Permuted Difference Coefficient Realization of FIR Digital Filters

KENJI NAKAYAMA

Abstract—A new realization for FIR digital filters, using permuted difference coefficients, is proposed in this paper. Its coefficients are obtained as the difference between the successive values of the original coefficients reordered in a sequence with falling magnitude. The proposed realization can hold desirable properties in FIR filters, such as an exactly linear phase characteristic and stable implementation, and is effectively applied to a wide range filter response. Quantization error analysis shows that the internal data word lengths must be somewhat increased to maintain the same roundoff noise as in a direct form realization. Computational complexity becomes about 23 percent and 18 percent for 99th- and 299th-order filters, taking the excess data word lengths into account, compared with the direct form.

I. Introduction

FIR filters can exactly realize a linear phase response and stable implementation. A high-order filter, however, is required for a high Q frequency response. Some limitation appears in a real time hardware realization. Therefore, computational complexity reduction in high-order FIR filters is very important. Several kinds of approaches to implement multiplying operations have been proposed by other simple ways, such as distributed arithmetic [1]-[3] and residue number systems [4]. The difference routing digital filter (DRDF) proposed by Gerwen *et al.* is one approach, which can sufficiently reduce multiplying operations for narrow-band and low Q filters [5]. Since the resonator coefficient values are restricted to integer, the obtainable filter responses are limited.

A new difference coefficient FIR digital filter realization, which does not lose the desirable features for FIR filters, such as a linear phase response and stable implementation, is proposed in this paper. The new difference coefficients are formed as the difference between the successive values of the original coefficients reordered in a sequence with falling magnitude. The proposed structure is called permuted difference coefficient digital filter (PDC-DF) in this paper. It can be effectively applied to a wide range filter response. The PDC-DF algorithm is described in Section II. Section III illustrates a hardware realization. Quantization error analysis is discussed in Section IV. Computational complexity and its numerical examples are described in Sections V and VI, respectively.

Manuscript received June 18, 1980; revised June 1, 1981.

The author is with the Transmission Division, Nippon Electric Company, Ltd., Kawasaki-City, Kanagawa, 211 Japan.

0096-3518/82/0400-0269$00.75 © 1982 IEEE

II. PDC-DF Algorithm

The PDC-DF algorithm can be summarized as follows: The original coefficients are reordered in a sequence with falling magnitude at first. New difference coefficients are formed as the difference between successive values for the reordered coefficients. This process is repeated, and high-order difference coefficients can be obtained.

Let $x(n)$ and $y(n)$ be the input and output signals for FIR filters, respectively. They satisfy

$$y(n) = \sum_{m=0}^{N-1} h_m x(n-m) \quad (1)$$

where h_m is an impulse response, and n is assumed to be larger than $N-1$. There are two kinds of forms, the transversal form and its transposed form. The PDC-DF algorithm can be applied to both forms. The transposed form can be obtained by reversing the signal flow in the first form. Therefore, only the transversal form is presented in this paper.

Let h_k^* be the absolute value for h_m, and let k be the reordered index. Then, h_k^* satisfies the following conditions

$$h_k^* = |h_m| \quad (2a)$$

and

$$0 \leqq h_0^* \leqq h_1^* \leqq h_2^* \cdots \leqq h_{N-1}^*. \quad (2b)$$

Let $x_k^*(n)$ be

$$x_k^*(n) = \text{sign}(h_m)\, x(n-m). \quad (3)$$

Equation (1) is rewritten using h_k^* and $x_k^*(n)$

$$y(n) = \sum_{k=0}^{N-1} h_k^* x_k^*(n). \quad (4)$$

First-order permuted difference coefficients $\Delta_k^{(1)}$ are defined by

$$\Delta_k^{(1)} = h_k^* - h_{k-1}^*, \quad k = 1, 2, \cdots, N-1 \quad (5a)$$

$$\Delta_0^{(1)} = h_0^*. \quad (5b)$$

Using $\Delta_k^{(1)}$, $y(n)$ can be expressed as

$$y(n) = \sum_{k=0}^{N-1} \Delta_k^{(1)} \cdot u_k^{(1)}(n) \quad (6)$$

where

$$u_k^{(1)}(n) = \sum_{i=k}^{N-1} x_i^*(n). \quad (7)$$

$u_k^{(1)}(n)$ can be calculated through the following accumulation

$$u_k^{(1)}(n) = u_{k+1}^{(1)}(n) + x_k^*(n), \quad k = 0, 1, \cdots, N-2 \quad (8a)$$

$$u_{N-1}^{(1)}(n) = x_{N-1}^*(n). \quad (8b)$$

The number of additions in (8) is N times.

Second-order permuted difference coefficients can be formed in the same way as the first-order difference coefficients. Let $\Delta_l^{*(1)}$ be $\Delta_k^{(1)}$, and l be the reordered index, that is

$$\Delta_l^{*(1)} = \Delta_k^{(1)} \quad (9a)$$

and

$$\Delta_0^{*(1)} \leqq \Delta_1^{*(1)} \leqq \cdots \leqq \Delta_{N-1}^{*(1)}. \quad (9b)$$

Second-order difference coefficients $\Delta_l^{(2)}$ are defined by

$$\Delta_l^{(2)} = \Delta_l^{*(1)} - \Delta_{l-1}^{*(1)}, \quad l = 1, 2, \cdots, N-1 \quad (10a)$$

$$\Delta_0^{(2)} = \Delta_0^{*(1)}. \quad (10b)$$

Using $\Delta_l^{(2)}$, $y(n)$ becomes

$$y(n) = \sum_{l=0}^{N-1} \Delta_l^{(2)} \cdot u_l^{(2)}(n) \quad (11)$$

where $u_l^{(2)}(n)$ is a partial sum of the reordered $u_k^{(1)}(n)$, represented as $u_l^{*(1)}(n)$ whose multiplicand is $\Delta_l^{*(1)}$

$$u_l^{(2)}(n) = \sum_{i=l}^{N-1} u_i^{*(1)}(n). \quad (12)$$

Equation (12) can be performed through N times additions as follows:

$$u_l^{(2)}(n) = u_{l+1}^{(2)}(n) + u_l^{*(1)}(n), \quad l = 0, 1, \cdots, N-2 \quad (13a)$$

$$u_{N-1}^{(2)}(n) = u_{N-1}^{*(1)}(n). \quad (13b)$$

The PDC-DF is actually performed through the following steps.

