K}{2} = \frac{3K}{4}.$$

Likewise, the total number of additions per output sample is

$$\frac{1}{2}\left\{1+2+1+(3)\left(\frac{K}{2}-1\right)\right\}=\frac{3K+2}{4}.$$

Compared to K multiplications and $K-1$ additions of the direct form implementation, this algorithm saves about 25% of total computations.

In the general case, the optimum algorithm needs $2M-1$ multiplications to calculate the linear convolution of two length-M sequences. By using this technique we need $2M-1$ subfilter operations and each filter is $1/M$th the length of the original filter and operates at $1/M$-th speed. We can therefore reduce the number of computations per output sample to about $(2M-1)/M^2$ of that of the direct form implementation at the cost of additional pre-additions and post-additions. It seems that the computational complexity can be reduced by choosing a large M, but the number of additions increases dramatically when M increases. This fact prevents us to choose a large M, and sometimes sub-optimal algorithms are chosen to reduce the number of additions at the cost of more subfilters. Also because all subfilters in the overlapped block structure can be processed in parallel, we can use different processor for each subfilter and thus further reduce the total computation time. The price paid for the reduction of computational complexity is the increased system delay time. The time delay is roughly equal to the block size and larger block means longer delay.

The total number of multiplications and additions only provide a rough estimate of the real computation time because flow control, indexing, and data movement also use computer time. Also different computer architectures may have very different behaviors for the same algorithm. To ensure that these algorithms work in a practical situation we have written a C program to test some of these on a Sparc II workstation and compared their computation time with that of the direct form implementation of the parent transfer function. The simulation results are summarized in Tables I and II. They list the total computer time needed to calculate 50 000 output samples for filters of different lengths. Table I list the result for a Type A overlapped block structure with $M=2$ with all subfilters realized in direct form. Theoretically the ratio between the computation times of the fast algorithm and the direct implementation should approach 75% when the filter length increases and the experimental results verify this fact. Table II lists the computation time of a Type A overlapped block structure with $M=3$. The block transfer function matrix $\boldsymbol{P}(z)$ has been decomposed using the following equation:

$$\begin{bmatrix} H_2 & 0 & 0 \\ H_1 & H_2 & 0 \\ H_0 & H_1 & H_2 \\ 0 & H_0 & H_1 \\ 0 & 0 & H_0 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ -1 & 2 & -2 & -1 & 2 \\ -2 & 1 & 3 & 0 & -1 \\ 1 & -1 & -1 & 1 & -2 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \boldsymbol{D} \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 2 & 4 \\ 0 & 0 & 1 \end{bmatrix} \qquad (27)$$

TABLE I
COMPARISON OF COMPUTER TIMES: $M = 2$ CASE

filter length	30	60	90	120	240	360	480
overlapped block	0.93	1.64	2.42	3.23	6.21	9.29	12.35
direct form	1.03	2.02	3.04	4.08	8.14	12.39	16.69
ratio	0.877	0.812	0.796	0.792	0.763	0.740	0.750

where $\boldsymbol{D}$ is a diagonal matrix given by

$$\boldsymbol{D} = \text{diag}\,[\tfrac{1}{2}\,H_0, \tfrac{1}{2}\,(H_0+H_1+H_2), \tfrac{1}{6}\,(H_0-H_1+H_2), \tfrac{1}{6}\,(H_0+2H_1+4H_2), H_2]. \qquad (28)$$

We need 5 subfilters and 20 additions for each block. Though the two integer matrices can be computed without any multiplications, we can also use some additional multiplications to reduce the number of additions. When the filter length increases, we can see that the ratio of the two computer times approaches the theoretical value of 5/9.

3.3 Finite Wordlength Effects

Because most commercial DSP chips are now optimized for multiply-add-accumulate type operation, they are designed to implement FIR filters in direct form very efficiently. Thus, we only consider here the direct form implementation of all subfilters to study the effect of finite wordlengths.

Since there are no coefficients to be quantized in the pre-filtering and post-filtering parts, coefficient quantization effect is completely determined by the subfilters. After all subfilter coefficients are quantized, we compute the quantized block transfer function matrix and find the equivalent nonoverlapped block transfer function matrix. The shift-invariant part of this system and all aliasing components are computed according to (1). The actual output signal is now given by

$$y(n) = y_q(n) + \alpha(n) = y_{uq}(n) + \varepsilon(n) + \alpha(n) \qquad (29)$$

where $y_q(n)$ is the quantized LSI component which is the sum of the ideal unquantized output $y_{uq}(n)$ and the error $\varepsilon(n)$ in the LSI component due to the coefficient quantization, and $\alpha(n)$ comes from the aliasing components.

