PAPER Special Section on Digital Signal Processing

FFT-Based Implementation of Sampling Rate Conversion with a Small Number of Delays

Xiaoxia ZOU†, Shogo MURAMATSU†, *and* Hitoshi KIYA†, *Members*

SUMMARY Block delay caused by using fast Fourier transform (FFT), and computational complexity in sampling rate conversion system are considered in this paper. The relationship between the number of block delays and the computational complexity is investigated. The proposed method can avoid the redundant operations of sampling rate conversion completely and moreover provide a good trade-off between the number of block delays and the computational complexity. As a result, it is shown that with the proposed method, the sampling rate conversion can be realized more efficiently under a small number of block delays.

key words: *multirate signal processing, sampling rate conversion, overlap-add/save method*

1. Introduction

Sampling rate conversion [1], [2] widely used in subband coding [3], adaptive signal processing [4], A/D and D/A conversions [5] etc., is an important topic in digital signal processing. A general model for sampling rate conversion with a rational ratio of U/D is shown in Fig. 1, where U and D are two positive integers. Note that the up-sampler inserts $U-1$ zero-samples between every two adjacent samples of $x(n)$, while the down-sampler picks up only one out of every D samples of $v(n)$ and discards the others. Therefore both up-sampler and down-sampler result in redundant operations, which should be avoided so that the sampling rate conversion can be performed efficiently.

Several methods such as the extended overlap-add and -save methods (EOLA and EOLS) [6] and the polyphase repersentation method [7], which can avoid the redundant operations, have been proposed. Among these methods, the polyphase representation method can avoid the redundant operations completely by segmenting the FIR filter $h(n)$. However, when a higher-order FIR filter is used, the computational complexity, which is directly proportional to the order of the FIR filter, may become very large.

In order to perform the FIR filtering efficiently, block filtering techniques such as overlap-add and -save methods (OLA or OLS) using FFT are often employed [8], [9]. Although computational saving can be achieved with these block techniques, an additional input-output delay in proportion to the block size, called block delay in this paper, has to be paid [10], [11]. When a high-order FIR filter is used, the number of block delays becomes very large if a small number of computations is desired. As a result, the real-time realization becomes difficult in this case.

Therefore, the purpose of this paper is to reduce both the computational complexity and the number of block delays. To accomplish this, the polyphase representation method and the generalized OLA and OLS [11] are used. We show that it is possible, with our proposed method, to make an optimal choice of the block size so that a good tradeoff between the number of block delays and the computational complexity can be achieved.

The paper is organized as follows: Sect. 2 introduces the polyphase representation for sampling rate conversion. Section 3 shows the proposed method, and Sect. 4 discusses the proposed algorithm under a special condition. The conclusions are given in Sect. 5.

2. Polyphase Representation for Sampling Rate Conversion System

2.1 Polyphase Representation

Consider the sampling rate conversion shown in Fig. 1. It has been shown in [7] that, if U and D are mutually prime numbers, such sampling rate conversion can be implemented more efficiently by using polyphase representation. Figure 2 shows the system structure with this method.

In the figure, the input sequence $x(n)$ is divided into D subsequences defined by

$$x_d(n) = x(-d + nD), \quad d = 0, 1, \ldots, D-1, \qquad (1)$$

and the output sequence $y(n)$ is divided into U subsequences, which are related to $y(n)$ by

$$y_u(n) = y(u + nU), \quad u = 0, 1, \ldots, U-1. \qquad (2)$$

The FIR filter $h_{u,d}(n)$ is defined by

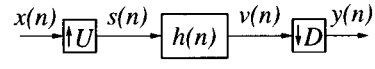


Fig. 1 General model for sampling rate conversion with a rational ratio of U/D.

Manuscript received December 10, 1996.
Manuscript revised March 4, 1997.

†The authors are with the Graduate School of Engineering, Tokyo Metropolitan University, Hachioji-shi, 192-03 Japan.

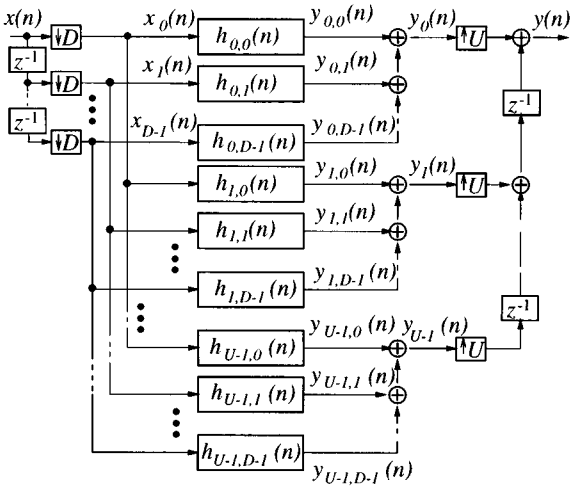


Fig. 2 The sampling rate conversion system structured by the polyphase representation.

$$h_{u,d}(n) = h(uD + dU + nDU),$$
$$n = \begin{cases} 0, 1, \ldots, \dfrac{L}{UD} - 1, & \dfrac{u}{U} + \dfrac{d}{D} < 1 \\ -1, 0, \ldots, \dfrac{L}{UD} - 2, & \dfrac{u}{U} + \dfrac{d}{D} \geq 1 \end{cases} \quad (3)$$

where $h(n)$ is the FIR filter in Fig. 1, and L is the length of $h(n)$. It can be seen that $h_{u,d}(n)$ are the polyphase components of the FIR filter $h(n)$ when $h(n)$ is divided into $D \times U$ FIR filters. Here we suppose that L is a multiple of UD. This condition can always be satisfied by padding sufficient zeros to the tail of $h(n)$.

It can be seen from Fig. 2 that all FIR filters $h_{u,d}(n)$ are at the lowest sampling rate side. This implies that all redundant operations due to the sampling rate conversion are avoided. However, when a high-order FIR filter $h(n)$ is used, the computational complexity, which is directly proportional to the order of the FIR filter $h(n)$, may become very large.