First-order PDC-DF	*Second-order PDC-DF*
Step 1: (3)	Step 1: (3)
Step 2: (8)	Step 2: (8)
Step 3: (6)	Step 3: (13)
	Step 4: (11).

Higher order permuted difference coefficients can be formed in the same way.

When the difference coefficients are rounded off, some of them become zero because difference coefficient magnitudes are well reduced from the original coefficients. A small number of difference coefficients can offer a small roundoff error and low computational complexity. Since the discussions in the following sections are carried out based on the reduced numbers of the difference coefficients. The equations described above are modified here.

Let N_1 and N_2 be the numbers of nonzero $\Delta_k^{(1)}$ and $\Delta_l^{(2)}$, respectively.

Equation (6) is rewritten

$$y(n) = \sum_{l=N-N_1}^{N-1} \Delta_l^{*(1)} \cdot u_l^{*(1)}(n) \quad (6')$$

where $u_l^{*(1)}(n)$ is $u_k^{(1)}(n)$ reordered according to the magnitude of $\Delta_k^{(1)}$. $u_k^{(1)}(n)$ is calculated through (8).

Equation (11) becomes

$$y(n) = \sum_{j=N-N_2}^{N-1} \Delta_j^{*(2)} u_j^{*(2)}(n) \quad (11')$$

where $\Delta_j^{*(2)}$ and $u_j^{*(2)}(n)$ are reordered $\Delta_l^{(2)}$ and $u_l^{(2)}(n)$, respectively.

Equation (13) becomes

$$u_l^{(2)}(n) = u_{l+1}^{(2)}(n) + u_l^{*(1)}(n),$$

$$l = N - N_1, N - N_1 + 1, \cdots, N - 2 \quad (13a')$$

$$u_{N-1}^{(2)}(n) = u_{N-1}^{*(1)}(n). \quad (13b')$$

The $u_k^{(1)}(n)$ and the $u_l^{(2)}(n)$ calculations require N times and N_1 times additions, respectively. N_2 times multiplications are performed to obtain the product of $\Delta_l^{(2)}$ and $u_l^{(2)}(n)$ as (11′).

III. Hardware Realization

A hardware realization is described for the second-order PDC-DF. The ninth-order filter is used to illustrate the structure. The original coefficients, the first- and second-order difference coefficients, are shown in Table I. The hardware realization is illustrated in Fig. 1. The numbers of nonzero $\Delta_k^{(1)}$ and $\Delta_l^{(2)}$, that is N_1 and N_2, become six and two, respectively. *DMn* is a data mapping block, in which the input data are reordered according to the corresponding coefficient magnitudes. The data mapping block can be easily realized, using a random access memory (RAM) together with a read only memory (ROM), where address data are stored. The sign bit multiplier is simply realized using an EX-OR gate only. A plus sign bit is 0 and a minus sign bit is 1. Three examples are shown below.

	(a)		(b)		(c)
Input	00.10101	Input	00.10101	Input	11.01010
sign (+)	00.00000	sign (-)	11.11111	sign (-)	11.11111
Output	00.10101	Output	11.01010	Output	00.10101

In the case of a minus sign bit, the output has error 11.11111, that is 11.01010 = 11.01011 + 11.11111 (b). In other words, the minus sign multiplier, using an EX-OR gate only, produces roundoff error, while no additions are required. The error, however, is always constant, which is 11.11111 for example. Then the filter output noise caused by the minus sign bit multipliers can be canceled exactly. Therefore, the sign bit multipliers are not taken as the roundoff noise sources in this paper. In this structure, the data mapping operation from l to j is employed, in order to perform the $\Delta_l^{(2)}$ multiplication effectively. The product of $u_l^{(2)}(n)$ and $\Delta_l^{(2)}$ calculation requires N_1 times multiplications in a time multiplexed hardware realization. Therefore, N_2 data corresponding to the nonzero difference coefficients $\Delta_j^{*(2)}$ are extracted to perform only N_2 times multiplications.

The scaling factors are shown by S_n ($n = 1, 2, 3$), in the same figure. S_2 is included in the second-order difference coefficients $\Delta_j^{*(2)}$. They are used to minimize the output roundoff noise in the PDC-DF, as discussed in the next section.

IV. Quantization Error Analysis

A. Sensitivity Analysis

There are two kinds of rounding off methods for the permuted difference coefficients. One method is to round off the original coefficients because the difference coefficient forming process does not cause any roundoff error. The other method is to use the original unrounded off coefficients in the process of forming the difference coefficients and do the rounding off on the final result. Roundoff error analysis in the former method is the same as in a direct realization. The sensitivity analysis was discussed by Chan *et al.* [6], and the results are used here.

TABLE I
ORIGINAL AND PERMUTED DIFFERENCE COEFFICIENTS IN NINTH-ORDER FILTER

m	hm	k	h_k^*	$\Delta_k^{(1)}$	l	$\Delta_l^{*(1)}$	$\Delta_l^{(2)}$	j	$\Delta_j^{*(2)}$
0	0.25	0	0.0	0.0	0	0.0	0.0	0	0.0
1	-0.625	1	0.25	0.25	1	0.0	0.0	1	0.0
2	-0.375	2	0.375	0.125	2	0.0	0.0	2	0.0
3	0.875	3	0.375	0.0	3	0.0	0.0	3	0.0
4	0.375	4	0.5	0.125	4	0.125	0.125	4	0.0
5	0.5	5	0.625	0.125	5	0.125	0.0	5	0.0
6	0.0	6	0.625	0.0	6	0.125	0.0	6	0.0
7	-0.875	7	0.875	0.25	7	0.125	0.0	7	0.0
8	0.625	8	0.875	0.0	8	0.25	0.125	8	0.125
9	1.0	9	1.0	0.125	9	0.25	0.0	9	0.125