Consider a Type A overlapped block structure with an up/down-sampling factor of 2 with (25) used to decompose the block transfer function matrix. After coefficient quantization, we have a quantized block transfer function matrix.

$$\begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & -1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} H_1+E_1 & 0 & 0 \\ 0 & H_0+H_1+E_2 & 0 \\ 0 & 0 & H_0+E_0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} H_1+E_1 & 0 \\ H_0+E_2-E_1 & H_1+E_2-E_0 \\ 0 & H_0+E_0 \end{bmatrix} \qquad (30)$$

where $E_0(z), E_1(z)$, and $E_2(z)$ are the quantization errors of the three subfilters respectively. The equivalent nonoverlapped

TABLE II
COMPARISON OF COMPUTER TIMES: $M = 3$ CASE

filter length	30	60	90	120	240	360	480
overlapped block	0.77	1.32	1.86	2.43	4.68	6.92	9.19
direct form	1.03	2.02	3.04	4.08	8.14	12.39	16.69
ratio	0.748	0.654	0.612	0.596	0.575	0.558	0.551

block transfer function matrix is then

$$\begin{bmatrix} 0 & 1 & 0 \\ z^{-1} & 0 & 1 \end{bmatrix} \begin{bmatrix} H_1 + E_1 & 0 \\ H_0 + E_2 - E_1 & H_1 + E_2 - E_0 \\ 0 & H_0 + E_0 \end{bmatrix} = \begin{bmatrix} H_0 + E_2 - E_1 & H_1 + E_2 - E_0 \\ z^{-1}(H_1 + E_1) & H_0 + E_0 \end{bmatrix}. \tag{31}$$

The shift-invariant component is shown in (32) at the bottom of the page and the aliasing component is shown in (33) also at the bottom of the page.

There are two obvious ways to quantize this system. We can either choose $E_2 = E_0 + E_1$ to eliminate the aliasing components, or we can choose $E_i(z)$ to minimize the difference between the responses of the quantized and the unquantized subfilters. In the first case the LSI component is

$$z^{-1}\{H_0(z^2) + E_0(z^2)\} + z^{-2}\{H_1(z^2) + E_1(z^2)\} \tag{34}$$

which is same as that obtained by a direct quantization of the original filter.

To understand the effect of the second scheme, consider the simplest case for which all $H_i(z) = h_i$ are scalars and quantization step size is Δ. Assume $h_0 = 2.3\Delta, h_1 = 7.4\Delta$, and $h_2 = 2\Delta$. h_0 and h_1, are then quantized, respectively to, 2Δ and 7Δ. Both E_0 and E_1 are in the range $(-\Delta/2, \Delta/2)$. When $-\Delta/2 \leq E_0 + E_1 \geq \Delta/2$, both cases generate the same E_2. On the other hand, if $E_0 + E_1 > \Delta/2$, the first scheme chooses $E_2 = E_0 + E_1 - \Delta$ instead of $E_0 + E_1$. The error in the LSI component is then $E_0 - \Delta/2 + z^{-1}(E_1 - \Delta/2)$. The sum of the square error is therefore

$$\left(E_0 - \frac{\Delta}{2}\right)^2 + \left(E_1 - \frac{\Delta}{2}\right)^2 = \frac{\Delta^2}{2} - \Delta(E_0 + E_1) + E_0^2 + E_1^2. \tag{35}$$

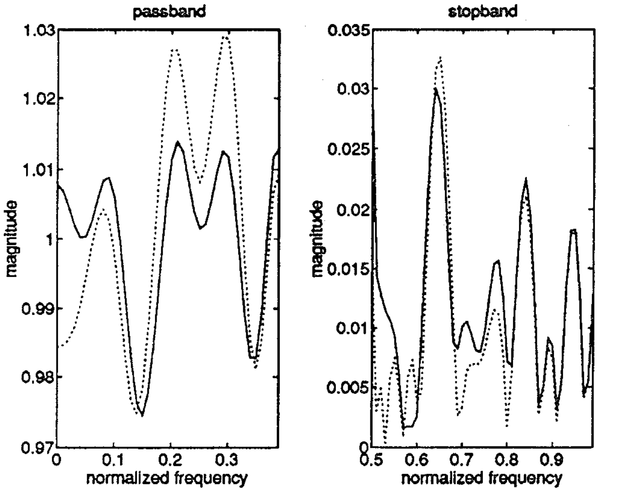


Fig. 10. Frequency responses of the LSI components of the quantized systems: Scheme 1 shown with dotted line and Scheme 2 shown with solid line.

Since $E_0 + E_1 > \Delta/2$, the error is less than $E_0^2 + E_1^2$. This means that this scheme has lower error in the LSI component at the cost of additional aliasing components.

