guessed form in [5]. It means that by Lemma 2, the guessed forms in [5] and [6] can be simplified into our final form [see (7)].

The significance of the Householder transform is due to its popular use in many fields of matrix computations and signal processing. The main contributions of this paper are threefold: First, we give the direct derivation of the complex Householder transform; second, we apply this derivation to derive the hyperbolic Householder transform directly; third, the previous guessed complex Householder transforms in [5] and [6] can be simplified by using our partial result, i.e., Lemma 2. In fact, according to the results of this correspondence, a block representation for products of hyperbolic Householder transform [7], which is very suitable for vector supercomputing, has also been derived.

ACKNOWLEDGMENT

The authors thank the two anonymous referees and Prof. V. E. DeBrunner for their comments that helped to improve the presentation of this paper.

REFERENCES

[1] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed. Baltimore, MD: Johns Hopkins Univ. Press, 1989.
[2] A. S. Householder, *The Theory of Matrices in Numerical Analysis*. New York: Dover, 1968.
[3] J. J. Dongarra, J. R. Bunch, C. B. Moler, and G. W. Stewart, *Linpack Users Guide*. Philadelphia, PA: SIAM, 1979, ch. 9, pp. 9.15–9.19.
[4] C. M. Rader and A. O. Steinhardt, "Hyperbolic householder transformations," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 1589–1602, 1986.
[5] V. C. Venkaiah, V. Krishna, and A. Paulraj, "Householder transform in $C^m$," *Digital Signal Process.*, vol. 3, pp. 226–227, 1993.
[6] X. G. Xia and B. W. Suter, "On the householder transform in $C^m$," *Digital Signal Process.*, vol. 5, pp. 116–117, 1995.
[7] W. M. Yan and K. L. Chung, "A block representation for products of hyperbolic householder transform," *Appl. Math. Lett.*, vol. 10, no. 1, pp. 109–112, 1997.

# Extended Overlap-Add and -Save Methods for Multirate Signal Processing

Shogo Muramatsu and Hitoshi Kiya

***Abstract*— The overlap-add method (OLA) and overlap-save method (OLS) are well known as efficient schemes for high-order FIR filtering. In this correspondence, new sampling rate conversion methods are proposed by extending the OLA and OLS and eliminating the redundancy caused by the conversion. First, for finite-duration sequences, a rate conversion with the DFT-domain approach is discussed. Then, using the result, the extended OLA and OLS are proposed for infinite-duration sequences. Last, the computational complexities of our proposed methods are shown.**

***Index Terms*—Fast Fourier transform (FFT), overlap-add/save method, sampling rate conversion.**

## I. INTRODUCTION

Multirate signal processing is indispensable as a fundamental scheme for reduction of computation, improvement of processing precision, compression of information, etc. [1]. In this work, we consider efficiently converting the sampling rate of a digital signal. The conversion can be divided into two classes. One is a rate reduction, and the other is a rate increase. For the discussion without loss of generality, we deal with a sampling rate conversion by an arbitrary rational factor $U/D$, where $D$ and $U$ are integers.

A matter of serious concern on rate conversion is filtering since it accounts for most of the operations, and the filter characteristics determine the performance of the system. Hence, the redundant operation caused by the conversion should be avoided. Furthermore, it is of interest to efficiently perform the high-order filtering in terms of the characteristics. For these reasons, various techniques avoiding the redundancy and decreasing the complexity have been investigated. The polyphase decomposition [1] and the FFT-based interpolation method [2]–[4] represent those techniques. The complexities of the former, however, increase linearly with the tap length of the filter. On the other hand, the latter covers only a finite-duration data sequence, although the complexities are insensitive to the tap length because of the DFT-domain approach.

Therefore, in order to provide an efficient rate conversion technique, which can handle infinite-duration sequences with the DFT-domain approach, we propose to extend the overlap-add method (OLA) and overlap-save method (OLS) for FIR filtering to the sampling rate conversion.

## II. SAMPLING RATE CONVERSION

In this section, we review sampling rate conversion and consider the DFT-domain approach, supposing that the input data sequence is of finite length.

### A. Rate Conversion with the Time-Domain Approach

Fig. 1 indicates the basic structure of the sampling rate conversion system [1].

Manuscript received November 28 1994; revised September 18, 1996. The associate editor coordinating the review of this paper and approving it for publication was Prof. Roberto H. Bamberger.