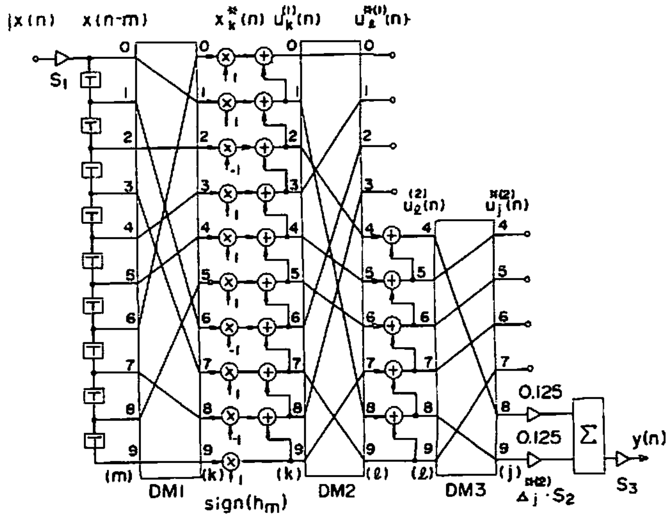


Fig. 1. Second-order PDC-DF hardware realization for ninth-order filter shown in Table I.

Let ϵ_m be the roundoff error in h_m; then the transfer function error $\Delta H(z)$ is expressed as

$$\Delta H(z) = \sum_{m=0}^{N-1} \epsilon_m z^{-m}. \quad (14)$$

Since ϵ_m is assumed to be distributed uniformly in the region $[-(\Delta c/2), (\Delta c/2)]$, and be white noise, $|\Delta H(e^{j\omega})|^2$ can be estimated from the Parseval relation as

$$|\Delta H(e^{j\omega})|^2 = \sum_{m=0}^{N-1} \epsilon_m^2 = N \frac{\Delta_c^2}{12}, \quad \Delta_c = 2^{-t_c} \quad (15)$$

where t_c is the least significant bit. Δ_c is expressed using the original coefficient word lengths L_0 and its maximum value $\max_m |h_m|$

$$\Delta_c = \max_m |h_m| \, 2^{-L_0}. \quad (16)$$

Furthermore, $\max_m |h_m|$ is related to bandwidth B, as discussed in Section V

$$\max_m |h_m| \cong 2B. \tag{17}$$

From (15)-(17)

$$|\Delta H(e^{j\omega})| \cong 2B\sqrt{\frac{N}{12}}\,2^{-L_0}. \tag{18}$$

Amplitude deviation is estimated by

$$|H(e^{j\omega}) + \Delta H(e^{j\omega})| \leqq |H(e^{j\omega})| + |\Delta H(e^{j\omega})|. \tag{19}$$

In this rounding off method, filter response deviation caused by finite word length is more significant in the stopband than in the passband.

Next, roundoff error in the latter method is analyzed. The first-order PDC-DF is discussed at first.

Let $\epsilon_k^{(1)}$ be the roundoff error in $\Delta_k^{(1)}$. It is distributed in the region $[-(\Delta c/2), (\Delta c/2)]$. When no permutations are required, the transfer function error caused by $\epsilon_k^{(1)}$ becomes

$$\Delta H_k(z) = \epsilon_k^{(1)} \sum_{m=k}^{N-1} \text{sign}\,(h_m)\, z^{-m}. \tag{20}$$

When $|\Delta H_k(e^{j\omega})|$ is assumed to be white, it can be expressed from the Parseval relation as

$$|\Delta H_k(e^{j\omega})|^2 = (\epsilon_k^{(1)})^2 \sum_{m=k}^{N-1} 1 = (N-k)\,(\epsilon_k^{(1)})^2. \tag{21}$$

If $|\Delta H_k(e^{j\omega})|$ is not assumed to be white, the error becomes larger than the estimation by (21). Since $\epsilon_k^{(1)}$ and $\epsilon_{k'}^{(1)}$ $(k \neq k')$ are independent from each other, the total transfer function error can be estimated based on its power

$$|\Delta H(e^{j\omega})|^2 = \sum_{k=0}^{N-1} |\Delta H_k(e^{j\omega})|^2. \tag{22}$$

From (21)

$$|\Delta H(e^{j\omega})|^2 = \sum_{k=0}^{N-1} (N-k)\,(\epsilon_k^{(1)})^2. \tag{23}$$

Equation (23) is rewritten as

$$|\Delta H(e^{j\omega})|^2 = \sum_{k=0}^{N-1} \sum_{i=0}^{k} (\epsilon_i^{(1)})^2. \tag{24}$$

Using the $\epsilon_k^{(1)}$ variance, that is $\Delta_c^2/12$, (24) becomes

$$|\Delta H(e^{j\omega})|^2 = \sum_{k=0}^{N-1} (k+1)\cdot\frac{\Delta_c^2}{12} \cong N^2\,\frac{\Delta_c^2}{24} \tag{25a}$$

and

$$|\Delta H(e^{j\omega})| \cong N\,\frac{\Delta_c}{\sqrt{24}}. \tag{25b}$$

Even if some permutations are required, the filter response deviation can be estimated by (25) under the condition that $|\Delta H_k(e^{j\omega})|$ is white. The second-order PDC-DF can be analyzed in the same way, and the following result is obtained

$$|\Delta H(e^{j\omega})| \cong N^2\,\frac{\Delta_c}{12}. \tag{26}$$

As a result, the former method can give less sensitivity. Therefore, the discussions in the following sections are carried out based on the former method.

B. Data Scaling and Output Roundoff Noise

It is well known that data must be scaled in order to prevent overflows in the multiplier input in two's-complement representation. The following analysis was carried out based on the second-order PDC-DF. The discussion for the first-order PDC-DF can be obtained in the same manner. The scaling factors are shown in Fig. 1. The internal scaling factor, S_2 is introduced to reduce the output roundoff noise.