2.2 FFT-Based Implementation and Block Delay

In this section, we show why the use of FFT causes block delay, and the number of block delays in the sampling rate conversion.

With OLA or OLS, the FIR filtering of $x_d(n)$ and $h_{u,d}(n)$ in Fig. 2 can be performed in DFT-domain. In this case, $y_u(n)$ is obtained as shown in Fig. 3 (a), where $H_{u,d}(k)$ is the N-point DFT of $h_{u,d}(n)$. Note that the output of the filtering and the filtering sum are respectively denoted by $y'_{u,d}(n)$ and $y'_u(n)$ instead of $y_{u,d}(n)$ and $y_u(n)$. This is because the FFT-based method in Fig. 3 (a) segments $x_d(n)$ into blocks, an additional input-output delay is resulted from the segmentation. Let the number of the delays be denoted by 'B,' then $y'_{u,d}(n)$ is related to $y_{u,d}(n)$ by $y'_{u,d}(n) = y_{u,d}(n - B)$. Accordingly, the following relationship holds:

$$y'_u(n) = y_u(n - B), \quad (4)$$

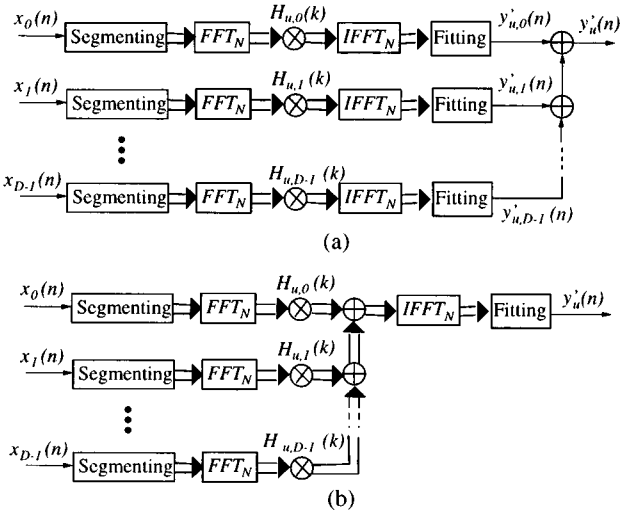


Fig. 3 The way of computing $y_u(n)$ by FFT and IFFT. $\otimes$ denotes element-by-element multiplication. $y'_u(n)$ is equivalent to $y_u(n)$ but delays due to the segmentation.

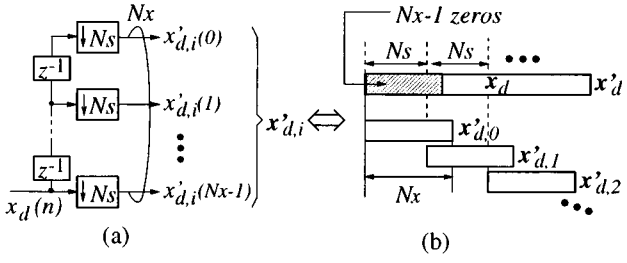


Fig. 4 Segmenting $x_d(n)$ into blocks.

where $y_u(n)$ is obtained from the filtering sum shown in Fig. 2, i.e., $y_u(n) = \sum_{d=0}^{D-1} y_{u,d}(n)$, and $y'_u(n)$ is obtained from the corresponding sum in the FFT approach shown in Fig. 3 (a), i.e., $y'_u(n) = \sum_{d=0}^{D-1} y'_{u,d}(n)$.

In Fig. 3 (a), since both the 'IFFT' and 'fitting' operations are linear, they can be moved right to pass the sum operation. This results in Fig. 3 (b), which has the same input-output relationship with Fig. 3 (a).

In the following discussion, we show the procedure for obtaining $y'_u(n)$ in Fig. 3 (b), and then lead the value of B.

Step 1: Segment $x_d(n)$ into block sequences.

We assume that $x_d(n) = 0$ for $n < 0$. Under this assumption, $x_d(n)$ is segmented into blocks for OLA and OLS as shown in Fig. 4 (b), and this segmentation corresponds to the multirate system in Fig. 4 (a) [12]. Here N_s is the amount of shift between two successive blocks, and N_x is the block size. For OLA $N_x = N_s < N$ and for OLS $N_x = N > N_s$, where N is the DFT size (See Eq. (24)). $\boldsymbol{x}_d$, $\boldsymbol{x}'_d$ and $\boldsymbol{x}'_{d,i}$ $(i = 0, 1, \ldots, \infty)$ are the vector representations of $x_d(n)$, $x'_d(n)$ and $x'_{d,i}(n)$ respectively, whose definitions will be given later. Note that $N_x - 1$ zeros are inserted to the left side of $\boldsymbol{x}_d$. The reason is that $N_x - 1$ samples have to be waited when

the data sequence $x_d(n)$ is segmented into blocks of size N_x. The waiting is equivalent to inserting $N_x - 1$ zeros to the left side of $\boldsymbol{x}_d$.

Now define $x'_d(n)$ as follows:

$$x'_d(n) = x_d(n - N_x + 1), \quad n = 0, 1, \ldots, \infty \tag{5}$$

then the i-th input block $x'_{d,i}(n)$ can be given by

$$x'_{d,i}(n) = x'_d(iN_s + n), \quad n = 0, 1, \ldots, N_x - 1. \tag{6}$$

Accordingly, $\boldsymbol{x}'_d$ and $\boldsymbol{x}'_{d,i}$ $(i = 0, 1, \ldots, \infty)$, which are the vector representations of $x'_d(n)$ and $x'_{d,i}(n)$ are defined by

$$\boldsymbol{x}'_d = [\overbrace{0\ 0 \cdots 0}^{N_x - 1}\ x_d(0)\ x_d(1) \cdots]^T \tag{7}$$

$$\boldsymbol{x}'_{d,i} = [x'_{d,i}(0)\ x'_{d,i}(1) \cdots x'_{d,i}(N_x - 1)]^T. \tag{8}$$

$\boldsymbol{x}_d$, which is the vector representation of $x_d(n)$, is defined by

$$\boldsymbol{x}_d = [x_d(0)\ x_d(1)\ x_d(2) \cdots]^T. \tag{9}$$

Step 2: Compute the N-point FFTs of $x'_{d,i}(n)$ $(d = 0, 1, \ldots, D-1)$, multiply them by the N-point FFTs of $h_{u,d}(n)$ $(d = 0, 1, \ldots, D-1)$, and then add the resulting sequences together as follows:

$$Y'_{u,i}(k) = \sum_{d=0}^{D-1} X'_{d,i}(k) H_{u,d}(k), \quad k = 0, 1, \ldots, N-1 \tag{10}$$

where $X'_{d,i}(k)$ and $H_{u,d}(k)$ are the N-point FFTs of $x'_{d,i}(n)$ and $h_{u,d}(n)$ respectively.

