A Study on Improvements in Computational Efficiency of a Filter Design

March 1988

Kuan Tung

A Study on Improvements in Computational Efficiency of a Filter Design

by

Koichi Ichige, B.E., M.E.

DISSERTATION

Presented to

Doctoral Program in Engineering,

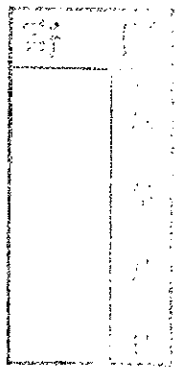
University of Tsukuba

In Partial Fulfillment

of the Requirement

for the Degree of

Doctor of Engineering



University of Tsukuba

March 1999

A Study on Improvements in Computational Efficiency of a Filter Design

The dissertation of Koichi Ichige is approved by supervisory committee:

M. Natori

Makoto Natori, Committee chairman

Rokuya Ishii

Rokuya Ishii (Yokohama National Univ.)

Yukio Ishibashi

Yukio Ishibashi

Kazuhiro Hirasawa

Kazuhiro Hirasawa

Takashi Kitagawa

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Signal processing technology was first developed as analog techniques, and digital techniques were later developed in 1960-70s. Recently digital techniques often take part in analog techniques due to the accurate performance of digital signals. Figure 1.1 shows a model of analog signal processing and of digital signal processing. In the model of digital

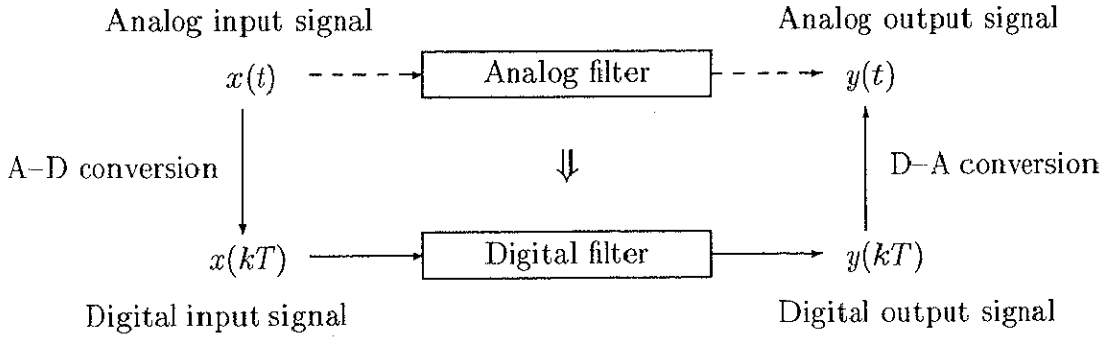


Figure 1.1: A model of analog signal processing (broken vectors) and of digital signal processing (solid vectors).

signal processing, what is important is Analog to Digital (A–D) conversion, the inverse Digital to Analog (D–A) conversion, and digital filter design. A–D/D–A conversions can be regarded as '*Transformation of signals*' from analog signals into digital ones, and digital filter design can also be regarded as '*Transformation of systems*' from analog filters into digital filters. These two transformations are considered to be significant.

It is important to make A–D/D–A conversions rapidly and with high accuracy. Recently Digital Signal Processors (DSPs), which construct digital devices, can work very rapidly and with high accuracy and A–D/D–A converters must follow the developed digital devises.

Digital filters can be divided into two types from a view point of impulse response. One type is the Infinite Impulse Response (IIR) digital filters and the other type is the Finite Impulse Response (FIR) digital filters. They have opposite advantages and disadvantages, and whichever is more appropriate for a given problem is chosen for real use. The research objects for digital filters are in the design, in the construction and in the implementation. Here the design stage is studied and is considered to be much improved.

IIR digital filter design methods are generally divided into two categories: transformations from analog filters and optimizations of the desired characteristics. Bilinear and frequency transformations are representative design methods of the former. Such transformations can employ the highly developed design techniques of analog filters, but they requires complicated hand computations. These computations become more complicated for long filters which realize minute characteristics. The latter design methods enable more flexible design, but so-called 'the best design algorithm' has not been established yet. Further, the designed filter is optimized by any means, so the analysis of the designed filter becomes difficult.

FIR digital filter design methods are also divided into two categories: classical design methods like window methods and frequency sampling, and optimization methods based on the minimax approximation criterion. In these methods, the Parks-McClellan optimization algorithm is most widely used because of its flexible and efficient performance.