To experimentally determine the performances of the two quantization schemes, we perform a computer simulation using an equiripple length-40 linear phase FIR filter designed with the following specifications: $\omega_p = 0.4\pi, \omega_s = 0.5\pi, \delta p = \delta s = 0.001$. Fig. 10 shows the frequency responses of the LSI components of the two quantized systems. It can be seen that the second quantization scheme has better LSI response which is consistent with our previous analysis. Fig. 11 shows the frequency responses of the aliasing component.

To verify our analysis, we used the technique proposed by Reng and Schssler [13] to measure the LSI and the aliasing components, and the measured results are almost identical to that obtained using the theoretical analysis given above.

IV. DFT BASED FAST FIR FILTERING ALGORITHMS

Use of efficient FFT algorithms to implement FIR filtering has been known for quite some time. The conventional overlap-add and the overlap-save algorithms, when implemented using FFT methods, greatly reduce the computational

$$\frac{1}{2}[z^{-1} \quad 1] \begin{bmatrix} H_0(z^2) + E_2(z^2) - E_1(z^2) & H_1(z^2) + E_2(z^2) - E_0(z^2) \\ z^{-2}(H_1(z^2) + E_1(z^2)) & H_0(z^2) + E_0(z^2) \end{bmatrix} \begin{bmatrix} 1 \\ z^{-1} \end{bmatrix} = \frac{z^{-1}}{2}\{2H_0(z^2) + E_0(z^2) + E_2(z^2) - E_1(z^2) + z^{-1}(2H_1(z^2) + E_1(z^2) + E_2(z^2) - E_0(z^2))\} \tag{32}$$

$$\frac{1}{2}[z^{-1} \quad 1] \begin{bmatrix} H_0(z^2) + E_2(z^2) - E_1(z^2) & H_1(z^2) + E_2(z^2) - E_0(z^2) \\ z^{-2}(H_1(z^2) + E_1(z^2)) & H_0(z^2) + E_0(z^2) \end{bmatrix} \begin{bmatrix} 1 \\ -z^{-1} \end{bmatrix} = \frac{z^{-1}}{2}\{E_2(z^2) - E_0(z^2) - E_1(z^2) + z^{-1}(E_1(z^2) + E_0(z^2) - E_2(z^2))\}. \tag{33}$$

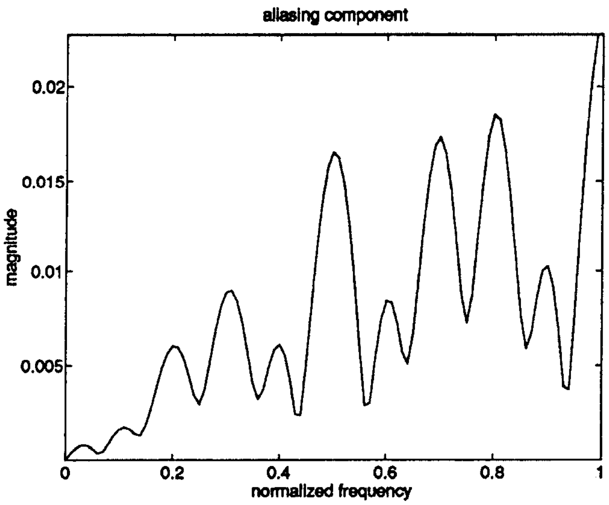


Fig. 11. Frequency responses of the aliasing component.

complexity of FIR filtering. The problem with these algorithms is that the block size must be larger than the filter length, and thus the system delay time increases. Vetterli [9] proposed an algorithm to solve this problem. Based on the overlapped block structure, we derive similar algorithms with improved performance.

4.1 Basic Algorithms

We first establish the relationship between the two special types of extended pseudocirculant matrices and the circulant matrix. For a down-sampling factor of M, the Type A matrix consists of the last M columns of a $(2M-1)\times(2M-1)$ matrix $\boldsymbol{C}_M$, and the Type S matrix consists of the first M rows of $\boldsymbol{C}_M$, where

$$\boldsymbol{C}_M = \begin{bmatrix} H_0 & H_1 & \cdots & H_{M-1} & 0 & \cdots & 0 \\ 0 & H_0 & \cdots & H_{M-2} & H_{M-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & H_0 & H_1 & \cdots & H_{M-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ H_1 & H_2 & \cdots & 0 & 0 & \cdots & H_0 \end{bmatrix}. \tag{36}$$