Step 3: Perform the N-point IFFT of $Y'_{u,i}(k)$ to yield $y'_{u,i}(n)$.

Step 4: Fit $y'_{u,i}(n)$ $(i = 0, 1, \ldots, \infty)$ together to form the overall output $y'_u(n)$ as follows:

$$y'_u(n) = \sum_{i=0}^{\infty} w_{u,i}(n - iN_s), \quad n = 0, 1, \ldots, \infty \tag{11}$$

where

$$w_{u,i}(n) = y'_{u,i}(n), n = 0, 1, \ldots, N-1$$

for OLA, and

$$w_{u,i}(n) = y'_{u,i}(n + N - N_s), n = 0, 1, \ldots, N_s - 1$$

for OLS. Such fitting can be implemented as shown in Figs. 5 (a) and (b), where $\boldsymbol{y}'_{u,i}$ $(i = 0, 1, \ldots, \infty)$ and $\boldsymbol{y}'_u$ are the vector representations of $y'_{u,i}(n)$ and $y'_u(n)$, i.e.,

$$\boldsymbol{y}'_{u,i} = [y'_{u,i}(0)\ y'_{u,i}(1) \cdots y'_{u,i}(N-1)]^T \tag{12}$$

$$\boldsymbol{y}'_u = [y'_u(0)\ y'_u(1) \cdots y'_u(N-1)]^T. \tag{13}$$

Now, let us show the value of B based on the above discussion. According to the convolution-multiplication property of DFT, Eq. (10) corresponds to

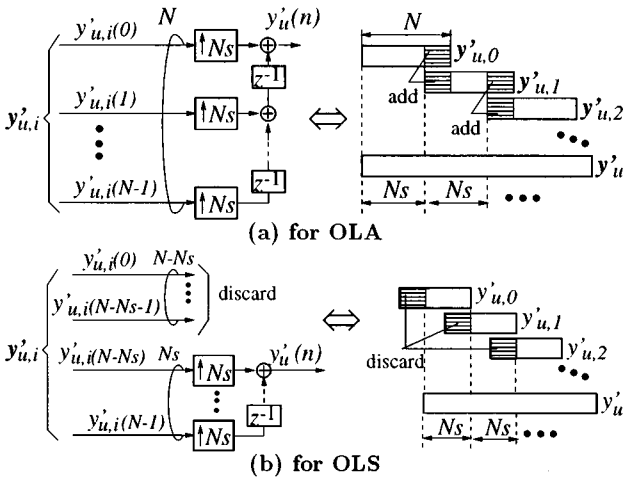


Fig. 5 Fitting $y'_{u,i}(n)$ $(i = 0, 1, \ldots, \infty)$ into $y'_u(n)$.

the sum of the N-point circular convolutions: $y'_{u,i}(n) = \sum_{d=0}^{D-1} x'_{d,i}(n) \textcircled{N} h_{u,d}(n)$, $n = 0, 1, \ldots, N-1$, where $\textcircled{N}$ denotes N-point circular convolution operation. Furthermore, according to the relationship between linear convolution and circular convolution, Eq. (10) also corresponds to the following sum of the linear convolutions:

$$y'_{u,i}(n) = \sum_{d=0}^{D-1} x'_{d,i}(n) * h_{u,d}(n), \quad n = N_x - N_s, N_x - N_s + 1, \ldots, N-1 \tag{14}$$

where '$*$' denotes linear convolution operation, $N_x = N_s$ for OLA, and $N_x = N$ for OLS. Now substituting Eq. (14) into Eq. (11), and using the following relationship obtained from Eqs. (5) and (6),

$$x_d(n - N_x + 1) = \sum_{i=0}^{\infty} x'_{d,i}(n - iN_s), \quad n = 0, 1, \ldots, \infty \tag{15}$$

we get

$$y'_u(n) = \sum_{d=0}^{D-1} x_d(n - N_s + 1) * h_{u,d}(n) \tag{16}$$

for both OLA and OLS. On the other hand, from Fig. 2 we have

$$y_u(n) = \sum_{d=0}^{D-1} x_d(n) * h_{u,d}(n). \tag{17}$$

Obviously, $y'_u(n) = y_u(n - N_s + 1)$. Therefore, the value of B in Eq. (4) is

$$B = N_s - 1. \tag{18}$$

Next, to lead the number of the delays in the final output of the sampling rate conversion, let us show in

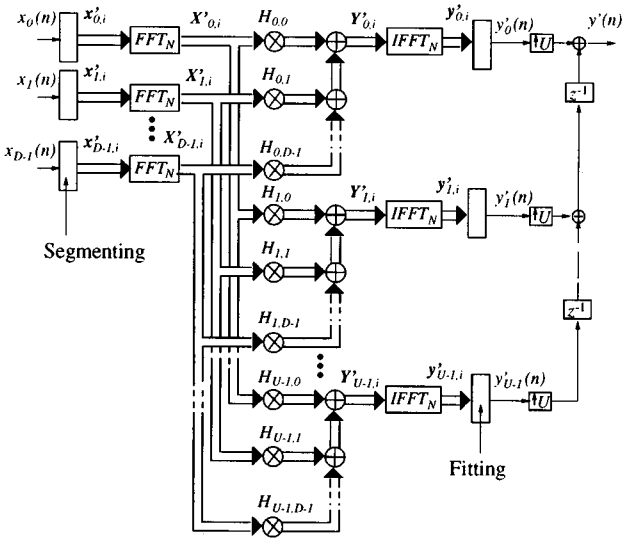


Fig. 6 The FFT-based implementation of the sampling rate conversion in Fig. 2. ⊗ denotes element-by-element multiplication.