Scaling Factor: The data have to be scaled at certain points to prevent overflows in the $\Delta_j^{*(2)}$ and the sign bit multiplier inputs; that is, $u_j^{*(2)}(n)$ and $x_k^*(n)$. S_1 is determined by the transfer function from the filter input to the $\Delta_j^{*(2)}$ multiplier input. It is highly dependent on the required permutations in the second and the third data mapping blocks, that is DM2 and DM3. The worst permutations for the output roundoff noise are employed in this paper. When no permutations are required in the first data mapping block DM1, $u_k^{(1)}(n)$ is calculated by

$$u_k^{(1)}(n) = S_1 \sum_{m=k}^{N-1} \text{sign}\,(h_m)\, x(n-m). \tag{27}$$

In the second data mapping block DM2, $u_k^{(1)}(n)$ $(k = 0, 1, \cdots, N_1 - 1)$ are assumed to be transformed to $u_l^*(n)$ $(l = N - N_1, N - N_1 + 1, \cdots, N-1)$. In this permutation, the maximum of $|u_l^{(2)}(n)|$ is bounded by

$$\begin{aligned}\max_l |u_l^{(2)}(n) &\leqq \sum_{k=0}^{N_1-1} |u_k^{(1)}(n)| \leqq \sum_{k=0}^{N_1-1} S_1 \sum_{m=k}^{N-1} |x(n-m)| \\ &= S_1\left\{\sum_{m=0}^{N_1-1} (m+1)\,|x(n-m)| \right.\\ &\quad \left. + \sum_{m=N_1}^{N-1} N_1\,|x(n-m)|\right\}.\end{aligned} \tag{28}$$

Using the maximum value of $|x(n)|$, (28) is rewritten as

$$\max_l |u_l^{(2)}(n)| \leqq \max_n |x(n)|\, S_1 \left\{\frac{N_1(N_1+1)}{2} + N_1(N-N_1)\right\}. \tag{29}$$

The scaling to guarantee against overflows in $u_l^{(2)}(n)$ by (29), is overly pessimistic. On the other hand, it can be assumed that no mutual correlation exists between the $x(n)$ samples. Therefore, the scaling factor can be based on estimates of the variance of $u_l^{(2)}(n)$. In other words, the data must be scaled so that the variance of $u_l^{(2)}(n)$ satisfies

$$\frac{\text{Var}\,[u_l^{(2)}(n)]}{\text{Var}\,[x(n)]} \leqq 1. \tag{30}$$

The variance of $u_l^{(2)}(n)$ is obtained by

$$\mathrm{Var}\,[u_l^{(2)}(n)] \leqq \mathrm{Var}\,[x(n)]\,(S_1^A)^2 \cdot \left\{\sum_{m=0}^{N_1-1} (m+1)^2 + N_1^2(N-N_1)\right\}$$

$$\cong \mathrm{Var}\,[x(n)]\,(S_1^A)^2(NN_1^2 - \tfrac{2}{3}N_1^3) \tag{31}$$

where the sign of equality is held for $l = N - N_1$. In DM3, $u_{N-N_1}^{(2)}(n)$ is assumed to be transformed to one of

$u_j^{*(2)}(n)\;(N - N_2 \leqq j \leqq N-1)$.

From (30) and (31), the input scaling factor S_1 is determined as

$$S_1 \cong (NN_1^2 - \tfrac{2}{3}N_1^3)^{-1/2}. \tag{32}$$

The assumptions used in deriving (32) are summarized once more in the following.

1) The worst permutations for the output roundoff noise are employed.
2) The $x(n)$ samples do not have a mutual correlation. That is, they are independent from each other.

In order to reduce the output roundoff noise, the output scaling factor S_3 has to be decreased. It can be accomplished by introducing the internal scaling factor S_2, which is included in $\Delta_j^{*(2)}$, as shown in Fig. 1. Furthermore, $\Delta_j^{*(2)} \cdot S_2$ and $(S_3)^{-1}$ must be less than or equal to unity to prevent overflows in the S_3 input. Considering the above conditions, S_2 and S_3 are determined as follows:

$$S_2 = \min\left\{(\max_l\,(\Delta_l^{(2)}))^{-1}, (S_1^A)^{-1}\right\} \tag{33}$$

$$S_3 = \max\left\{\max_l\,(\Delta_l^{(2)})\,(S_1^A)^{-1}, 1\right\} \tag{34}$$

Output Roundoff Noise: The main roundoff noise sources are the input scaling factor S_1 and the $\Delta_j^{*(2)}$ multiplier. The transfer functions from these noise sources to the filter output are $S_2S_3H(z)$ and S_3, respectively. The output roundoff noise becomes

$$N_{\mathrm{PDC}} = \frac{\Delta_d^2}{12}\left\{\frac{1}{2\pi}\int_{-\pi}^{\pi} (S_2S_3\,|H(e^{j\omega})|)^2\,d\omega + (S_3)^2N_2\right\} \tag{35}$$

where the roundoff error is distributed uniformly in the region $[-(\Delta d/2), (\Delta d/2)]$. Using the bandwidth B, the following relation is obtained

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} |H(e^{j\omega})|^2\,d\omega = 2B. \tag{36}$$

From (32)-(34) and (36), (35) is rewritten as

$$N_{\mathrm{PDC}} \cong \frac{\Delta_d^2}{12}\{2B + N_2(\max_l\,(\Delta_l^{(2)}))^2\}\,(NN_1^2 - \tfrac{2}{3}N_1^3),$$

$$(\max_l\,(\Delta_l^{(2)}))^{-1} \leqq (NN_1^2 - \tfrac{2}{3}N_1^3)^{1/2} \tag{37a}$$

or

$$N_{\mathrm{PDC}} \cong \frac{\Delta_d^2}{12}\{2B(NN_1^2 - \tfrac{2}{3}N_1^3) + N_2\}$$

$$(\max_l\,(\Delta_l^{(2)}))^{-1} \geqq (NN_1^2 - \tfrac{2}{3}N_1^3)^{1/2}. \tag{37b}$$

The above derived formula for N_{PDC} is based on the worst permutations. Therefore, it is somewhat reduced in the actual applications.

From the above result, the output roundoff noise is determined by the numbers of nonzero difference coefficients, the bandwidth and the $\max_l(\Delta_l^{(2)})$. These factors are highly dependent on filter characteristics. Their relations are discussed in Section V and numerical examples are illustrated in Section VI.