The authors are with the Department of Electrical Engineering, Graduate School of Engineering, Tokyo Metropolitan University, Tokyo, Japan.

Publisher Item Identifier S 1053-587X(97)-6458-1.

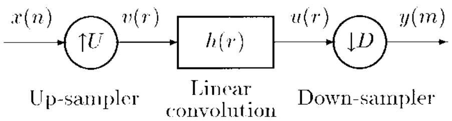


Fig. 1. Basic structure of sampling rate conversion by a rational factor $U/D$.

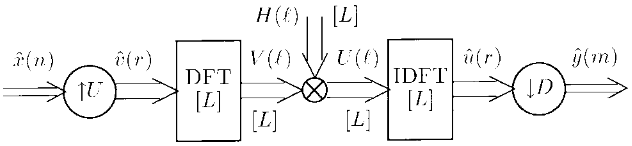


Fig. 2. Structure of sampling rate conversion with the circular convolution approach for a finite-length sequence.

In Fig. 1

$x(n)$ input data sequence;
$y(m)$ final output data sequence;
$h(r)$ impulse response of a filter.

These sequences are related as follows:

$$v(r) = \begin{cases} x\left(\dfrac{r}{U}\right), & r = nU \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

$$u(r) = v(r) * h(r) \tag{2}$$

$$y(m) = u(mD) \tag{3}$$

where the operator "$*$" stands for linear convolution.

### *B. Filtering with the DFT-Domain Approach*

The product of two DFT's is equivalent to the circular convolution of the corresponding time-domain sequences, and the use of the FFT makes it efficient [5]. For the following discussion, we consider substituting the DFT-domain approach for the filtering in the rate conversion, supposing that the input data sequence is of finite length $N_X$ and that an FIR filter of length $L_H$ is given. Fig. 2 shows the structure. Note that this approach does not produce the same result as that of the linear-convolution one.

In Fig. 2, $\hat{x}(n)$ denotes the finite-duration input sequence, and $\hat{y}(m)$ denotes the output sequence (the hat notation "$\hat{}$" is used to distinguish finite-duration sequences from infinite-duration ones). The DFT size $L$ corresponds to the period of the circular convolution, where it is assumed that $L \geq UN_X$ and $L \geq L_H$. Additionally, $U(\ell)$, $V(\ell)$, and $H(\ell)$ represent the $L$-point DFT's of corresponding intermediate sequences $\hat{u}(r)$, $\hat{v}(r)$, and the impulse response $h(r)$ of the filter, respectively. They are related to each other by

$$U(\ell) = H(\ell)V(\ell), \qquad 0 \leq \ell \leq L-1. \tag{4}$$

### *C. Rate Conversion with the DFT-Domain Approach*

Although the structure shown in Fig. 2 contains the redundancy caused by upsampling and downsampling, if the DFT size $L$ is chosen to be a common multiple of $U$ and $D$ [which is denoted as cm$(U, D)$], then the redundancy can be eliminated. Let $N = L/U$ and $M = L/D$, where both $N$ and $M$ are integers because $L$ is a cm$(U, D)$. Then, the reduced structure can be constructed as shown in Fig. 3.

The procedure involved is as follows.

Step 1) Obtain the $N$-point DFT of $\hat{x}(n)$ as

$$X(k) = \sum_{n=0}^{N-1} \hat{x}(n) W_N^{kn}, \qquad 0 \leq k \leq N-1 \tag{5}$$

where $W_N = e^{-j(2\pi/N)}$.

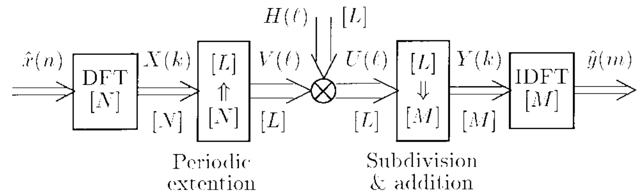


Fig. 3. Structure of sampling rate conversion with the DFT-domain approach for a finite-length sequence.

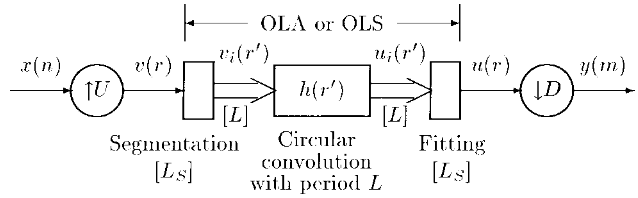


Fig. 4. Structure of sampling rate conversion with OLA or OLS.