Fig. 6 the whole structure of the sampling conversion system based on the FFT approach. In Fig. 6, $\boldsymbol{X}'_{d,i}$, $\boldsymbol{H}_{u,d}$ and $\boldsymbol{Y}'_{u,i}$ are the vector representations of $X'_{d,i}(k)$, $H_{u,d}(k)$ and $Y'_{u,i}(k)$ respectively, i.e.,

$$\boldsymbol{X}'_{d,i} = [X'_{d,i}(0)\ \ X'_{d,i}(1)\cdots X'_{d,i}(N-1)]^T \quad (19)$$

$$\boldsymbol{H}_{u,d} = [H_{u,d}(0)\ \ H_{u,d}(1)\cdots H_{u,d}(N-1)]^T \quad (20)$$

$$\boldsymbol{Y}'_{u,i} = [Y'_{u,i}(0)\ \ Y'_{u,i}(1)\cdots Y'_{u,i}(N-1)]^T. \quad (21)$$

It can be seen from Fig. 6 that the sampling rate of $y'(n)$ is U times the one of $y'_u(n)$, therefore the $N_s - 1$ delays in $y'_u(n)$ lead to $U(N_s-1)$ delays in $y'(n)$. Consequently, the following relationship holds:

$$y'(n) = y(n - U(N_s - 1)). \quad (22)$$

The $U(N_s - 1)$ delays are called block delays in this paper. Let B_d be the number of block delays, then

$$B_d = U(N_s - 1). \quad (23)$$

It is obvious that the number of block delays can be reduced by choosing small input block size N_s. However, choosing small input block size results in large number of computations.

2.3 Relationship between Computational Complexity and Block Delay

Suppose that all input samples have real values. It can be seen from Fig. 6 that D N-point FFTs, U N-point IFFTs, $U \times D \times N$ complex multiplications and $(D-1) \times U \times N$ complex additions for Eq. (10) are needed for every $U \times N_s$ output samples of $y'(n)$. For OLA approach, other $U \times (N - N_s)$ real additions are necessary because of the overlap-adding.

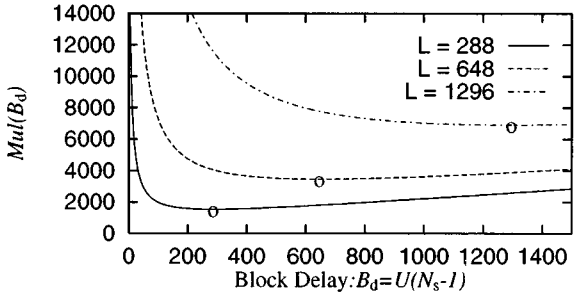


Fig. 7 The relationship between the number of block delays and the number of multiplications per output sample for $U = 3$ and $D = 1$. L is the length of the FIR filter $h(n)$. DFT is computed directly without using fast algorithm. The point marked by empty circle corresponds to the minimum Mul.

For the filtering of the length-$L/(UD)$ filter $h_{u,d}(n)$ with the length-N_s input block $x'_{d,i}(n)$, the DFT size N has to satisfy

$$N \geq N_s + L/(UD) \quad (24)$$

because of wrap-around effects (Note that $h_{u,d}(n)$ is noncausal for $u/U + d/D \geqq 1$). Also from Eqs. (23), (24) can be rewritten to:

$$N \geq B_d/U + L/(UD) + 1. \quad (25)$$

Let Mul and Add respectively be the number of real-valued multiplications and additions per output sample of $y'(n)$, where we assume that a complex multiplication is done by three real additions and three real multiplications [9]. Then Mul and Add can be computed as follows:

$$Mul = \frac{(D+U)\mu(N) + 3UDN/2}{UN_s} \quad (26)$$

$$Add = \frac{(D+U)\alpha(N) + UND(\frac{5}{2} - \frac{1}{D}) + \beta}{UN_s}, \quad (27)$$

where $\mu(N)$ and $\alpha(N)$ denote the numbers of real multiplications and additions for N-point FFT(or IFFT) respectively, and β is given by

$$\beta = \begin{cases} L/D & \text{for OLA} \\ 0 & \text{for OLS} \end{cases}. \quad (28)$$

Since both N and N_s in Eqs. (26) and (27) can be determined by the number of block delays B_d, Mul and Add can be considered as the functions of B_d.

As an example, let $U = 3$ and $D = 1$. First, let us consider the case without any FFT for the computation of DFT. In Fig. 7, we plot the Mul's for $L = 288$, 648 and 1296, where L is the length of $h(n)$. Note that for every L, there is a minimum Mul (marked by empty circle). Let the minimum Mul be denoted by Mul_O, and let the number of block delays corresponding to Mul_O be denoted by B_O. We can see that B_O increases with L monotonically, and Mul increases as B_d decreases in

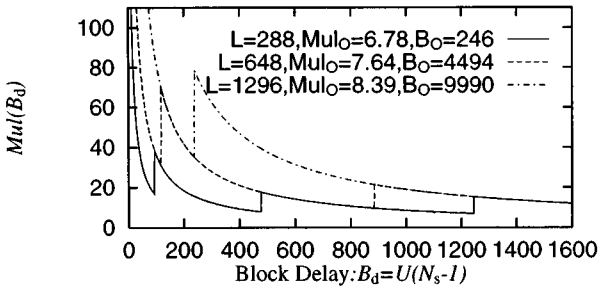


Fig. 8 The relationship between B_d and Mul for $U = 3$ and $D = 1$, as the split-radix FFT is applied to the example of Fig. 7.

the range of $B_d \leqq B_O$. Therefore, when small number of block delays is required by the real-time system, Mul becomes very large for high-order $h(n)$.