V. Computational Complexity

A. Computational Complexity

In the previous sections, the PDC-DF algorithm and the quantization error analysis are discussed. Computational complexity is discussed here, based on the above results. The computational complexity can be considered to determine hardware size and an execution time in actual realizations. Therefore, it is highly dependent on the number of operations and the internal data word lengths.

The number of additions required in the $u_k^{(1)}(n)$ and the $u_l^{(2)}(n)$ calculations given by (8) and (13′), are N and N_1, respectively. N_2 times multiplications are required to obtain the product of $u_l^{(2)}(n)$ and $\Delta_l^{(2)}$ given by (11′). The internal scaling factor can take the power of two, and is simply realized using digital devices. Therefore, they are not counted in the computational complexity.

Let L_{PDC} be the $\Delta_l^{(2)}$ word lengths, which are given by (38). L_0 are the h_m word lengths.

$$L_{\mathrm{PDC}} = L_0 - \log_2\,(\max_m |h_m| / \max_l\,(\Delta_l^{(2)})). \tag{38}$$

The computational complexity for the PDC-DF is defined as follows:

$$0_{\mathrm{PDC}} = W_{\mathrm{PDC}}(N + N_1 + L_{\mathrm{PDC}} \cdot N_2). \tag{39}$$

W_{PDC} are the internal data word lengths in the PDC-DF, and are expressed as

$$W_{\mathrm{PDC}} \cong W_0 + \log_4\left(\frac{N_{\mathrm{PDC}}}{\Delta_d^2/12}\right) \tag{40}$$

where the first term is the base word length to accomplish the desirable noise gain, and the second term indicates the output roundoff noise contribution. 0_{PDC} corresponds to the number of equivalent additions. The important feature of the PDC-DF is to reduce the computational complexity from the conventional method. In order to compare the PDC-DF computational complexity with that for a direct realization, the latter is briefly stated here. The computational complexity becomes

$$0_D = W_D(L_0 + 1)N \tag{41}$$

where W_D are the internal data word lengths expressed as

$$W_D = W_0 + \log_4 N. \tag{42}$$

B. Relation Between Computational Complexity and Filter Responses

The factors determining the computational complexity, which are W_{PDC}, N_1, N_2, and L_{PDC}, are mostly determined by filter responses, or, in other words, the original coefficient distribution function. Therefore, the relation between the filter response and the original coefficient distribution function is discussed at first. The results are given as a lemma and its proof is described in the Appendix.

Lemma: Let the maximum, the mean, the variance, and the standard variance of h_m be $\max_m |h_m|, E[h_m]$, Var $[h_m]$, and $SV[h_m]$. They can be expressed, using filter length N and the bandwidth B normalized by the sampling rate as follows:

$$\max_m |h_m| = 2B \tag{43}$$

$$E[h_m] = \frac{1}{N}: \text{ LPF} \tag{44a}$$

$$= 0: \text{ BPF} \tag{44b}$$

$$\text{Var } [h_m] = \frac{2B}{N} - \frac{1}{N^2} \cong \frac{2B}{N}: \text{ LPF} \tag{45a}$$

$$= \frac{2B}{N}: \text{ BPF} \tag{45b}$$

$$SV[h_m] = \{\text{Var } [h_m]\}^{1/2} \cong \sqrt{\frac{2B}{N}}. \tag{46}$$

From (38), L_{PDC} is determined by $\max_l (\Delta_l^{(2)})$, which is mainly determined by the difference between the large magnitude original coefficients, normalized by $\max_m |h_m|$, and their probability density. High probability density for the large magnitude original coefficients gives small $\max_l (\Delta_l^{(2)})$ and vice versa. The small $\max_l (\Delta_l^{(2)})$ can decrease L_{PDC}. On the other hand, small magnitude values $\Delta_k^{(1)}$ and $\Delta_l^{(2)}$ are mainly generated as the difference between the small magnitude original coefficients.

When the original coefficients are rounded off, small magnitude values $\Delta_k^{(1)}$ and $\Delta_l^{(2)}$ are likely to be zero. This indicates that the numbers of nonzero difference coefficients N_1 and N_2 are reduced from N. Hence, high probability density for the small magnitude original coefficients gives small N_1 and N_2. The probability density for the large and the small magnitude original coefficients can be evaluated by the ratio of the standard variance and the maximum value of $|h_m|$

$$D = \frac{SV[h_m]}{\max_m |h_m|} \cong \frac{1}{\sqrt{2BN}}. \tag{47}$$

The large magnitude D indicates high probability density for the large magnitude original coefficients, and vice versa. From (47), D is determined by two parameters B and N, which specify the filter response and the filter length. The relation between the computational complexity given by (39) and these parameters is discussed in detail in the following.

Narrow-band filters: Since B is small, in other words D becomes large, the probability density for the large magnitude original coefficients becomes high. It implies that L_{PDC} is small and N_1 and N_2 are not so reduced.

Wide-band filters: D becomes small, and the small magnitude original coefficients have high probability density. Therefore L_{PDC} is not small, while N_1 and N_2 are well reduced.

The output roundoff noise N_{PDC} is mainly determined by B, $\max_l (\Delta_l^{(2)})$, N_1 and N_2. Since B and $\max_l (\Delta_l^{(2)})$ are inversely proportional to N_1 and N_2, N_{PDC} is not so dependent on B. The number of operations is determined by L_{PDC}, N_1, and N_2. From the above discussion, L_{PDC} is inversely proportional to N_1 and N_2. Therefore, the number of operations is not dependent on B. The computational complexity 0_{PDC} is determined by both the number of operations and the roundoff noise. Combining both factors, it can be estimated that the computational complexity is not so dependent on the bandwidth, that is, filter responses.

Filter length: D, N_1, and N_2 are well reduced in high-order filters. L_{PDC} is, however, almost the same as that for low-order filters. The reason can be explained as follows: under the same probability density for the original coefficients, the high-order filters can give small value difference coefficients, compared with low-order filters. Considering the contributions of all factors, high-order filters can give a lower computational complexity than low-order filters.

The results of this paragraph can be illustrated in Fig. 2, using the truncated ideal impulse responses for low-pass filters with narrow and wide bands and a bandpass filter.