Step 2) Periodically extend $X(k)$ as

$$V(\ell) = X[((\ell))_N], \qquad 0 \leq \ell \leq L-1 \tag{6}$$

where the function $((\cdot))_N$ denotes the integer of its argument modulo $N$.

Step 3) Calculate the product of $H(\ell)$ and $V(\ell)$ as in (4).

Step 4) Subdivide $U(\ell)$ and add the subsequences together as

$$Y(k) = \frac{1}{D} \sum_{d=0}^{D-1} U(dM + k), \qquad 0 \leq k \leq M-1. \tag{7}$$

Step 5) Obtain the $M$-point IDFT of $Y(k)$ as

$$\hat{y}(m) = \frac{1}{M} \sum_{k=0}^{M-1} Y(k) W_M^{-km}, \qquad 0 \leq m \leq M-1. \tag{8}$$

Fig. 3 indicates that the DFT and the IDFT size are reduced to one $U$th and one $D$th of those in the structure shown in Fig. 2, respectively, and that Steps 2 and 4 are set in the DFT domain instead of upsampling and downsampling in the time domain. Here, note that Steps 2 and 4 can be implemented without any multiplication, except for scaling by the factor $1/D$, which can be included in the filter $H(\ell)$.

The above procedure can be easily proven from the definitions of the DFT and the IDFT [(1) and (3)] and the division theorem for integers. In Section IV, we will apply this technique to infinite-duration sequences.

## III. Block Filtering

In this section, we discuss the infinite duration input case and consider applying the conventional OLA and OLS to the filtering in Fig. 1 as shown in Fig. 4 [5].

Suppose that $x(n)$ is an input sequence of infinite duration, and $h(r)$ is the impulse response of an FIR filter of length $L_H$ satisfying causality.

### *A. The Overlap-Add Method (OLA)*

The filtering of the upsampled sequence $v(r)$ with $h(r)$ using OLA can be implemented as the following procedure [5].

Step 1) Segment the upsampled sequence $v(r)$ into blocks of length $L_S$ and append $L - L_S$ zeros after each segment to make them $L$ point input data blocks as

$$v_i(r') = \begin{cases} v(iL_S + r'), & 0 \le r' \le L_S - 1 \\ 0, & L_S \le r' \le L - 1, \end{cases} \quad i = 0, 1, 2, 3, \cdots \tag{9}$$

where $v_i(r')$ is the $i$th input data block. Here, the block size $L$ has to be chosen to satisfy

$$L \ge L_S + L_H - 1. \tag{10}$$

Step 2) Execute the circular convolution of the input data block $v_i(r')$ and the impulse response $h(r')$ with period $L$. As we mentioned before, this can be done by multiplying the corresponding DFT's as in (4).

Step 3) Overlap and add the output data block $u_i(r)$ together to obtain the overall filtered sequence $u(r)$ as

$$u(r) = \sum_{i=0}^{K} u'_{\lfloor r/L_S \rfloor - i}[iL_S + ((r))_{L_S}], \quad r = 0, 1, 2, 3, \cdots \tag{11}$$

$$u'_i(r') = \begin{cases} u_i(r'), & 0 \le r' \le L_S + L_H - 2 \\ 0, & L_S + L_H - 1 \le r' \le (K+1)L_S - 1 \end{cases} \tag{12}$$

where $K = \lceil (L_H - 1)/L_S \rceil$, which denotes the number of overlapping blocks. In addition, the function $\lfloor \cdot \rfloor$ denotes the integer value of its argument, and the function $\lceil \cdot \rceil$ denotes the smallest integer greater than or equal to its argument.

The above Steps 1 and 3 correspond to the "segmentation" and "fitting" in Fig. 4, respectively.

### B. The Overlap-Save Method (OLS)

OLS is a good alternative technique of OLA. Hence, we also consider applying OLS to the system shown in Fig. 1 as shown in Fig. 4. The procedure involved is as follows [5]:

Step 1) Segment $v(r)$ into blocks of length $L_S$, and append the last $L - L_S$ points of the previous data sequences after each segment as

$$v_i(r') = \begin{cases} v(iL_S + r'), & 0 \le r' \le L_S - 1 \\ v(iL_S - L + r'), & L_S \le r' \le L - 1. \end{cases} \tag{13}$$

Step 2) Execute the circular convolution of $v_i(r')$ and $h(r')$ with period $L$.