Next, let us consider the case with an FFT. To take a simple example, suppose that the DFT size N, which should satisfy Eq. (24), is a power of 2 so that the split-radix FFT[13] can be used. The Mul's, Mul_O's and B_O's in this case are show in Fig. 8, where those sharp increases on Mul are caused by the sudden increases of the DFT size. As well as in Fig. 7, one can see that B_O increases with L monotonically, and when small B_d is required, Mul becomes very large, especially for large L.

Although the above conclusions are derived for multiplications, they also hold for additions. To sum up, when small number of block delays is required, the number of computations becomes very large for high-order FIR filter $h(n)$. Since both the number of block delays and the computational complexity are well concerned in the real-time system, a tradeoff between them should be obtained, which is the purpose of our proposed method.

3. New Implementation Method of FFT-Based Sampling Rate Conversion

In this section, we use the generalized OLA and OLS techniques[11] for the implementation of the sampling rate conversion. These techniques, which were proposed for ordinary FIR filtering, segment both input sequence and FIR filter into blocks so that large-sized convolution of an input block with a high-order FIR filter can be converted into small-sized convolutions.

3.1 New Implementation Method

Suppose that the length-$L/(UD)$ filter $h_{u,d}(n)$ can be divided into P FIR filters of length L_s, i.e.,

$$L_s = L/(PUD), \tag{29}$$

as follows:

$$h_{u,d}(n) = \sum_{p=0}^{P-1} h_{u,d}^p(n - pL_s), \tag{30}$$

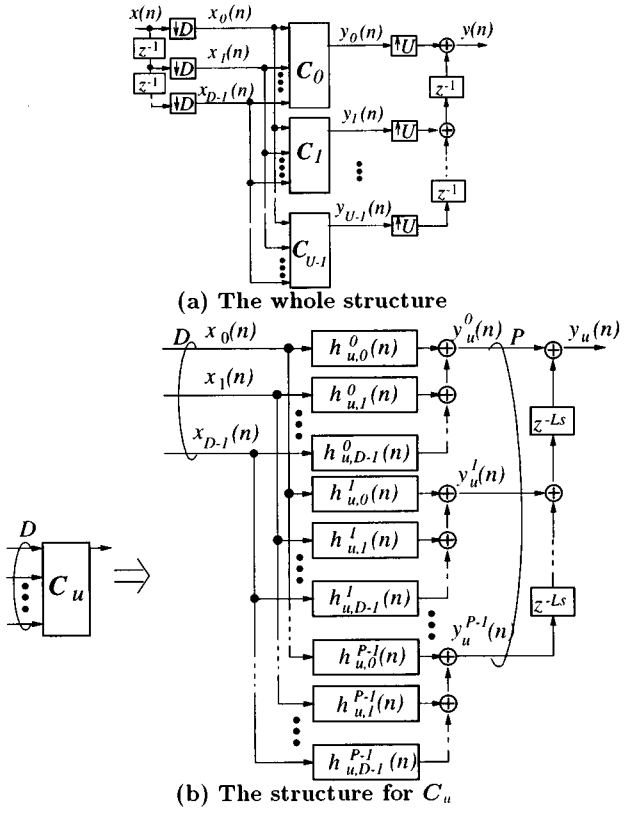


Fig. 9 The structure of the sampling rate conversion system when the FIR filter $h_{u,d}(n)$ is segmented into P filters $h_{u,d}^p(n)$ $(p = 0, 1, \ldots, P-1)$ of length $L_s = L/(PUD)$. The input-output relationship is the same as that of Fig. 2.

where $h_{u,d}^p(n)$ is the pth segment of $h_{u,d}(n)$,

$$h_{u,d}^p(n) = h_{u,d}(pL_s + n), \quad n = 0, 1, \ldots, L_s - 1. \tag{31}$$

Then the system structure of Fig. 2 can be converted to the one shown in Fig. 9 according to the following instruction.

Using Eq. (30), we can compute $y_u(n)$ of Fig. 2 as follows:

$$\begin{aligned} y_u(n) &= \sum_{d=0}^{D-1} x_d(n) * h_{u,d}(n) \\ &= \sum_{d=0}^{D-1} x_d(n) * \sum_{p=0}^{P-1} h_{u,d}^p(n - pL_s). \end{aligned} \tag{32}$$

Next, using the linearity property of convolution, we have

$$y_u(n) = \sum_{p=0}^{P-1} \sum_{d=0}^{D-1} x_d(n) * h_{u,d}^p(n - pL_s), \tag{33}$$

which can be expressed in z-domain by

$$Y_u(z) = \sum_{p=0}^{P-1} \left\{ \sum_{d=0}^{D-1} X_d(z) H_{u,d}^p(z) \right\} z^{-pL_s}, \tag{34}$$

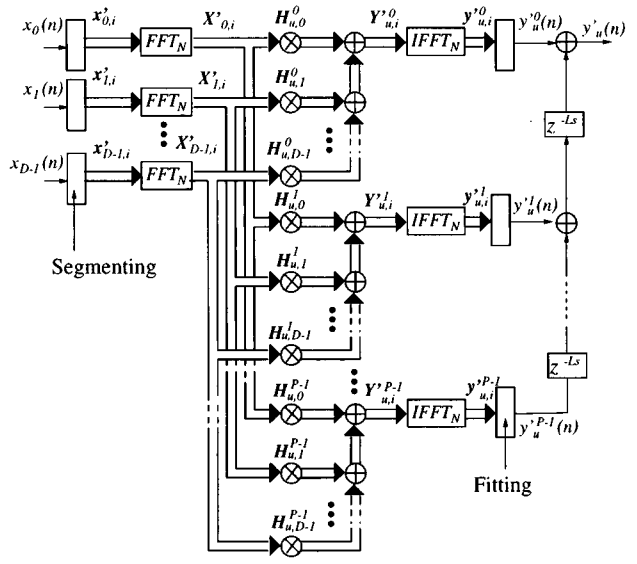


Fig. 10 The FFT-based implementation for C_u.