That is, we can represent these two special types of matrices using the following two equations:

$$\begin{bmatrix} H_{M-1} & 0 & \cdots & 0 \\ H_{M-2} & H_{M-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ H_0 & H_1 & \cdots & H_{M-1} \\ 0 & H_0 & \cdots & H_{M-2} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & H_0 \end{bmatrix} = \boldsymbol{C}_M \begin{bmatrix} \boldsymbol{0}_{M-1,M} \\ \boldsymbol{I}_M \end{bmatrix} \tag{37}$$

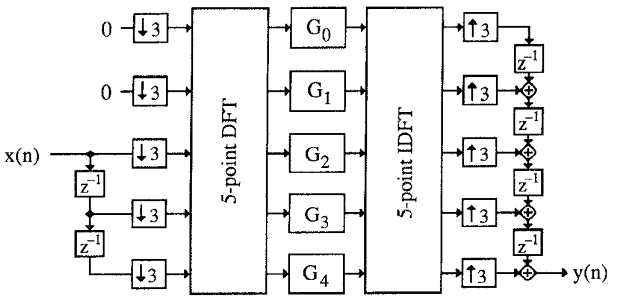


Fig. 12. DFT based algorithm using the Type A overlapped block structure.

$$\begin{bmatrix} H_0 & H_1 & \cdots & H_{M-1} & 0 & \cdots & 0 \\ 0 & H_0 & \cdots & H_{M-2} & H_{M-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & H_0 & H_1 & \cdots & H_{M-1} \end{bmatrix} = [\boldsymbol{I}_M \quad \boldsymbol{0}_{M-1,M}]\boldsymbol{C}_M. \tag{38}$$

where $\boldsymbol{0}_{M-1,M}$ is a null matrix of order $(M-1)\times M$, and $\boldsymbol{I}_M$ is a $M\times M$ identity matrix. The matrix $\boldsymbol{C}_M$ is a circulant matrix and therefore can be diagonalized by the DFT matrix. That is

$$\boldsymbol{C}_M = \boldsymbol{A}\begin{bmatrix} G_0 & 0 & 0 & \cdots & 0 \\ 0 & G_1 & 0 & \cdots & 0 \\ 0 & 0 & G_2 & \ddots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & G_{2M-2} \end{bmatrix}\boldsymbol{B} \tag{39}$$

where $\boldsymbol{A}$ is a $(2M-1)$-point IDFT matrix, $\boldsymbol{B}$ is a $(2M-1)$-point DFT matrix and

$$\begin{bmatrix} G_0 \\ \vdots \\ G_{M-1} \\ G_M \\ \vdots \\ G_{2M-2} \end{bmatrix} = (2M-1)\boldsymbol{A}\begin{bmatrix} H_0 \\ \vdots \\ H_{M-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{40}$$

Fig. 12 shows an overlapped block filter structure with $M=3$ and decomposed using (37) and (39). The transfer function of this system is

$$H(z) = H_0(z^3) + z^{-1}H_0(z^3) + z^{-2}H_0(z^3). \tag{41}$$

This algorithm is very similar to the conventional overlap-add method except in this case the size of the DFT need not be larger than the filter length. We can also implement the same transfer function using (38) and (39) as indicated in Fig. 13. This structure looks exactly the same as the conventional overlap-save method when all subfilters are of length 1.

In general, the DFT based structure only needs $2M-1$ subfilters which are of length $1/M$th of the original filter and operate at $1/M$th speed of the direct form implementation of the parent filter where M is the down-sampling factor. Therefore we can reduce the total number of computations to approximately $(2M-1)/M^2$ of the direct implementation at the cost of additional $(2M-1)$-point IDFT and DFT, and M additions at the output stage. This structure can greatly

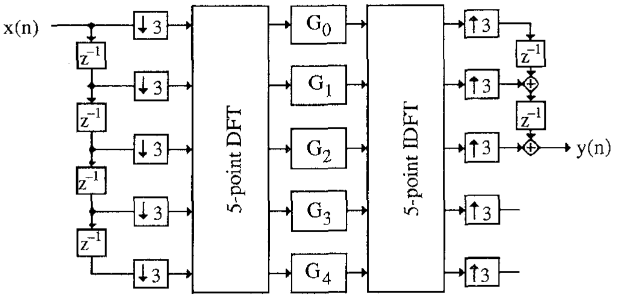


Fig. 13. DFT based algorithm using the Type S overlapped block structure.

reduce the computational load by making use of fast DFT and IDFT schemes.