Step 3) Discard the last $L - L_S$ points of the output data block $u_i(r')$, and fit the block of the remaining $L_S$ points together to obtain the overall filtered sequence $u(r)$ as in (11) with $K = 0$, that is, no overlapping, where $u'_i(r') = u_i(r'), 0 \le r' \le L_S - 1$.

The above Steps 1 and 3 correspond to the "segmentation" and the "fitting" operation in Fig. 4, respectively. Note that although the above OLS deals with the circularly rotated version of the data block, the final result can be completed because of the periodic property of the circular convolution.

## IV. Proposed Methods

The implementation of the structure shown in Fig. 4 is more efficient than that in Fig. 1 if a high-order FIR filter is employed and the FFT algorithm is used for the circular convolution. However, there still exists the redundancy caused by upsampling and downsampling. Hence, in this section, we propose to eliminate it by extending the conventional OLA and OLS for the sampling rate conversion and using the system shown in Fig. 3.

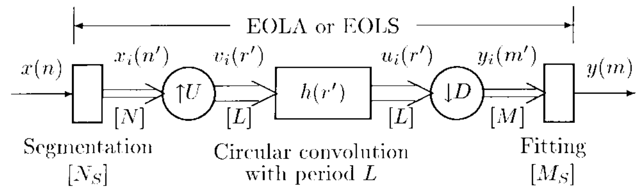


Fig. 5. Structure of sampling rate conversion with EOLA or EOLS.

First, let us consider shifting both of the "segmentation" and "fitting" operators in Fig. 4, as shown in Fig. 5. The shift can be achieved when the segment length $L_S$ in Fig. 4 is chosen to be a cm$(U, D)$. Then, the processing between these operations can be replaced with the system, as shown in Fig. 2. Furthermore, if the period $L$ is chosen to be a cm$(U, D)$, the processing can be implemented by the structure as shown in Fig. 3. As a result, the redundancy can be eliminated.

In the following, we show the procedure of our proposed methods. Suppose that

$$L, L_S \in \{\text{cm}(U, D)\} \tag{14}$$

and let $N = L/U$, $M = L/D$, $N_S = L_S/U$, and $M_S = L_S/D$.

### A. The Extended Overlap-Add Method (EOLA)

When employing OLA, the condition in (10) has to be satisfied. However, if the segment length $L_S$ in Fig. 4 is a multiple of the factor $U$, it can be relaxed as

$$L \ge \{L_S - (U-1)\} + L_H - 1 = L_S + L_H - U \tag{15}$$

because the last $U - 1$ points in each segment of $v(r)$ are certain to be zero due to the upsampling. Therefore, (15) can be regarded as the condition of EOLA. The following shows the procedure of EOLA.

Step 1) Segment the input sequence $x(n)$ into blocks of length $N_S$ and append $N - N_S$ zeros after each segment to make them $N$-point input data blocks as

$$x_i(n') = \begin{cases} x(iN_S + n'), & 0 \le n' \le N_S - 1 \\ 0, & N_S \le n' \le N - 1. \end{cases} \quad i = 0, 1, 2, 3, \cdots. \tag{16}$$

Step 2) Execute the sampling rate conversion in the DFT domain as shown in Fig. 3 by regarding the input data block $x_i(n')$ and the output data block $y_i(m')$ as $\hat{x}(n)$ and $\hat{y}(m)$, respectively.

Step 3) Overlap and add the output data block $y_i(m')$ together to obtain the overall output sequence $y(m)$ as

$$y(m) = \sum_{i=0}^{K} y'_{\lfloor m/M_S \rfloor - i}[iM_S + ((m))_{M_S}] \quad m = 0, 1, 2, 3, \cdots \tag{17}$$

$$y'_i(m') = \begin{cases} y_i(m'), & 0 \le m' \le M_S + \left\lceil \dfrac{L_H - U}{D} \right\rceil - 1 \\ 0, & M_S + \left\lceil \dfrac{L_H - U}{D} \right\rceil \le m' \le (K+1)M_S - 1 \end{cases} \tag{18}$$

where $K = \lceil (L_H - U)/L_S \rceil$, which denotes the number of overlapping blocks.