The lemma is based on the truncated ideal filter response [7]. The above mentioned properties are, however, approximately held for the other FIR filters designed through the Remez-exchange method [8], the linear programming approach [9], and the minimum phase design method [10].

Other filters besides frequency selective filters, for example an all-pass function, are not discussed here. One example is illustrated in Section VI.

VI. Numerical Example

Numerical examples to show the coefficient distribution function and the computational complexity for several kinds of filter responses are illustrated. Comparison between the PDC-DF and a direct realization is also discussed here. The FIR filters used in the following discussions are as follows: 99th- and 299th-order low-pass filters and bandpass filters designed through the Remez-exchange method, and a 252nd all-pass function, whose group delay response is linear. The bandwidth B is determined by a -6 dB gain point.

Coefficient Distribution Function: Table II shows probability density functions for $|h_m|$, $\Delta_k^{(1)}$ and $\Delta_l^{(2)}$. The vertical and the horizontal axes imply the probability density of the coefficients in percentage and the coefficient value normalized by $\max_m |h_m|$, respectively. L_0 are taken as 10 bits for the 99th filters and the all-pass function. 12 bits are taken for the 299th filters. From this table, the analytical results obtained in Section V can be confirmed. L_{PDC}, N_1, and N_2 are shown

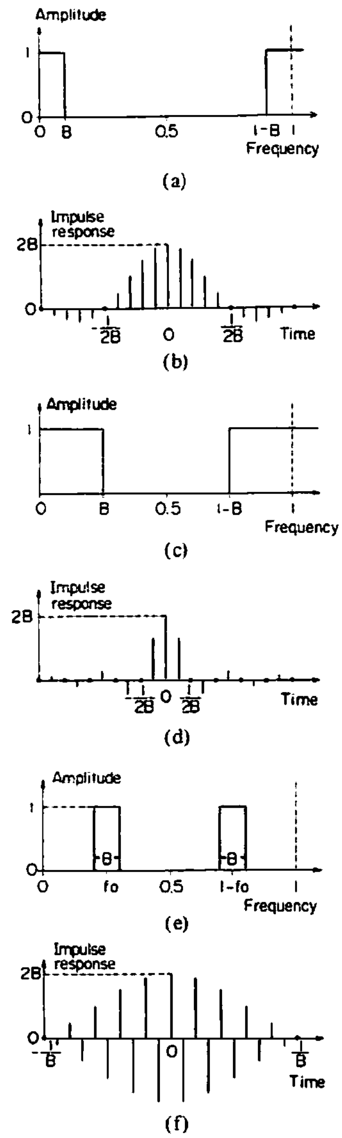


Fig. 2. Truncated ideal filter responses. (a) and (b) Narrow-band LPF (B = 0.1 Hz). (c) and (d) Wide-band LPF (B = 0.25 Hz). (e) and (f) Narrow-band BPF (B = 0.1 Hz). Sampling rate and f_0 are taken as 1 Hz and 0.25 Hz, respectively.

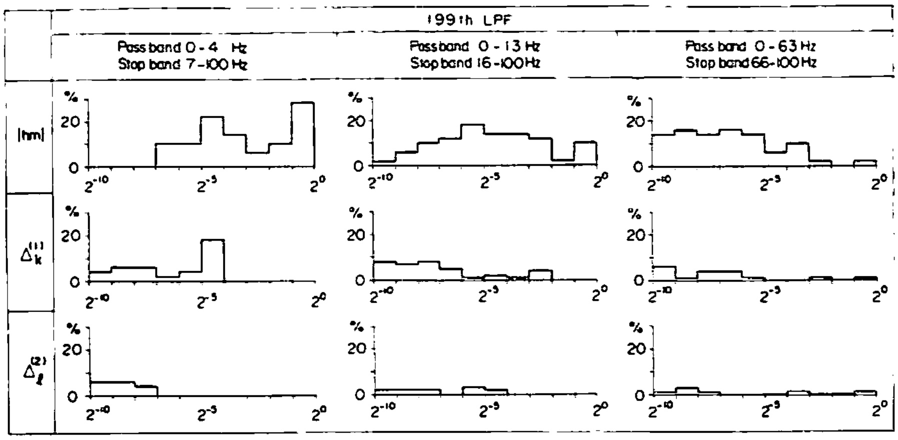
in Fig. 3. N_1 and N_2 are normalized by N. Since, L_{PDC} are not normalized, there is about a 2 bit difference between the 99th and the 299th filters. This means that reductions in L_{PDC}, in both the 99th and the 299th filters, are almost the same. From this result, the L_{PDC} independency from filter length can also be recognized. The obtained relation between the computational complexity factors L_{PDC}, N_1, and N_2, and the bandwidth B and the filter length N is also confirmed from this figure.

Output Roundoff Noise: Fig. 4 shows the output roundoff noises given by (37) in dB, that is, 10 log $(N_{PDC}/(\Delta_d^2/12))$, and 10 log (N) in a direct form. From this figure, the output roundoff noise is increased, compared with the direct realization. Therefore, the excess data word lengths are required to maintain the same roundoff noise. For example, they are around 4 and 6 bits for the 99th and the 299th filters, respectively. Since (37) is based on the worst permutations for the output roundoff noise, as discussed in Section IV, it can be somewhat decreased in the actual applications. These excess data word lengths are included in the computational complexity 0_{PDC}, as discussed in Section V.