4.2 FFT Based Algorithms

When the block size is an integer power of 2, we can use the radix-2 FFT algorithm to implement the DFT blocks very efficiently. The problem is that for a down-sampling factor of M, the Type A block structure has a $(2M-1)\times M$ size block transfer function matrix and therefore the size of the blocks is always odd. To use FFT algorithm to perform the decomposition, we need to modify the Type A matrix slightly. The simplest approach would be to add another zero element. To illustrate this approach, consider an FIR filter with a transfer function of the form

$$\begin{aligned} H(z) = {} & H_0(z^4) + z^{-1}H_1(z^4) + z^{-2}H_2(z^4) \\ & + z^{-3}H_3(z^4). \end{aligned} \tag{42}$$

We can implement the above transfer function by using an overlapped block structure with an up/down-sampling factor of 4, input block size of 4, output block size of 8, and the following block transfer function matrix $\boldsymbol{P}(z)$:

$$\boldsymbol{P}(z) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ H_3 & 0 & 0 & 0 \\ H_2 & H_3 & 0 & 0 \\ H_1 & H_2 & H_3 & 0 \\ H_0 & H_1 & H_2 & H_3 \\ 0 & H_0 & H_1 & H_2 \\ 0 & 0 & H_0 & H_1 \\ 0 & 0 & 0 & H_0 \end{bmatrix}. \tag{43}$$

It can be easily verified that $\boldsymbol{P}(z)$ is still an extended pseudocirculant matrix and the overlapped block system is shift-invariant with a transfer function given by (42). It can be further transformed by the following equation:

$$\boldsymbol{P}(z) = \begin{bmatrix} H_0 & H_1 & H_2 & H_3 & 0 & 0 & 0 & 0 \\ 0 & H_0 & H_1 & H_2 & H_3 & 0 & 0 & 0 \\ 0 & 0 & H_0 & H_1 & H_2 & H_3 & 0 & 0 \\ 0 & 0 & 0 & H_0 & H_1 & H_2 & H_3 & 0 \\ 0 & 0 & 0 & 0 & H_0 & H_1 & H_2 & H_3 \\ H_3 & 0 & 0 & 0 & 0 & H_0 & H_1 & H_2 \\ H_2 & H_3 & 0 & 0 & 0 & 0 & H_0 & H_1 \\ H_1 & H_2 & H_3 & 0 & 0 & 0 & 0 & H_0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{44}$$

The first matrix on the right-hand side of the above equation is a circulant matrix and can be diagonalized by the 8-point DFT matrix which can be implemented by using a radix-2 FFT algorithm.

Note that the proposed modification leads to a structure requiring 8 subfilters instead of 7 ($2M$ instead of $2M-1$ in the general case). What we gain here is the use of more efficient DFT and IDFT blocks.

For a length-K FIR filter, the direct implementation needs K multiplications and K_1 additions per output sample. If we use an FFT-based overlapped block structure with a down-sampling factor M, we have $2M$ branches where each branch is an FIR filter of length K/M operating at M times slower rate. We also need a $2M$-point FFT and a $2M$-point IFFT generating M output samples. The total number of multiplications per output sample is therefore

$$2M\left(\frac{1}{M}\right)\left(\frac{K}{M}\right) + \frac{2M\log_2 2M}{M} = \frac{2K}{M} + 2\log_2 2M$$

and the total number of additions is

$$2M\left(\frac{1}{M}\right)\left(\frac{K}{M-1}\right) + \frac{2(2M\log_2 2M)}{M} \cong \frac{2K}{M} + 4\log_2 2M.$$

The number of computations can be reduced by using a larger M but at the cost of a longer delay.

Because the optimum algorithms to compute the linear convolution of two length-M sequences need $(2M-1)$ multiplications, in general, the fast linear convolution based algorithms need $(2M-1)$ subfilters. It seems that the FFT-based algorithms need one more subfilter. But if we consider more carefully, it is possible to improve further the FFT-based algorithms. Consider modifying the above algorithm by changing block transfer function matrix to

$$\hat{\boldsymbol{P}}(z) = \begin{bmatrix} \hat{H}_4 & 0 & 0 & 0 \\ \hat{H}_3 & \hat{H}_4 & 0 & 0 \\ \hat{H}_2 & \hat{H}_3 & \hat{H}_4 & 0 \\ \hat{H}_1 & \hat{H}_2 & \hat{H}_3 & \hat{H}_4 \\ \hat{H}_0 & \hat{H}_1 & \hat{H}_2 & \hat{H}_3 \\ 0 & \hat{H}_0 & \hat{H}_1 & \hat{H}_2 \\ 0 & 0 & \hat{H}_0 & \hat{H}_1 \\ 0 & 0 & 0 & \hat{H}_0 \end{bmatrix}.$$

The equivalent nonoverlapped block transfer matrix $\hat{\boldsymbol{Q}}(z)$ is as follows: (See two matrices at the bottom of the next page.)