where $Y_u(z)$, $X_d(z)$ and $H^p_{u,d}(z)$ are the z-transforms of $y_u(n)$, $x_d(n)$ and $h^p_{u,d}(n)$, respectively. Equation (34) indicates that the convolution of $x_d(n)$ with $h_{u,d}(n)$ can be converted into small-sized convolutions of $x_d(n)$ with $h^p_{u,d}(n)$ for $p = 0, 1, \ldots, P-1$. Therefore, the system structure in Fig. 2 can be changed to the one shown in Fig. 9, where $y^p_u(n)$ and $y_u(n)$ are obtained as follows:

$$y^p_u(n) = \sum_{d=0}^{D-1} x_d(n) * h^p_{u,d}(n), \tag{35}$$

$$y_u(n) = \sum_{p=0}^{P-1} y^p_u(n - pL_s), \tag{36}$$

according to Eq. (33).

Now let us implement C_u of Fig. 9 (b) in DFT-domain. Note that the structure for computing $y^p_u(n)$ in Fig. 9 (b) is similar to that for computing $y_u(n)$ in Fig. 2, hence as $y_u(n)$ can be obtained in FFT approach as shown in Fig. 6, so $y^p_u(n)$ can be obtained similarly as shown in Fig. 10, in which $\boldsymbol{H}^p_{u,d}$, $\boldsymbol{Y}'^p_{u,i}$ and $\boldsymbol{y}'^p_{u,i}$ are the vectors defined as follows:

$$\boldsymbol{H}^p_{u,d} = [H^p_{u,d}(0)\ \ H^p_{u,d}(1) \cdots H^p_{u,d}(N-1)]^T, \tag{37}$$

$$\boldsymbol{Y}'^p_{u,i} = [Y'^p_{u,i}(0)\ \ Y'^p_{u,i}(1) \cdots Y'^p_{u,i}(N-1)]^T, \tag{38}$$

$$\boldsymbol{y}'^p_{u,i} = [y'^p_{u,i}(0)\ \ y'^p_{u,i}(1) \cdots y'^p_{u,i}(N-1)]^T, \tag{39}$$

where $H^p_{u,d}(k)$ is the N-point DFT of $h^p_{u,d}(n)$, $Y'^p_{u,i}(k)$ is computed by

$$Y'^p_{u,i}(k) = \sum_{d=0}^{D-1} X'_{d,i}(k) H^p_{u,d}(k), \quad k = 0, 1, \ldots, N-1 \tag{40}$$

and $y'^p_{u,i}(n)$ is the N-point IDFT of $Y'^p_{u,i}(k)$.

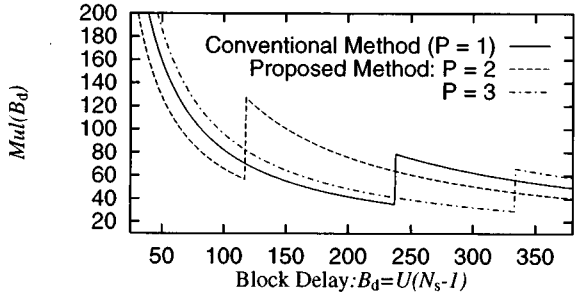


Fig. 11 The relationship among the block delay, the number of multiplications per output sample, and the segmentation number of $h_{u,d}(n)$ for the PR-SC method, when $U = 3$, $D = 1$ and $L = 1296$. Here the split-radix FFT is used.

The method shown in Fig. 9 is the proposed method, called PR-SC method (PR: Polyphase Representation, SC: Small-sized Convolution) in this paper.

3.2 Computational Complexity and Block Delay

According to Figs. 9 and 10, we calculate the Mul and Add for the sampling rate conversion with PR-SC method as follows:

$$Mul = \frac{(D+PU)\mu(N) + 3UDNP/2}{UN_s} \tag{41}$$

$$Add = \frac{(D+PU)\alpha(N) + UDNP(\frac{5}{2} - \frac{1}{D})}{UN_s} + \frac{U(P-1)(N - L/PUD) + \beta}{UN_s} \tag{42}$$

where β is given by Eq. (28), and the DFT size N is given by

$$N \geqq B_d/U + L/(PUD) + 1. \tag{43}$$

Recall the example of Sect. 2.3, and let the length of $h(n)$ be 1296. Suppose that the split-radix FFT is used. Figure 11 shows the Mul's for $P = 1$, 2 and 3, where $P = 1$ corresponds to the conventional method shown in Fig. 8. It can be seen that in the range of $B_d \leqq 110$, the proposed method for $P = 2$ requires fewer multiplications, compared to the conventional one. For example, given $B_d = 100$, about 20% computational saving can be achieved.

Similar conclusion can also be made for additions. Thus, we can conclude that the proposed method can trade off the number of block delays against the number of computations more easily when a small number of block delays is required.

4. PR-SC Method Using Fewer IFFTs

4.1 Reducing the Number of IFFTs

In Fig. 10, P IFFTs are used. In this section, we show

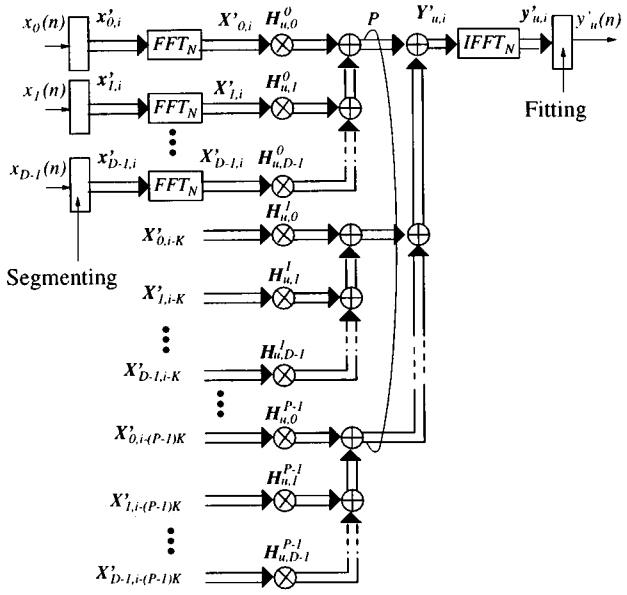


Fig. 12 Computing $y'_{u,i}(n)$ with only one IFFT. The input-output relationship is the same as that of Fig. 10.

that if L_s, which is the length of $h^p_{u,d}(n)$, is a multiple of N_s, only one IFFT is needed, and C_u can be implemented with the structure shown in Fig. 12.