### B. The Extended Overlap-Save Method (EOLS)

When employing OLS, the condition in (10) has to be satisfied. However, if the segment length $L_S$ in Fig. 4 is a multiple of the factor $D$, it can be relaxed as

$$L \geq \{L_S - (D-1)\} + L_H - 1 = L_S + L_H - D \tag{19}$$

because the last $D-1$ points in each segment of $v(r)$ are of no use to obtain $y_i(m)$ due to the downsampling. As a result, we can discard the corresponding points of $x_i(n)$ in Fig. 5. Therefore, (19) can be regarded as the condition of EOLS.

The following shows the procedure of EOLS.

Step 1) Segment $x(n)$ into blocks of length $N_S$, and append the last $N - \{N_S - \lfloor (D-1)/U \rfloor\}$ points of the previous data sequences after each segment as

$$x_i(n') = \begin{cases} x(iN_S + n'), & 0 \leq n' \leq N_S - \left\lfloor \dfrac{D-1}{U} \right\rfloor - 1 \\ x(iN_S - N + n'), & N_S - \left\lfloor \dfrac{D-1}{U} \right\rfloor \leq n' \leq N-1 \end{cases} \tag{20}$$

Step 2) Execute the sampling rate conversion in the DFT domain of $x_i(n')$ as in Fig. 3.

Step 3) Discard the last $M - M_S$ points of the output data block $y_i(m')$, and fit the block of the remaining $M_S$ points together to obtain the overall output sequence $y(m)$ as in (17) with $K = 0$, that is, no overlapping, where $y_i'(m') = y_i(m')$, $0 \leq m' \leq M_S - 1$.

## V. Computational Complexities

In this section, we discuss the computational complexities of EOLA and EOLS, supposing that both the input data sequence $x(n)$ and the impulse response $h(r)$ are real and that the DFT of $h(r)$ is computed once and stored and absorbs the scale factor $1/D$ in (7). In addition, we assume that one complex multiplication can be performed by three real additions and three real multiplications (the 3/3 algorithm).

The number of real multiplications per output ♯MUL and the number of real additions per output ♯ADD of our proposed methods can be expressed as

$$\sharp\text{MUL} = \frac{\mu(N) + \mu(M) + 3\left(\left\lfloor \dfrac{L}{2} \right\rfloor + 1\right)}{M_S} \tag{21}$$

$$\sharp\text{ADD} = \frac{\alpha(N) + \alpha(M) + 3\left(\left\lfloor \dfrac{L}{2} \right\rfloor + 1\right) + (D-1)M + \beta}{M_S} \tag{22}$$

where $\beta = \lceil (L_H - U)/D \rceil$ for EOLA, and $\beta = 0$ for EOLS. In addition, $\mu(N)$ and $\alpha(N)$ denote the total number of real multiplications and additions of an $N$-point DFT or IDFT, respectively. These numerators and denominators are the total number of the operations per block and the practical number of output points per block, respectively. If the DFT size $N$ and the IDFT size $M$ are products of primes or powers of two, then the FFT algorithms, such as prime-factor algorithm [6] and split-radix FFT [7], can be used.

In order to show the availability of EOLA and EOLS, we compare the number of real multiplications per output of EOLA for the tap length $L_H$ with that of the polyphase structure [1] under the condition that a) $U/D = 2$ and b) $U/D = \frac{1}{2}$. Fig. 6 shows this, where we assume that the period $L$ is suitably chosen to be the smallest power of two greater than $4L_H$, that the number in the output segment points $M_S$ is chosen to be as large as possible, and that the real-valued split-radix FFT [$\mu(N) = N/2 \log_2 N - 3N/2 + 2$] [8] is employed. On the other hand, the number of real multiplications per output of the polyphase structure is estimated by $L_H/U$. Since the complexities of EOLS are similar to those of EOLA, we omit showing the complexities.