This algorithm is based on Remez exchange method which was originally proposed to solve Chebyshev approximation problem, and requires the filter length in advance of design. Thus the FIR filter design problem is considered to be finding a filter with a minimum filter length which satisfy given specifications. Since a shorter filter length is better for computation and implementation, many filters are designed to see how long the minimum filter length is. It takes a great deal of computational cost and time.

In this dissertation, various computations in A–D/D–A conversions and in designing digital filters are studied, which make up the most significant role of digital signal processing. More specifically, this dissertation aims at improving computational efficiency of (i) A–D/D–A conversions based on wavelet transform, (ii) design of IIR digital filters based on frequency transformation, and (iii) design of FIR digital filters based on the minimax criterion using the Parks-McClellan algorithm.

In Chapter 2, a simple A–D/D–A conversion scheme based on wavelet transform is proposed. In wavelet theory, a given signal is first approximated by a linear combination of shifted scaling functions in the sense of least square. Then the coefficients for the scaling functions are transformed into wavelet coefficients by the two-scale relation, and vice versa. In the end, the linear combination of the scaling functions gives a reconstruction of the original signal.

Here we should note (as emphasized in [22]) that the two-scale relation operates not on sampled values of a signal but on the scaling function coefficients. The mutual conversion between a signal and its scaling function coefficients has been mathematically formulated [22] by means of inner products and weighted sums in the continuous-time domain, but there has been little consideration of its implementation.

We shall look at the case of B-spline scaling functions [37] for example, which are one of the most popular scaling functions in wavelet theory. It is just possible to implement reconstruction of a signal from coefficients for the B-spline scaling functions (abbreviated as B-spline coefficients hereafter) as a linear combination of the shifted B-splines.

Generally a practical remedy for such a situation is an oversampling discrete-time implementation. By replacing the B-splines by their sampled values (called discrete B-splines [38]), we may have a good discrete-time implementation if the oversampling ratio is high. Even faster computation is possible if we use multifold Recursive Running Sum (RRS) functions [2, 4, 14], which can be regarded as another discrete approximation for the B-splines, instead of discrete B-splines.

Focusing on the case of B-spline scaling functions and the use of multifold RRS functions, this chapter presents a scheme of discrete-time implementation for conversion from a continuous-time signal into its B-spline coefficients (B-spline decomposition) and the inverse operation (reconstruction).

In Chapter 3, an automatic design procedure for IIR digital filters based on bilinear and frequency transformations is proposed. Furthermore, its fast algorithms are studied.

Frequency transformations derive filters of various types from a filter of low-pass type. The transformation formulae are well-known and widely used. But the direct application of these transformations yields a complicated formula of the target transfer function, which has to be reduced by hand computation into the form of a rational polynomial. The hand computation of those transformations could be replaced by an automatic procedure if the relation between the coefficients of the transfer functions is formulated.

The purpose of this chapter is to derive the explicit formulae which connect those

coefficients by matrices. The matrices also enable an automatic design of IIR digital filters from an analog low-pass filter.

The derived automatic design procedures still require a large number of operations, which include a lot of binomial coefficients. This problem should be solved if the elements of the matrices are easily computed. Here, some properties of the matrices for IIR digital filters are studied and fast algorithms to compute the matrices are also established and evaluated.

Chapter 4 presents an accurate estimation formula for minimum filter length of optimum FIR digital filters. For designing optimum (the minimax criterion based) linear-phase FIR digital filters, an iterative optimizing algorithm using Remez exchange method [34] has been established for FIRs with odd filter length [28], and also for those with even filter length [31]. This algorithm is the most widely used for the design of linear-phase FIRs because of its flexible and efficient performance. However the algorithm requires the filter length of designed filter to be known in advance, and then optimizes the amplitude characteristics in the minimax sense for a specified filter length.

In many practical cases, specifications of a target filter is given first, and then many filters are designed so as to see how long the minimum filter length must be. It is hard to know the exact value of the minimum filter length N which satisfy given specifications.

To conjecture an appropriate filter length from given specifications in advance, two estimation formulae have been proposed by Herrmann *et al.* [24, 29] and by Kaiser [27] for the design of optimum FIR low-pass filters. However these formulae cannot achieve enough accuracy because of lack of some considerations.