Let us consider the OLA case at first. In Fig. 10, the output sequence $y'_u(n)$ is obtained by

$$y'_u(n) = \sum_{p=0}^{P-1} y'^p_u(n - pL_s), \quad n = 0, 1, \ldots, \infty, \tag{44}$$

where $y'^p_u(n)$ $(p = 0, 1, \ldots, P-1)$ can be formed by fitting $y'^p_{u,i}$ according to Eq. (11), i.e.,

$$y'^p_u(n) = \sum_{i=0}^{\infty} y'^p_{u,i}(n - iN_s), \quad n = 0, 1, \ldots, \infty. \tag{45}$$

Substituting Eq. (45) into Eq. (44) yields

$$y'_u(n) = \sum_{p=0}^{P-1} \sum_{i=0}^{\infty} y'^p_{u,i}(n - iN_s - pL_s), \quad n = 0, 1, \ldots, \infty. \tag{46}$$

Next, note that the DFT products in Eq. (40) correspond to the following convolution sum,

$$y'^p_{u,i}(n) = \sum_{d=0}^{D-1} x'_{d,i}(n) * h^p_{u,d}(n), \quad n = 0, \ldots, N-1. \tag{47}$$

Substituting Eq. (47) into Eq. (46) yields

$$y'_u(n) = \sum_{p=0}^{P-1} \sum_{i=0}^{\infty} \sum_{d=0}^{D-1} x'_{d,i}(n - iN_s - pL_s) * h^p_{u,d}(n). \tag{48}$$

Now suppose that L_s is a multiple of N_s, i.e.,

$$L_s = KN_s, \tag{49}$$

where K is a positive integer. Then Eq. (48) becomes

$$\begin{aligned} y'_u(n) &= \sum_{p=0}^{P-1} \sum_{i=0}^{\infty} \sum_{d=0}^{D-1} x'_{d,i}(n - N_s(i + pK)) * h^p_{u,d}(n) \\ &= \sum_{p=0}^{P-1} \sum_{i=pK}^{\infty} \sum_{d=0}^{D-1} x'_{d,i-pK}(n - iN_s) * h^p_{u,d}(n). \end{aligned} \tag{50}$$

Since $x'_{d,i-pK}(n - iN_s) = 0$ for $i < pK$, the range of i can be changed from $i = pK, pK+1, \ldots, \infty$ to $i = 0, 1, \ldots, \infty$, and accordingly, Eq. (50) can be rewritten to

$$y'_u(n) = \sum_{i=0}^{\infty} \sum_{p=0}^{P-1} \sum_{d=0}^{D-1} x'_{d,i-pK}(n - iN_s) * h^p_{u,d}(n). \tag{51}$$

On the other hand, from Eq. (11) we have

$$y'_u(n) = \sum_{i=0}^{\infty} y'_{u,i}(n - iN_s). \tag{52}$$

Comparing Eq. (51) with Eq. (52), we get

$$y'_{u,i}(n) = \sum_{p=0}^{P-1} \sum_{d=0}^{D-1} x'_{d,i-pK}(n) * h^p_{u,d}(n). \tag{53}$$

Therefore the N-point DFT of $y'_{u,i}(n)$ can be calculated by

$$Y'_{u,i}(k) = \sum_{p=0}^{P-1} \sum_{d=0}^{D-1} X'_{d,i-pK}(k) H^p_{u,d}(k), \quad k = 0, 1, \ldots, N-1. \tag{54}$$

Similarly, one can prove that Eq. (54) also holds in the OLS case.

According to Eq. (54), we can lead the structure of Fig. 12, which has the same input-output relationship with Fig. 10 but uses only one IFFT.

4.2 Computational Complexity and Block Delay

Now let us consider the computational complexity and the block delay. According to Eqs. (49) and (29), N_s should satisfy

$$N_s = L/(KPUD). \tag{55}$$

Thus the number of block delays in this case is limited by

$$B_d = L/(KPD) - U. \tag{56}$$

Table 1 The possible choices of K, and the corresponding number of block delays.

$P=2$				$P=3$			
K	B_d	K	B_d	K	B_d	K	B_d
1	645	18	33	1	429	16	24
2	321	24	24	2	213	18	21
3	213	27	21	3	141	24	15
4	159	36	15	4	105	36	9
6	105	54	9	6	69	48	6
8	78	72	6	8	51	72	3
9	69	108	3	9	45		
12	51			12	33		

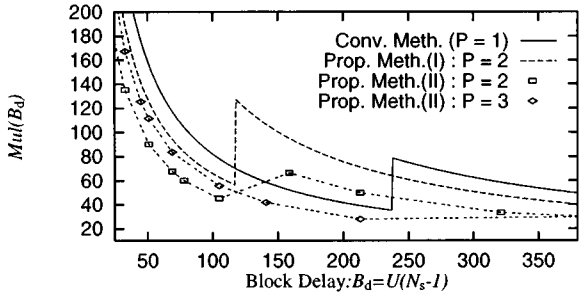


Fig. 13 The relationship among B_d, Mul and P for the conventional and the proposed methods, when $N_s = L/(KPUD)$, $L = 1296$, $U = 3$ and $D = 1$. Here the split-radix FFT is used. (Conv.: Conventional, Prop.: Proposed, Meth.: Method)

Next according to Figs. 12 and 9, we compute the Mul and Add as follows:

$$Mul = \frac{(D+U)\mu(N) + 3UDNP/2}{UN_s} \tag{57}$$

$$Add = \frac{(D+U)\alpha(N) + UDNP(\frac{5}{2} - \frac{1}{PD}) + \beta}{UN_s} \tag{58}$$

where $\beta = L/(PD)$ for OLA, $\beta = 0$ for OLS, and N is given by

$$N \geqq B_d/U + L/(PUD) + 1. \tag{59}$$

Consider the example of Sect. 3.2 again. In Table 1, we list all possible choices of K, and the corresponding numbers of block delays, for $P = 2$ and 3.