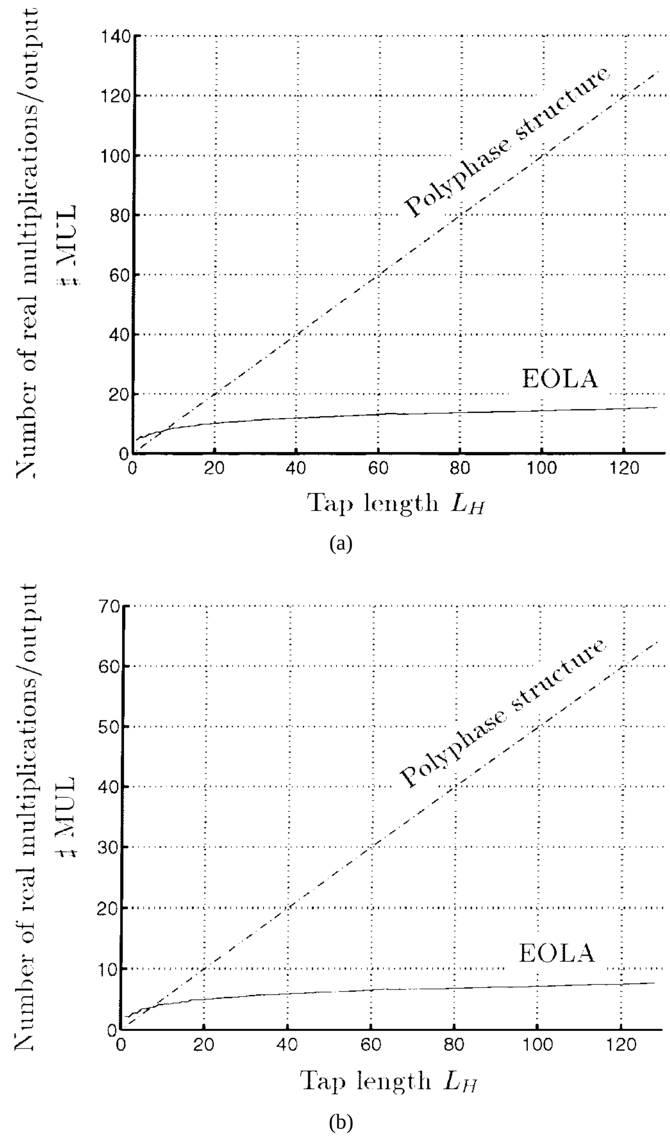


Fig. 6. Comparison of the number of real multiplications per output. (a) Decimation by an integer factor 2 ($U = 1$, $D = 2$) and (b) interpolation by an integer factor 2 ($U = 2$, $D = 1$).

From Fig. 6, we see that the EOLA (EOLS) is more efficient than the polyphase structure under both conditions (a) and (b) when employing a high-order FIR filter. We note that this statement can be generalized for the other conditions.

## VI. Conclusions

In this work, we proposed two methods for the sampling rate conversion—the extended overlap-add method and the extended overlap-save method—and verified their efficiency by the comparison with the polyphase structure. Since these proposed methods are as effective as the conventional OLA and OLS, the implementation leads to high efficiency when employing a high-order FIR filter and utilizing the FFT.

## References

[1] R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1983.
[2] S. Holm, "FFT pruning applied to time domain interpolation and peak localization," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-35, pp. 1176–1777, Dec. 1987.
[3] J. W. Adams, "A subsequence approach to interpolation using the FFT," *IEEE Trans. Circuits Syst.*, vol. 37, pp. 623–625, May 1990.
[4] D. Fraster, "Interpolation by the FFT revisited—An experimental investigation," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. 37, pp. 665–675, May 1989.
[5] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing.* New York: Macmillian, 1988.
[6] D. P. Kolba and D. W. Parks, "A prime factor FFT algorithm using high-speed convolution," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-25, pp. 281–294, Aug. 1977.
[7] H. V. Sorensen, M. T. Heideman, and C. S. Burrus, "On computing the split-radix FFT," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-34, pp. 152–156, Feb. 1986.
[8] H. V. Sorensen, D. L. Jones, M. T. Heideman, and C. S. Burrus, "Real-valued fast Fourier transform algorithms," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-35, pp. 849–863, 1987.