The objective of this chapter is to estimate the minimum filter length of optimum FIR

low-pass filters more accurately. The proposed estimation formula is mainly based on the observations from our experimental results. The accuracy of the proposed formula is evaluated by some design examples and quantitative distances in comparison with those of the conventional formulae.

Finally, some concluding remarks are given in Chapter 5.

Chapter 2

A Scheme of B-spline Decomposition and Reconstruction of Continuous-time Signals

2.1 Introduction

Wavelet theory provides a modern technique for sub-band coding of signals. In its application to continuous-time signals, a given signal is first approximated by a linear combination of shifted scaling functions in the sense of least square. Then the coefficients for the scaling functions are transformed into wavelet coefficients by the two-scale relation, and vice versa. Also, the linear combination of the scaling functions gives a reconstruction of the original signal.

Here we should note (as emphasized in [22]) that the two-scale relation operates not on sampled values of a signal but on the scaling function coefficients. The mutual conversion between a signal and its scaling function coefficients has been mathematically formulated [22] by means of inner products and weighted sums in the continuous-time domain. However there has been little consideration of its implementation.

We shall look at the case of B-spline scaling functions [37] for example, which are one of the most popular scaling functions in the wavelet theory. It is just possible to

implement reconstruction of a signal from coefficients for the B-spline scaling functions (abbreviated as B-spline coefficients hereafter) as a linear combination of the shifted B-splines. The linear combination can be constructed as a weighted sum of shifted B-splines generated by a network of integrators [1, 10, 11] in continuous-time. However this analog circuit requires a great deal of hardware complexity since it includes many integrators and D-A converter chips. The situation is worse when converting a given signal into B-spline coefficients. The coefficients are mathematically represented by the inner products of the signal and the shifted duals of the B-spline. Its circuit implementation must get more complicated since measurement of the inner products needs more integrators, multipliers, and A-D converter chips. Besides, analog implementation is prone to internal noise. For these reasons, analog implementations will not be considered further.

Generally a practical remedy for such a situation is an oversampling discrete-time implementation. By replacing the B-splines by discrete B-splines [38], we may have a good discrete-time implementation if the oversampling ratio is as high as hundreds or thousands. At such a high oversampling ratio, the fast scheme [38] of convolving the discrete B-splines with a discrete-time signal would be helpful for reducing the computational complexity. Even faster computation is possible if we use multifold Recursive Running Sum (RRS) functions [2, 4, 14], which can be regarded as another discrete approximation for the B-splines, instead of discrete B-splines. In fact, use of multifold RRS functions roughly halves the necessary additions and subtractions, and eliminates multiplications [2, 4, 14] as discussed in subsection 2.3.2. Moreover, in the cases that the oversampling ratio is ten or more, the multifold RRS functions get as close to the B-splines as to the discrete B-splines in the sense of mean square error [2, 4, 14] as shown in subsection 2.3.3.

Focusing on the case of B-spline scaling functions and the use of multifold RRS functions, the present dissertation studies a scheme of discrete-time implementation for conversion from a continuous-time signal into its B-spline coefficients (B-spline decomposition) and the inverse operation (reconstruction). The rest of this chapter is organized as follows: Following mathematical preparation in Sections 2.2 and 2.3, we substitute the multifold RRS functions for the B-splines to have an approximate scheme of B-spline decomposition and reconstruction in Section 2.4. Section 2.5 is the main contribution of this paper devoted to analyzing the stability and precision of the scheme. We do not use the mean-square norms that were used in [2, 4, 14] but use the supremum norms since we have to consider the maximum or worst value of signals to see stability and precision with this particular circuit. The analysis results in conditions for the circuit parameters to assure stability and required precision. We can design a stable circuit that meets a required precision in accordance with those conditions. Section 2.6 gives a design example and its numerical simulations.

2.2 Preliminaries

We recall mathematical formulae that will be referred to in the following sections.