Suppose that the split-radix FFT is used again. The comparison of Mul's of the conventional and the proposed methods is show in Fig. 13, where Prop.Meth.(I) and (II) denote the proposed methods of Sect. 3 and this section respectively. It can be seen that Prop.Meth.(II) with $P = 2$ requires fewer multiplications in the range of $B_d \leqq 105$, and Prop.Meth.(II) with $P = 3$ does in the range of $B_d > 105$. Therefore, we can say that Prop.Meth.(II) is more efficient than the other methods under fewer number of block delays. As an example, let $B_d = 105$, Prop.Meth.(II) can reduce the number of multiplications by 28.3% compared to Prop.Meth.(I), and by 42.3% compared to the conventional method.

From this discussion, we can see that a good tradeoff between the number of block delays and the computational complexity can be achieved with Prop.Meth.(II).

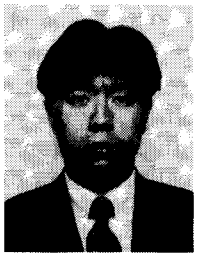
5. Conclusions

We have investigated the relationship between the block delays and the computational complexity of the sampling rate conversion implemented in DFT-domain, and have presented two new implementation approaches that can avoid the redundant operations and moreover achieve a good tradeoff between the number of block delays and the computational complexity. It has been shown that the sampling rate conversion with our proposed methods can be realized more efficiently under a small number of block delays.

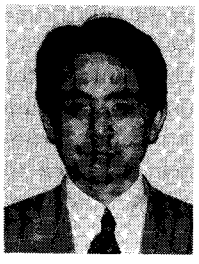
References

[1] R.E. Crochiere and L.R. Rabiner, "Multirate Digital Signal Processing," Prentice-Hall, Englewood Cliffs NJ, 1983.
[2] P.P. Vaidyanathan, "Multirate Systems and Filter Banks," Prentice-Hall, Englewood Cliffs NJ, 1993.
[3] J.W. Woods, "Subband Coding of Images," Kluwer Academic Publishers, 1981.
[4] A.N. Akansu and R.A. HAddad, "Adaptive Filtering in Subbands," IEEE ICASSP'88, pp.1572–1575, 1988.
[5] J.C. Candy and G.C. Temes, "Oversampling Delta-Sigma Data Converters," IEEE Press, 1992.
[6] S. Muramatsu and H. Kiya, "The extended OLA and OLS for sampling rate conversion," IEICE Trans., vol.J77-A, no.8, pp.1046–1055, Aug. 1994.
[7] H. Chia-Chuan, "Polyphase filter matrix for rational sampling rate conversions," IEEE Int. Conf. on ASSP, pp.2173–2176, Dallas, April 1987.
[8] J.G. Proakis and D.G. Manolakis, "Digital Signal Processing," Macmillian, 1988.
[9] H.J. Nussbaumer, "Fast Fourier Transform and Convolution Algorithms," Berlin: Springer, 1991.
[10] M. Vetterli, "Running FIR and IIR filtering using multirate filter banks," IEEE Trans. Acoust., Speech & Signal Process., vol.36, pp.730–738, May 1988.
[11] M. Vetterli, "Wavelets and Subband Coding," pp.361–363, Prentice-Hall, Englewood Cliffs NJ, 1995.
[12] M. Vetterli, "A theory of multirate filter banks," IEEE Trans. Acoust., Speech & Signal Process., vol.ASSP-35, pp.356–372, March 1987.
[13] H.V. Sorensen, M.T. Heideman, and C.S. Burrus, "On computing the split-radix FFT," IEEE Trans., Acoust., Speech & Signal Process., vol.ASSP-34, no.11, pp.152–156, Feb. 1986.

Xiaoxia Zou was born in Wanxian City, Sichuan Province, China, on April 30, 1968. She received the B.E. degree in mechanical engineering from Mechanical Engineering Institute of Shanghai, and the M.E. degree in electrical engineering from Tokyo Metropolitan University. She is currently working in NEC Informatec Systems, Ltd. Her research interests are in digital signal processing and image processing.



Shogo Muramatsu was born in Tokyo, Japan, on May 2, 1970. He received the B.E. and M.E. degrees in electrical engineering from Tokyo Metropolitan University, Tokyo, Japan, in 1993 and 1995, respectively. In 1997, he joined Tokyo Metropolitan University, where he is currently a research associate of Electrical Engineering, Graduate School of Engineering. His research interests are in digital signal processing, multirate systems, and image processing. Mr.Muramatsu is a member of the Institute of Electrical and Electronics Engineers, Inc. (IEEE) of USA.



Hitoshi Kiya was born in Yamagata, Japan, on November 16, 1957. He received the B.E. and M.E. degrees in electrical engineering from Nagaoka University of Technology, Niigata, Japan, and the D.E. degree in electrical engineering from Tokyo Metropolitan University, Tokyo, Japan, in 1980, 1982 and 1987, respectively. In 1982, he joined Tokyo Metropolitan University, where he is currently an Associate Professor of Electrical Engineering, Graduate School of Engineering. He was a visiting researcher of the university of Sydney in Australia from Oct.1995 to Mar.1996. His research interests are in digital signal processing, multirate systems, adaptive filtering, image processing, and efficient algorithms for VLSI implementation. Dr. Kiya is a Member of the Institute Electrical and Electronics Engineers, Inc. (IEEE) of USA, the Image Electronics Engineers of Japan and the Institute of Television Engineers of Japan.