# Design of Linear Phase $M$-Channel Perfect Reconstruction FIR Filter Banks

Takayuki Nagai, Takaaki Fuchie, and Masaaki Ikehara

***Abstract*—We propose two approaches to design $M$ channel nonparaunitary filter banks that satisfy perfect reconstruction (PR) and linear phase (LP) properties. In the first approach, the PR condition is imposed on only a highpass filter. Although this method does not require nonlinear optimization, it has a demerit in that the order of a highpass filter becomes high. In the second approach, two filters are optimized simultaneously using a Lagrange–Newton method. We can design PR filter banks that have the same length. The PR constraint is also formulated as a linear and nonlinear equation of the analysis filter coefficients. Finally, some design examples are included.**

## I. Introduction

Multirate filter banks find applications in a wide variety of digital signal processing systems such as subband coding, TDM–FDM transmultiplexers, spectral estimation, and adaptive signal processing [1], [2]. When the filter banks are applied to such applications, it is desirable that the analysis and synthesis filters have linear phase (LP) and that the filter bank achieves perfect reconstruction. Filter banks with such properties have been well studied, and various design approaches have been successfully developed [3]–[11]. Nguyen *et al.* presented a design method for nonparaunitary filter banks with LP using lattice structure [7]. This method, however, consumes time since the objective function is a nonlinear function with many parameters. Moreover, it is well known that the nonlinear optimization involves the initial value problem, which increases the computational complexity for the approximation. In the two-channel case, the Lagrange multiplier method to design nonparaunitary filter banks was developed in [10]. On the other hand, Kurosawa *et al.* presented a efficient design procedure for three-channel PR filter banks that does not need nonlinear optimization [9]. These design methods have a demerit that the order of a highpass filter becomes higher than that of other analysis filters to make it have a good amplitude characteristic. It has, however, attractive features such as that the procedure is simple and does not need the nonlinear optimization. Horng and Willson [10] also proposed a Lagrange–Newton method to overcome the problem of filter length.

Manuscript received February 28, 1995; revised April 25, 1996. The associate editor coordinating the review of this paper and approving it for publication was Prof. Roberto H. Bamberger.

The authors are with the Department of Electrical Engineering, Keio University, Yokohama, Japan (e-mail: ikehara@tkhm.elec.keio.ac.jp).

Publisher Item Identifier S 1053-587X(97)06451-9.

In this correspondence, we present two efficient approaches to design $M$-channel nonparaunitary filter banks with LP based on these methods [9], [10]. In the first approach, we first design suitably $M-1$ analysis filters. Then, the remainder of the analysis system that has highpass response is designed under the condition that the determinant of the polyphase matrix is monomial. If no degree of freedom exists, a highpass filter is uniquely determined but has poor response. Therefore, the external parts that guarantee the PR and LP are added to improve the highpass response. When there are some degrees of freedom, we design the highpass filter directly by using a Lagrange multiplier method. Second approach is based on a Lagrange–Newton method. At first, $M-2$ suitably analysis filters are designed, and then, the remainders (for example, lowpass and highpass filters) are optimized simultaneously by using the Lagrange–Newton method. One can overcome the problem such that the order of the highpass filter becomes very high in the first approach. It is possible to design the filter bank with same order. This is because the PR condition is not imposed on only a highpass filter but also a lowpass filter to increase the degree of freedom for designing the PR system. We also formulate the problems, which are difficult in the $M$-channel case, to solve the simultaneous equation of the PR constraint and express it as linear and nonlinear equations of the filter coefficients.

In the next section, we show the preliminaries for $M$-channel nonparaunitary PR filter banks with LP. Section III presents two design methods when no degree and some degrees of freedom exist for a highpass filter. In Section IV, a Lagrange–Newton method is proposed. Then, we describe the efficient method to solve the simultaneous equation of PR constraint in Section V. Finally, we show some examples.

## II. Preliminaries

### A. $M$-Channel LP PR Filter Banks

Fig. 1 shows a $M$-channel filter bank. The analysis filter $H_k(z)$ and synthesis filter $G_k(z)$ can be written in the polyphase forms as

$$H_k(z) = \sum_{l=0}^{M-1} z^{-l} E_{k,l}(z^M) \tag{1}$$

$$G_k(z) = \sum_{l=0}^{M-1} z^{-(M-1-l)} R_{l,k}(z^M). \tag{2}$$

Defining the two matrices $\boldsymbol{E}(z)$ and $\boldsymbol{R}(z)$ as the polyphase component matrices, the PR constraint can be written as [3]

$$\boldsymbol{R}(z)\boldsymbol{E}(z) = z^{-r}\boldsymbol{I} \tag{3}$$

where $r$ is some positive integer. Then, the synthesis bank is obtained by

$$\boldsymbol{R}(z) = z^{-r}\boldsymbol{E}^{-1}(z). \tag{4}$$