Let $\boldsymbol{N}$, $\boldsymbol{R}$ and $\boldsymbol{Z}$ denote the set of all the natural numbers, that of all the real numbers, that of all the integers, respectively. Norms used in this paper are denoted as follows:

$$\|f\|_{L^1} := \int_{-\infty}^{\infty} |f(t)|dt, \quad \|f\|_{L^\infty} := \sup_{t\in\boldsymbol{R}} |f(t)|, \quad \|\{c_k\}_{k\in\boldsymbol{Z}}\|_{\ell^\infty} := \sup_{k\in\boldsymbol{Z}} |c_k|.$$

2.2.1 B-splines and Their Duals

B-spline β^m of order m is defined [37] as the $(m-1)$–fold convolution integral of the rectangular function β^1, *i.e.*,

$$\beta^m(t) := (\underbrace{\beta^1 \circledast \beta^1 \circledast \cdots \circledast \beta^1}_{m})(t),$$

where

$$\beta^1(t) := \begin{cases} 1, & 0 \le t < 1, \\ 0, & \text{elsewhere}, \end{cases}$$
$$(f \circledast g)(t) := \int_{-\infty}^{\infty} f(\tau) g(t-\tau) d\tau.$$

It can be expressed [37] in an explicit form as

$$\beta^m(t) = m \sum_{i=0}^{m} \frac{(-1)^i}{(m-i)!i!} (t-i)_+^{m-1}, \tag{2.1}$$

where $(a)_+$ denotes the truncated power function: $(a)_+ := \max\{a, 0\}$. the followings has been proven [37] that

$$\beta^m(t) = 0, \quad t < 0 \text{ or } t \ge m. \tag{2.2}$$
$$\sum_{k=-\infty}^{\infty} \beta^m(t-k) = 1. \tag{2.3}$$

The dual $\tilde{\beta}^m$ of B-spline is defined as a linear combination of shifted B-splines that satisfies

$$\int_{-\infty}^{\infty} \beta^m(\tau - k) \tilde{\beta}^m(t - \tau - \ell) d\tau = \delta_{k,\ell}, \quad k, \ell = 0, \pm 1, \pm 2, \ldots, \tag{2.4}$$

where

$$\delta_{k,\ell} = \begin{cases} 1, & k = \ell, \\ 0, & k \ne \ell. \end{cases}$$

This (2.4) has been derived from [22] to produce

$$\bar{\beta}^m(t) = \sum_{\ell=-\infty}^{\infty} g_\ell \beta^m(t-\ell), \tag{2.5}$$

where

$$g_\ell = \int_{-1/2}^{1/2} e^{j2\pi f\ell} \Big/ \sum_{p=-\infty}^{\infty} \left\{ \frac{\sin\pi(f-p)}{\pi(f-p)} \right\}^{2m} df. \tag{2.6}$$

Since the dual coefficients $\{g_\ell\}_{\ell\in\boldsymbol{Z}}$ of (2.6) decay exponentially as $|\ell|$ gets large [37], we can truncate the coefficients to have an approximation

$$\hat{\beta}^m(t) := \sum_{\ell=-M}^{M} \hat{g}_\ell \beta^m(t-\ell),$$

for $\bar{\beta}$, where $M \in \boldsymbol{N}$ and

$$\hat{g}_\ell := \begin{cases} g_\ell, & \ell = 0, \pm 1, \ldots, \pm M \\ 0, & \text{otherwise.} \end{cases} \tag{2.7}$$

2.3 Multifold RRS Functions and Their Properties

2.3.1 Definition and Expression

As a simple discrete version of B-splines, the multifold RRS function would be defined as follows.

Definition 2.1 Multifold RRS function $y_N^m(k)$ of order m has been defined as the $(m-1)$-fold discrete convolution of the sampled rectangular function

$$b_N^1(k) := \begin{cases} 1, & k = 0, 1, \ldots, N-1, \\ 0, & \text{elsewhere,} \end{cases}$$

divided by a scaling factor N^{m-1} $(N \geq 2)$, *i.e.*,

$$y_N^m(k) := \frac{1}{N^{m-1}} (\underbrace{b_N^1 * b_N^1 * \cdots * b_N^1}_{m})(k), \tag{2.8}$$

where

$$(a * b)(k) := \sum_{\ell=-\infty}^{\infty} a(\ell) b(k-\ell).$$

■

This definition (2.8) coincides the representation

$$b_N^m(k) = \frac{1}{N^{m-1}} \{ (\underbrace{b_N^1 * b_N^1 * \cdots * b_N^1}_{m}) * b_1^m \}(k), \tag{2.9}$$

of the discrete B–splines $b_N^m(k)$ [38] of order m defined as the sampled values of the B–spline $\beta^m(t)$, *i.e.*,

$$b_N^m(k) := \beta^m \left(\frac{k}{N} \right), \quad k = 0, \pm 1, \pm 2, \cdots,$$

if we omit the last factor of (2.9).