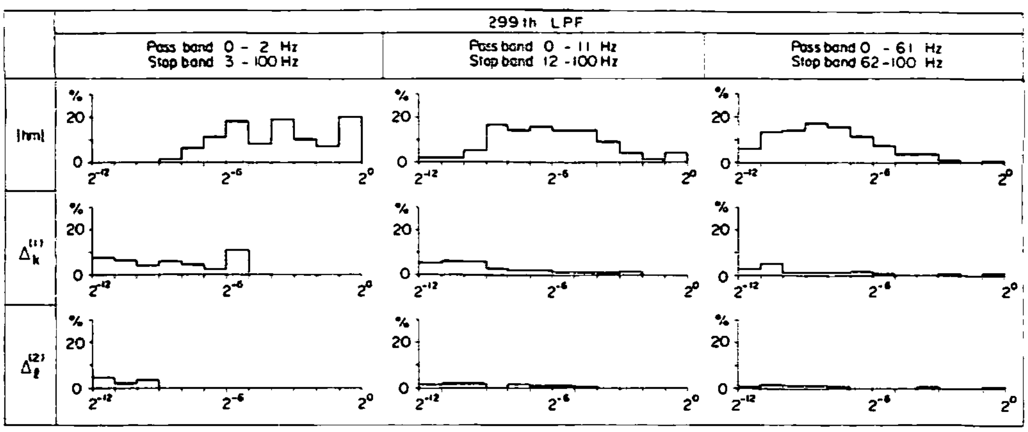
Computational Complexity: The computational complexity, evaluated by (39) is illustrated in Fig. 5, where 0_{PDC} is normalized by 0_D, and W_D are taken as 16 bits. The excess data word lengths are taken as 4 bits for the 99th filters and 6 bits for the 299th filters, from the output roundoff noise shown in Fig. 4. They can be considered as the maximum excess data word lengths. The normalized computational complexities become around 23 percent and 18 percent for the 99th and the 299th filters, respectively. The 0_{PDC} dependency on the band-

TABLE II

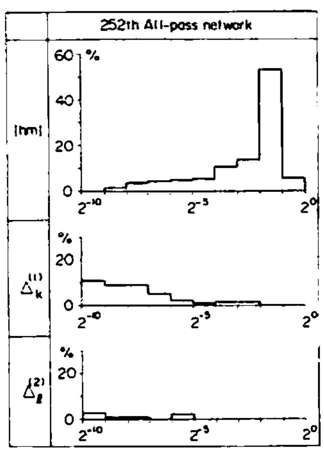
PROBABILITY DENSITY FUNCTIONS OF $|h_m|$, $\Delta_k^{(1)}$ AND $\Delta_l^{(2)}$. VERTICAL AND HORIZONTAL AXES INDICATE PROBABILITY DENSITY IN PERCENT AND COEFFICIENT VALUE NORMALIZED BY $\max_m |h_m|$. L_0 ARE TAKEN AS 10 BITS FOR (a) AND (c), AND 12 BITS FOR (b).



(a)



(b)



(c)

width B is small as discussed in Section V. The high-order filters can give small computational complexity. Table III shows the numerical example for the all-pass function. The computational complexity reduction is about 80 percent, compared with the direct realization.

The PDC-DF can be effectively applied to minimum phase FIR filters in the same manner, because its coefficient distribution function is similar to that for linear phase FIR filters. Consequently, it can be concluded that the proposed approach is very useful for a wide-range filter response.

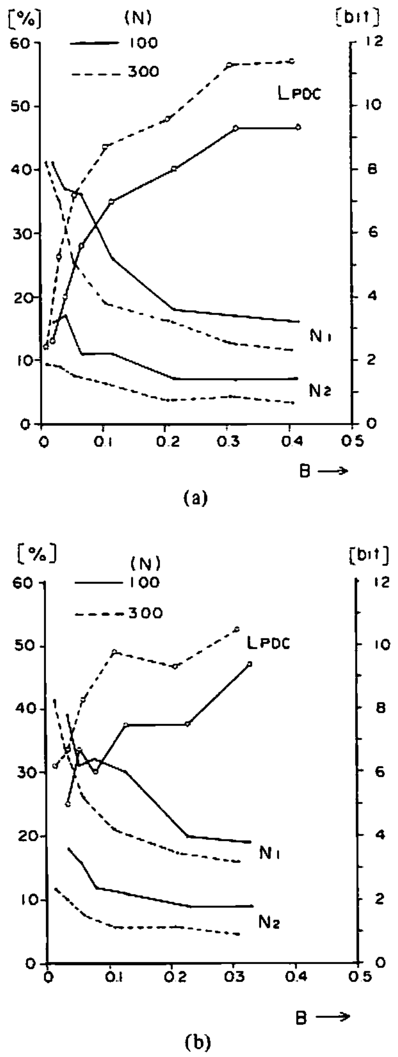


Fig. 3. $\Delta_l^{(2)}$ word lengths L_{PDC} and number of nonzero difference coefficients N_1 and N_2. N_1 and N_2 are normalized by N. (a) LPF. (b) BPF.

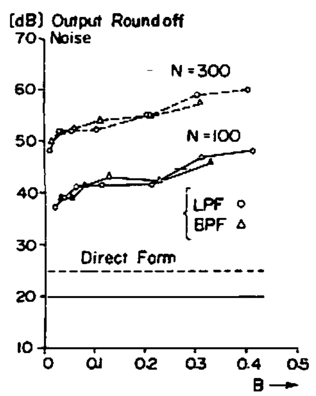


Fig. 4. Output roundoff noise evaluated by 10 log $(N_{PDC}/(\Delta_d^2/12))$ and 10 log N in dB. Worst permutations for output roundoff noise are employed.

VII. Conclusion

FIR filters give desirable features, such as a linear phase response and stable implementation, while high Q filters require very high computational complexity in a direct realization. In this paper, a new difference coefficient digital filter is proposed for such high-order filters. Its coefficients are formed as the difference between the successive values of the original coefficients reordered in a sequence with falling magnitude. It is effectively applied to a wide-range filter response. The quantization error analysis is discussed. Slightly large data word lengths are required in the new approach to maintain the same roundoff noise as in a direct realization. The computational complexities of the proposed realization become around 23 percent and 18 percent for the 99th and the 299th filters, respectively, compared with the conventional method.

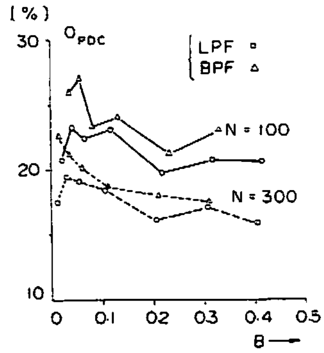


Fig. 5. Computational complexity O_{PDC} normalized by O_D, where W_D are taken as 16 bits.