It is clear that $\hat{\boldsymbol{Q}}(z)$ is a pseudocirculant matrix corresponding to a shift-invariant system with a transfer function given by

$$\begin{aligned} H(z) = {} & \hat{H}_0(z^4) + z^{-1}\hat{H}_1(z^4) + z^{-2}\hat{H}_2(z^4) + z^{-3}\hat{H}_3(z^4) \\ & + z^{-4}\hat{H}_4(z^4) \end{aligned} \tag{45}$$

and thus $\hat{\boldsymbol{P}}(z)$ is an extended pseudocirculant matrix.

We can also represent $\hat{\boldsymbol{P}}(z)$ by the following equation:

$$\hat{\boldsymbol{P}}(z) = \begin{bmatrix} \hat{H}_0 & \hat{H}_1 & \hat{H}_2 & \hat{H}_3 & \hat{H}_4 & 0 & 0 & 0 \\ 0 & \hat{H}_0 & \hat{H}_1 & \hat{H}_2 & \hat{H}_3 & \hat{H}_4 & 0 & 0 \\ 0 & 0 & \hat{H}_0 & \hat{H}_1 & \hat{H}_2 & \hat{H}_3 & \hat{H}_4 & 0 \\ 0 & 0 & 0 & \hat{H}_0 & \hat{H}_1 & \hat{H}_2 & \hat{H}_3 & \hat{H}_4 \\ \hat{H}_4 & 0 & 0 & 0 & \hat{H}_0 & \hat{H}_1 & \hat{H}_2 & \hat{H}_3 \\ \hat{H}_3 & \hat{H}_4 & 0 & 0 & 0 & \hat{H}_0 & \hat{H}_1 & \hat{H}_2 \\ \hat{H}_2 & \hat{H}_3 & \hat{H}_4 & 0 & 0 & 0 & \hat{H}_0 & \hat{H}_1 \\ \hat{H}_1 & \hat{H}_2 & \hat{H}_3 & \hat{H}_4 & 0 & 0 & 0 & \hat{H}_0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and the 8×8 matrix in the above equation can be diagonalized by a 8-point DFT matrix. That is

$$\begin{bmatrix} \hat{H}_0 & \hat{H}_1 & \hat{H}_2 & \hat{H}_3 & \hat{H}_4 & 0 & 0 & 0 \\ 0 & \hat{H}_0 & \hat{H}_1 & \hat{H}_2 & \hat{H}_3 & \hat{H}_4 & 0 & 0 \\ 0 & 0 & \hat{H}_0 & \hat{H}_1 & \hat{H}_2 & \hat{H}_3 & \hat{H}_4 & 0 \\ 0 & 0 & 0 & \hat{H}_0 & \hat{H}_1 & \hat{H}_2 & \hat{H}_3 & \hat{H}_4 \\ \hat{H}_4 & 0 & 0 & 0 & \hat{H}_0 & \hat{H}_1 & \hat{H}_2 & \hat{H}_3 \\ \hat{H}_3 & \hat{H}_4 & 0 & 0 & 0 & \hat{H}_0 & \hat{H}_1 & \hat{H}_2 \\ \hat{H}_2 & \hat{H}_3 & \hat{H}_4 & 0 & 0 & 0 & \hat{H}_0 & \hat{H}_1 \\ \hat{H}_1 & \hat{H}_2 & \hat{H}_3 & \hat{H}_4 & 0 & 0 & 0 & \hat{H}_0 \end{bmatrix} = \boldsymbol{A}(\mathrm{diag}\,[G_0, G_1, G_2, G_3, G_4, G_5, G_6, G_7])\boldsymbol{B},$$

where $\boldsymbol{A}$ is the 8-point IDFT matrix, $\boldsymbol{B}$ is the 8-point DFT matrix, and

$$\begin{bmatrix} G_0 \\ G_1 \\ G_2 \\ G_3 \\ G_4 \\ G_5 \\ G_6 \\ G_7 \end{bmatrix} = 8\boldsymbol{A} \begin{bmatrix} \hat{H}_0 \\ \hat{H}_1 \\ \hat{H}_2 \\ \hat{H}_3 \\ \hat{H}_4 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{46}$$

If we can find a set of polynomials $\hat{H}_i, i = 0,\ 1,\ 2,\ 3,\ 4$ such that

$$H_0(z) = \hat{H}_0(z) + z^{-1}\hat{H}_4(z) \tag{47}$$

and $H_i(z) = \hat{H}_i(z), i = 1,\ 2,\ 3$, then $\hat{H}(z)$ is equal to $H(z)$.