Proposition 2.2 The RRS function $y_N^m(k)$ can be expressed in an explicit form as

$$y_N^m(k) = \begin{cases} b_N^1(k), & m = 1, \\ \dfrac{m}{N^{m-1}} \displaystyle\sum_{i=0}^{m} \frac{(-1)^i}{(m-i)!i!} \prod_{j=1}^{m-1} (k - Ni + j)_+, & m = 2, 3, \ldots. \end{cases} \tag{2.10}$$

Proof Apparently (2.10) is true in the case $m = 1$. In the case $m = 2$, we have

$$\begin{aligned} y_N^2(k) &= \frac{1}{N} (b_N^1 * b_N^1)(k) \\ &= \frac{1}{N} \sum_{\ell=-\infty}^{\infty} b_N^1(\ell) b_N^1(k-\ell) \\ &= \frac{1}{N} \sum_{\ell=0}^{N-1} b_N^1(k-\ell) \\ &= \frac{1}{N} \sum_{\ell=k-N+1}^{k} b_N^1(\ell) \\ &= \begin{cases} (k+1)/N, & k = 0, 1, \cdots, N-1, \\ (2N-k-1)/N, & k = N, N+1, \cdots, 2N-1, \\ 0, & \text{otherwise} \end{cases} \end{aligned}$$

$$= \frac{1}{N}\{(k+1)_+ - 2(k-N+1)_+ + (k-2N+1)_+\}$$
$$= \frac{2}{N}\sum_{i=0}^{2}\frac{(-1)^i}{(2-i)!i!}(k-Ni+1)_+, \tag{2.11}$$

which means that (2.10) holds good for $m = 2$. Assume that (2.10) holds good for $m \geq 2$, *i.e.*,

$$y_N^m(k) = \frac{m}{N^{m-1}}\sum_{i=0}^{m}\frac{(-1)^i}{(m-i)!i!}\prod_{j=1}^{m-1}(k-Ni+j)_+. \tag{2.12}$$

Then we have

$$\begin{aligned}
&y_N^{m+1}(k)\\
&= \frac{1}{N^m}(\underbrace{b_N^1 * b_N^1 * \cdots * b_N^1}_{m+1})(k)\\
&= \frac{N^{m-1}}{N^m}(b_N^1 * y_N^m)(k)\\
&= \frac{1}{N}\sum_{\ell=-\infty}^{\infty} b_N^1(\ell)y_N^m(k-\ell)\\
&= \frac{1}{N}\sum_{\ell=0}^{N-1} y_N^m(k-\ell)\\
&= \frac{1}{N}\sum_{\ell=k-N+1}^{k} y_N^m(\ell)\\
&= \frac{1}{N}\sum_{\ell=k-N+1}^{k}\frac{m}{N^{m-1}}\sum_{i=0}^{m}\frac{(-1)^i}{(m-i)!i!}\prod_{j=1}^{m-1}(\ell-Ni+j)_+\\
&= \frac{m}{N^m}\sum_{i=0}^{m}\frac{(-1)^i}{(m-i)!i!}\sum_{\ell=k-N+1}^{k}\prod_{j=1}^{m-1}(\ell-Ni+j)_+\\
&= \frac{m}{N^m}\sum_{i=0}^{m}\frac{(-1)^i}{(m-i)!i!}\sum_{\ell=k-N+1}^{k}\frac{1}{m}\left\{\prod_{j=1}^{m}(\ell-Ni+j)_+ - \prod_{j=1}^{m}(\ell-1-Ni+j)_+\right\}\\
&= \frac{1}{N^m}\sum_{i=0}^{m}\frac{(-1)^i}{(m-i)!i!}\left\{\prod_{j=1}^{m}(k-Ni+j)_+ - \prod_{j=1}^{m}(k-N(i+1)+j)_+\right\}\\
&= \frac{1}{N^m}\left\{\frac{1}{m!0!}\prod_{j=1}^{m}(k+j)_+ + \sum_{i=1}^{m}\frac{(-1)^i}{(m-i)!i!}\prod_{j=1}^{m}(k-Ni+j)_+\right.\\
&\qquad \left. - \sum_{i=0}^{m-1}\frac{(-1)^i}{(m-i)!i!}\prod_{j=1}^{m}(k-N(i+1)+j)_+ - \frac{(-1)^m}{0!m!}\prod_{j=1}^{m}(k-N(m+1)+j)_+\right\}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{N^m}\left\{ \frac{m+1}{(m+1)!0!}\prod_{j=1}^{m}(k+j)_+ \right. \\
&\quad + \sum_{i=1}^{m}\left\{ \frac{(-1)^i}{(m-i)!i!} - \frac{(-1)^{i-1}}{(m-i+1)!(i-1)!} \right\}\prod_{j=1}^{m}(k-Ni+j)_+ \\
&\quad \left. + \frac{(-1)^{m+1}(m+1)}{0!(m+1)!}\prod_{j=1}^{m}(k-N(m+1)+j)_+ \right\} \\
&= \frac{1}{N^m}\left\{ \frac{m+1}{(m+1)!0!}\prod_{j=1}^{m}(k+j)_+ + \sum_{i=1}^{m}\frac{(-1)^i(m+1)}{(m+1-i)!i!}\prod_{j=1}^{m}(k-Ni+j)_+ \right. \\
&\quad \left. + \frac{(-1)^{m+1}(m+1)}{0!(m+1)!}\prod_{j=1}^{m}(k-N(m+1)+j)_+ \right\} \\
&= \frac{m+1}{N^m}\sum_{i=0}^{m+1}\frac{(-1)^i}{(m+1-i)!i!}\prod_{j=1}^{m}(k-Ni+j)_+, \qquad (2.13)
\end{aligned}
$$