TABLE III
NUMERICAL EXAMPLE FOR 252ND ALL-PASS FUNCTION, WHOSE GROUP DELAY IS LINEAR

252nd All-Pass Function	
N	253
$\max_m \lvert h_m \rvert$	0.179
N_1	37 %
N_2	5 %
$\max_l(\Delta_l^{(2)})$	5.22×10^{-3}
L_{PDC}	4.9 bits
N_{PDC}	65.3 dB
O_{PDC}	20.4 %

Appendix
Proof of Lemma

Proof is based on the truncated ideal filter response, using a rectangular time window. This assumption does not lose generality for high-order FIR filters.

1) LPF: An ideal impulse response $h_i(n)$ is expressed as

$$h_i(n) = 2B \frac{\sin(2\pi Bn)}{2\pi Bn}. \tag{A1}$$

FIR filter coefficients h_m, with filter length N, are obtained by windowing $h_i(n)$, as shown in Fig. 2(b). Let $W(n)$ be a time window,

$$h_m = W\left(m - \frac{N-1}{2}\right) h_i\left(m - \frac{N-1}{2}\right). \tag{A2}$$

Since

$$W(0) = 1 \tag{A3}$$

then

$$\max_m |h_m| = h_{(N-1/2)} = 2B \tag{A4}$$

where N is considered to be odd. When N is even, the $\max_m |h_m|$ is approximately $2B$ for linear phase FIR filters. Let $H(z)$ be the transfer function with the impulse response h_m

$$H(z) = \sum_{m=0}^{N-1} h_m z^{-m} \tag{A5}$$

where

$$z = e^{j\omega T}, \quad T = 1. \tag{A6}$$

The mean value of h_m can be expressed as follows:

$$E[h_m] = \frac{1}{N} \sum_{m=0}^{N-1} h_m = \frac{1}{N} H(e^{j0}). \tag{A7}$$

Since

$$H(e^{j0}) = 1 \tag{A8}$$

in LPF

$$E[h_m] = \frac{1}{N}. \tag{A9}$$

The variance of h_m can be expressed using the means of h_m and h_m^2

$$\begin{aligned} \text{Var}\,[h_m] &= E[(h_m - E[h_m])^2] \\ &= E[h_m^2] - E^2[h_m]. \end{aligned} \tag{A10}$$

Furthermore, from the Parseval relation,

$$E[h_m^2] = \frac{1}{N} \sum_{m=0}^{N-1} h_m^2 = \frac{1}{2\pi N} \int_{-\pi}^{\pi} |H(e^{j\omega})|^2 \, d\omega. \tag{A11}$$

Since

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} |H(e^{j\omega})|^2 \, d\omega = 2B \tag{A12}$$

then

$$E[h_m^2] = \frac{2B}{N}. \tag{A13}$$

From (A9) and (A13)

$$\text{Var}\,[h_m] = \frac{2B}{N} - \frac{1}{N^2}. \tag{A14}$$

2) BPF: The impulse response for the BPF shown in Fig. 2(f) given by

$$h_m = W\left(m - \frac{N-1}{2}\right) h_i^*\left(m - \frac{N-1}{2}\right) \cdot 2\cos\left(2\pi f_0 \left(m - \frac{N-1}{2}\right)\right) \tag{A15}$$

where

$$h_i^*(n) = B \frac{\sin(\pi B n)}{\pi B n}. \tag{A16}$$

Hence

$$\max_m |h_m| = h_{(N-1/2)} = 2B. \tag{A17}$$

The mean value for h_m becomes

$$\frac{1}{N} \sum_{n=0}^{N-1} h_n = \frac{1}{N} H(e^{j0}) = 0. \tag{A18}$$

The variance of h_m is

$$\begin{aligned} \text{Var}\,[h_m] &= E[h_m^2] - E^2[h_m] \\ &= \frac{2B}{N} \end{aligned} \tag{A19}$$

Q.E.D.

References

[1] C. S. Burrus, "Digital filter structures described by distributed arithmetic," *IEEE Trans. Circuits Syst.*, vol. CAS-24, pp. 674-680, Dec. 1977.

[2] A. Peled and B. Liu, "A new hardware realization of digital filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-22, pp. 456-462, Dec. 1974.

[3] K. Nakayama and T. Mizukami, "A new realization of sum of products and inverse sum of products by partitioned arithmetics," *European Conf. Circuit Theory, Design*, Sept. 1980, pp. 533-538.

[4] W. K. Jenkins and B. J. Leon, "The use of residue number systems in the design of finite impulse response digital filters," *IEEE Trans. Circuits Syst.*, vol. CAS-24, pp. 191-201, Apr. 1977.

[5] P. J. Van Gerwen *et al.*, "A new type of digital filter for data transmission," *IEEE Trans. Commun.*, vol. COM-23, pp. 222-234, Feb. 1975.

[6] D. S. K. Chan and L. R. Rabiner, "Analysis of quantization errors in the direct form for finite impulse response digital filters," *IEEE Trans. Audio Electroacoust.*, vol. AU-21, pp. 354-366, Aug. 1973.

[7] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975.

[8] T. W. Parks and J. H. McClellan, "Chebyshev approximation for nonrecursive digital filters with linear phase," *IEEE Trans. Circuit Theory*, vol. CT-19, pp. 189-194, Mar. 1972.

[9] L. R. Rabiner, "The design of finite impulse response digital filters using linear programming techniques," *Bell Syst. Tech. J.*, vol. 51, pp. 1177-1198, July-Aug. 1972.

[10] O. Herrmann and W. Schuessler, "Design of nonrecursive digital filters with minimum phase," *Electron. Lett.*, vol. 6, pp. 328-329, May 1970.

Kenji Nakayama was born in Mie Prefecture, Japan, on February 24, 1947. He received the B.E. degree in electronics engineering from the Tokyo Institute of Technology, Tokyo, Japan, in 1971.

From 1971 to 1972 he was engaged in research at the T.I.T. Since he joined the Transmission Division, Nippon Electronic Company, Ltd. in 1972, he has worked on the research and development of *LC*, digital and switched capacitor filters. His current research interests are in the areas of digital signal processing and network theory.

Mr. Nakayama is a member of the Institute of Electronics and Communication Engineers of Japan.