For instance, if

$$\begin{aligned} H(z) &= 1 + 2z^{-1} + 3z^{-2} + 4z^{-3} + 5z^{-4} \\ &\quad + 6z^{-5} + 7z^{-6} + 8z^{-7} \\ &= (1 + 5z^{-4}) + z^{-1}(2 + 6z^{-1}) + z^{-2}(3 + 7z^{-4}) \\ &\quad + z^{-3}(4 + 8z^{-4}) \\ &= H_0(z^4) + z^{-1}H_1(z^4) + z^{-2}H_2(z^4) + z^{-3}H_3(z^4) \end{aligned}$$

it is easy to show that we can choose $\hat{H}_0(z) = 1 + z^{-1}(5 - a)$ and $\hat{H}_4(z) = a$. Because there are infinite number of choices of $\hat{H}_0(z)$ and $\hat{H}_4(z)$, it is possible to choose certain pairs of $\hat{H}_0(z)$ and $\hat{H}_4(z)$, which can reduce the computational complexity.

One way to do this is to choose $\hat{H}_0(z)$ and $\hat{H}_4(z)$ in such a way that $G_i(z) = 0$ for some i. A procedure to determine $\hat{H}_0(z)$ and $\hat{H}_4(z)$ is as follows: Assume that $h_{i,k}$ is the kth element of the ith subfilter and $H(z) = \Sigma_{i=0}^{R-1} h_i z^{-i}$. First let $\hat{h}_{0,0} = h_{0,0}$. If we want $G_i(z) = 0$, we know from (46) that

$$\hat{h}_{0,0} + W_8^{-i}h_{1,0} + W_8^{-2i}h_{2,0} + W_8^{-3i}h_{3,0} + W_8^{-4i}h_{4,0} = 0$$

where $W_8 = e^{-j2\pi/8}$. From above equation we find $\hat{h}_{4,0}$, and then $\hat{h}_{1,0}$ can be calculated from (47). By repeating this process, we can find $\hat{H}_0(z)$ and $\hat{H}_4(z)$.

A problem associated with this approach is that the filter implemented has a transfer function $H(z) + \hat{h}_{4,K/4-1}z^{-K}$, where

$$\hat{h}_{4,K/4-1} = -\sum_{i=0}^{K-1} W_8^{-i}h_i = -H(W_8^{-i}).$$

As $\hat{h}_{4,K/4-1}$ is not equal to zero in general, we must cancel this term by adding one multiplication and one addition per output sample to keep the transfer function unchanged. When $H(1) = 0$ (as is the case when $H(z)$ is a highpass filter), we can make $G_0(z) = 0$. In this case $\hat{h}_{4,K/4-1} = 0$ and no cancellation is needed. When $H(-1) = 0$ (as in the case when $H(z)$ is a lowpass filter), we can choose $\hat{H}_0(z)$ and $\hat{H}_4(z)$ in such a way that $G_4(z) = 0$ and no additional computation is added. Therefore we can reduce the number of subfilters by one with little cost in general, and none in some cases.

When calculating the total number of multiplications and addition, we assume that complex arithmetic operations are used. If input signals and filter coefficients are all real, we need to compute half of the subfiltering operations because of the symmetrical property. Therefore we can reduce further the computational complexity.

$$\hat{\boldsymbol{Q}}(z) = \begin{bmatrix} z^{-1} & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & z^{-1} & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & z^{-1} & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & z^{-1} & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\hat{\boldsymbol{P}}(z) = \begin{bmatrix} \hat{H}_0 + z^{-1}\hat{H}_4 & \hat{H}_1 & \hat{H}_2 & \hat{H}_3 \\ z^{-1}\hat{H}_3 & \hat{H}_0 + z^{-1}\hat{H}_4 & \hat{H}_1 & \hat{H}_2 \\ z^{-1}\hat{H}_2 & z^{-1}\hat{H}_3 & \hat{H}_0 + z^{-1}\hat{H}_4 & \hat{H}_1 \\ z^{-1}\hat{H}_1 & z^{-1}\hat{H}_2 & z^{-1}\hat{H}_3 & \hat{H}_0 + z^{-1}\hat{H}_4 \end{bmatrix}.$$

V. Concluding Remarks

The conventional block digital filtering approach has been generalized to the overlapped block case. We have derived the condition for shift-invariance operation of a linear overlapped block digital filter and the implementation of a LSI transfer function using an overlapped block structure. A set of fast FIR algorithms based on the overlapped block structure have been derived. These algorithms are highly parallel and thus can be implemented using multiple processors to reduce the total computation time. Finite wordlength effects of some of these fast filtering algorithms have been discussed. Implementation of IIR transfer functions using the overlapped block structure is feasible but their higher computational complexity compared to their nonblock implementation make them less attractive for practical applications [14].

References

[1] B. Gold and K. Jordan, "A note on digital filter synthesis," *Proc. IEEE (Letters)*, vol. 65, pp. 1717–1718, Oct. 1968.