which means that (2.10) also works with m replaced by $m+1$. The above (2.11)–(2.13) and the mathematical induction completes a proof of Proposition 2.2. ■

This function is named after the RRS (Recursive Running Sum) digital filter [19] in which a cascade performs multifold discrete convolution of the sampled rectangular function $b_N^1(k)$ at a small computational cost.

2.3.2 Properties

It is proven that the relative distance of the RRS function from the original B–spline becomes zero at the limit $N \to \infty$.

Define the relative distance Δy_N^m of the RRS function $y_N^m(k)$ from the B–spline $\beta^m(t)$ by

$$
\Delta y_N^m := \frac{\rho(\tilde{y}_N^m, \beta^m)^2}{\|\beta^m\|^2}, \qquad (2.14)
$$

where $\tilde{y}_N^m(t)$ is the staircase interpolation of $y_N^m(k)$. This interpolation is defined [2] so that it gets symmetric with respect to the midpoint of $[0, m)$, *i.e.*,

$$
\tilde{y}_N^m(t) := y_N^m(k), \quad \frac{k}{N} + \frac{m-1}{2N} \le t < \frac{k+1}{N} + \frac{m-1}{2N}. \qquad (2.15)
$$

and is written as

$$\tilde{y}_N^m(t) = \begin{cases} \beta^1(t), & m = 1, \\ \dfrac{m}{N^{m-1}} \displaystyle\sum_{i=0}^{m} \frac{(-1)^i}{(m-i)!i!} \prod_{j=1}^{m-1} \left(\left\lfloor tN - \frac{m-1}{2} \right\rfloor - Ni + j \right)_+, & m = 2, 3, \ldots, \end{cases} \tag{2.16}$$

where $\lfloor a \rfloor$ is the maximum integer not exceeding a.

Lemma 2.3

$$\tilde{y}_N^m(t) = 0, \quad t < 0 \ \text{ or } \ t \geq m. \tag{2.17}$$

Proof Clearly (2.17) is true in the case $m = 1$ since

$$\tilde{y}_N^1(t) = \beta^1(t) = \begin{cases} 1, & 0 \leq t < 1, \\ 0, & \text{elsewhere.} \end{cases} \tag{2.18}$$

Assume that (2.17) holds good for $m \geq 1$, *i.e.*,

$$\tilde{y}_N^m(t) = 0, \qquad t < 0 \ \text{ or } \ t \geq m. \tag{2.19}$$

Then we have

$$\begin{aligned} \tilde{y}_N^{m+1}(t) &= y_N^{m+1} \left(\left\lceil tN - \frac{m-1}{2} \right\rceil \right) \\ &= \frac{1}{N} \left(b_N^1 * y_N^m \right) \left(\left\lceil tN - \frac{m-1}{2} \right\rceil \right) \\ &= \frac{1}{N} \sum_{\ell=-\infty}^{\infty} b_N^1(\ell) y_N^m \left(\left\lceil tN - \frac{m-1}{2} \right\rceil - \ell \right) \\ &= \frac{1}{N} \sum_{\ell=0}^{N-1} y_N^m \left(\left\lceil tN - \frac{m-1}{2} \right\rceil - \ell \right) \\ &= \frac{1}{N} \sum_{\ell=0}^{N-1} \tilde{y}_N^m \left(t - \frac{\ell}{N} \right) \\ &= 0, \quad t < 0 \ \text{ or } \ t \geq m+1, \end{aligned} \tag{2.20}$$

which means (2.17) holds good with m replaced by $m + 1$. The above (2.18)–(2.20) and the mathematical induction complete a proof of Lemma 2.3 ■

On the basis of (2.1)–(2.17), we obtain the following theorem.

Theorem 2.4

$$\Delta y_N^m = \frac{\rho(\tilde{y}_N^m, \beta^m)^2}{\|\beta^m\|^2} \to 0, \quad (N \to \infty). \tag{2.21}$$

Proof By (2.18), obviously (2.21) is true for any $N \geq 2$ in the case $m = 1$. In the case $m \geq 2$, we have

$$\begin{aligned}
&\rho(\tilde{y}_N^m, \beta^m)^2 \\
&= \|\tilde{y}_N^m - \beta^m\|^2 \\
&= \int_{-\infty}^{\infty} |\tilde{y}_N^m(t) - \beta^m(t)|^2 \, dt \\
&= \int_0^m |\tilde{y}_N^m(t) - \beta^m(t)|^2 \, dt, \qquad \text{(by (2.17) of Lemma 2.3 and (2.2))} \\
&= \int_0^m \left| \frac{m}{N^{m-1}} \sum_{i=0}^{m} \frac{(-1)^i}{(m-i)!i!} \prod_{j=1}^{m-1} \left(\left\lceil tN - \frac{m-1}{2} \right\rceil - Ni + j \right)_+ \right. \\
&\qquad \left. - m \sum_{i=0}^{m} \frac{(-1)^i}{(m-i)!i!} (t-i)_+^{m-1} \right|^2 dt, \qquad \text{(by (2.1) and (2.16))} \\
&= \int_0^m \left| m \sum_{i=0}^{m} \frac{(-1)^i}{(m-i)!i!} \left\{ \frac{1}{N^{m-1}} \prod_{j=1}^{m-1} \left(\left\lceil tN - \frac{m-1}{2} \right\rceil - Ni + j \right)_+ - (t-i)_+^{m-1} \right\} \right|^2 dt \\
&\leq \int_0^m \left| m \sum_{i=0}^{m} \left| \frac{1}{N^{m-1}} \prod_{j=1}^{m-1} \left(\left\lceil tN - \frac{m-1}{2} \right\rceil - Ni + j \right)_+ - (t-i)_+^{m-1} \right| \right|^2 dt \\
&\leq \int_0^m \left| m \sum_{i=0}^{m} \left| \frac{1}{N^{m-1}} \prod_{j=1}^{m-1} \left(\left\lceil tN - \frac{m-1}{2} \right\rceil + j \right)_+ - (t)_+^{m-1} \right| \right|^2 dt \\
&= \int_0^m \left| m(m+1) \left| \frac{1}{N^{m-1}} \prod_{j=1}^{m-1} \left(\left\lceil tN - \frac{m-1}{2} \right\rceil + j \right)_+ - (t)_+^{m-1} \right| \right|^2 dt \\
&\leq m^2(m+1)^2 \int_0^m \left| \frac{1}{N^{m-1}} \left| \prod_{j=1}^{m-1} \left(\lceil tN \rceil - \frac{m}{2} + 1 + j \right)_+ - (\lceil tN \rceil)_+^{m-1} \right| \right|^2 dt \\
&\leq m^2(m+1)^2 \int_0^m \left| \frac{1}{N^{m-1}} \left| \prod_{j=1}^{m-1} (\lceil tN \rceil + 1 + j) - \lceil tN \rceil^{m-1} \right| \right|^2 dt \\
&= m^2(m+1)^2 \sum_{k=0}^{mN-1} \frac{1}{N} \left| \frac{1}{N^{m-1}} \left| \prod_{j=1}^{m-1} (k+1+j) - k^{m-1} \right| \right|^2
\end{aligned}$$