[2] C. S. Burrus, "Block implementation of digital filters," *IEEE Trans. Circuit Theory*, vol. CT-18, pp. 697–701, Nov. 1971.

[3] ——, "Block realization of digital filters," *IEEE Trans. Audio Electroacoust.*, vol. AU-20, pp. 230–235, Oct. 1972.

[4] S. K. Mitra and R. Gnanasekaran, "Block implementation of recursive digital filters—New structures and properties," *IEEE Trans. Circuits Syst.*, vol. CAS-25, pp. 200–207, Apr. 1978.

[5] C. Barnes and S. Shinnaka, "Block-shift-invariance and block implementation of discrete-time filters," *IEEE Trans. Circuits Syst.*, vol. CAS-27, pp. 667–672, Aug. 1980.

[6] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[7] C. Loeffler and C. S. Burrus, "Optimal design of periodically time-varying and multirate digital filters," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-32, pp. 991–997, Oct. 1984.

[8] P. P. Vaidyanathan and S. K. Mitra, "Polyphase networks, block digital filtering, LPTV systems, and alias-free QMF banks: A unified approach based on pseudo-circulants," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 36, pp. 381–391, Mar. 1988.

[9] M. Vetterli, "Running FIR and IIR filtering using multirate filter banks," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 36, pp. 730–738, May 1988.

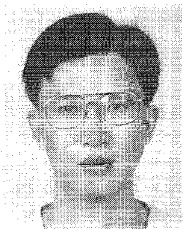
[10] Z. Mou and P. Duhamel, "Short-length FIR filters and their use in fast nonrecursive filtering," *IEEE Trans. Signal Processing*, vol. 39, pp. 1322–1332, June 1991.

[11] R. Blahut, *Fast Algorithms for Digital Signal Processing*. Reading, MA: Addison-Wesley, 1984.

[12] S. Winograd, *Arithmetic Complexities of Computations*, CBMS-NSF Regional Conf. Series in Applied Mathematics, SIAM Pub. 33, 1980.

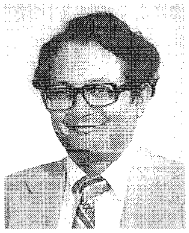
[13] R. Reng and H. W. Schssler, "Measurement of aliasing distortion and quantization noise in multirate systems," in *Proc. IEEE Int. Symp. Circuits Syst.*, San Diego, CA, 1988, pp. 1281–1284.

[14] I.-S. Lin, "Overlapped block digital filtering," Ph.D. dissertation, Dept. Electrical Computer Eng., Univ. California, Santa Barbara, Nov. 1993.



Ing-Song Lin received the B.S.E.E. and M.S.E.E. degrees from National Taiwan University in 1983 and 1985 respectively, and the Ph.D. degree in electrical and computer engineering from the University of California, Santa Barbara in 1994.

Since 1994, he has been with Chung-Shan Institute of Science and Technology, Taiwan, where he is presently an associate scientist. His research interests include digital signal and image processing.



Sanjit K. Mitra (SM'69-F'74) received the B.Sc. (Hons.) degree in physics in 1953 from Utkal University, Cuttack, India; the M.Sc. (Tech.) degree in radio physics and electronics in 1956 from Calcutta University; the M.S. and Ph.D. degrees in electrical engineering from the University of California, Berkeley, in 1960 and 1962, respectively.

From June 1962 to June 1965, he was with Cornell University, Ithaca, NY, as an Assistant Professor of Electrical Engineering. He was with AT&T Bell Laboratories, Holmdel, NJ, from June 1965 to January 1967. Since then, he has been on the faculty of the University of California, first at the Davis campus and more recently at the Santa Barbara campus as a Professor of Electrical and Computer Engineering, where he served as Chairman of the Department from July 1979 to June 1982.

Dr. Mitra served as the President of the IEEE Circuits and Systems Society in 1986. He is currently a member of the editorial boards of the *International Journal on Circuits, Systems and Signal Processing*, the *International Journal on Multidimensional Systems and Signal Processing*; *Signal Processing*; and the *Journal of the Franklin Institute*. He is the recipient of the 1973 F. E. Terman Award and the 1985 AT&T Foundation Award of the American Society of Engineering Education, the Education Award of the IEEE Circuits and Systems Society in 1989, the Distinguished Senior U.S. Scientist Award from the Alexander von Humboldt Foundation of West Germany in 1989 and the Technical Achievement Award of the IEEE Signal Processing Society in 1996. In May 1987 he was awarded an Honorary Doctorate of Technology degree from the Tampere University of Technology, Tampere, Finland. He is a Fellow of the AAAS and SPIE, and a member of EURASIP and ASEE.