$$\leq m^2(m+1)^2\frac{mN}{N}\left|\frac{1}{N^{m-1}}\left|\prod_{j=1}^{m-1}(mN+1+j)-(mN)^{m-1}\right|\right|^2$$

$$= m^3(m+1)^2\left|\prod_{j=1}^{m-1}\left(m+\frac{1+j}{N}\right)-m^{m-1}\right|^2$$

$$= m^3(m+1)^2\left|\sum_{\ell=1}^{m-1}\frac{c_\ell}{N^\ell}m^{m-\ell-1}\right|^2 \to 0, \quad (N\to\infty),$$

where c_ℓ denotes the coefficient for $m^{m-\ell-1}$ of a polynomial in m, involving neither m nor N. This completes a proof of Theorem 2.4. ■

This theorem guarantees that the multifold RRS function can become as close to the original B–spline as required by making the sampling interval $1/N$ shorter.

Besides, the following equality is derived from (2.3) and (2.15).

$$\sum_{k=-\infty}^{\infty}\tilde{y}_N^m(t-k)=1, \quad \text{for any } m, N\in \boldsymbol{N} \text{ and } t\in \boldsymbol{R}. \tag{2.22}$$

2.3.3 Numerical Evaluation

In use of a discrete version of the B–splines in the continuous domain, an original B–spline is usually substituted by a staircase interpolation of the discrete version. Here the relative distance of the RRS function from the original B–spline is numerically compared with that of the discrete B–spline.

The relative distance Δb_N^m of the discrete B–spline $b_N^m(k)$ from the original B–spline $\beta^m(t)$ is defined by

$$\Delta b_N^m := \frac{\rho(\tilde{b}_N^m, \beta^m)^2}{\|\beta^m\|^2},$$

where $\tilde{b}_N^m(t)$ is the staircase interpolation of $b_N^m(k)$. The staircase interpolation $\tilde{b}_N^m(t)$ is

also defined so that it gets symmetric with respect to the midpoint of $[0, m]$, *i.e.*,

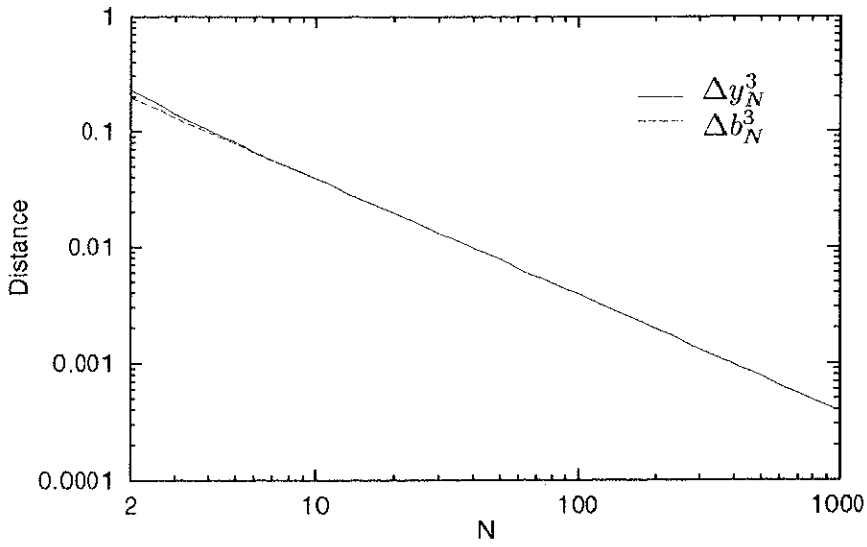
$$\tilde{b}_N^m(t) := \begin{cases} b_N^1(k), & \dfrac{k}{N} \le t < \dfrac{k+1}{N}, \qquad m = 1, \\ b_N^m(k), & \dfrac{k}{N} - \dfrac{1}{2N} \le t < \dfrac{k+1}{N} - \dfrac{1}{2N}, \quad m = 2, 3, \cdots \end{cases}$$

$$= \begin{cases} \beta^1(t), & m = 1, \\ \dfrac{m}{N^{m-1}} \displaystyle\sum_{i=0}^{m} \frac{(-1)^i}{(m-i)!i!} \left(\left\lceil tN + \frac{1}{2} \right\rceil - Ni \right)_+^{m-1}, & m = 2, 3, \cdots. \end{cases}$$

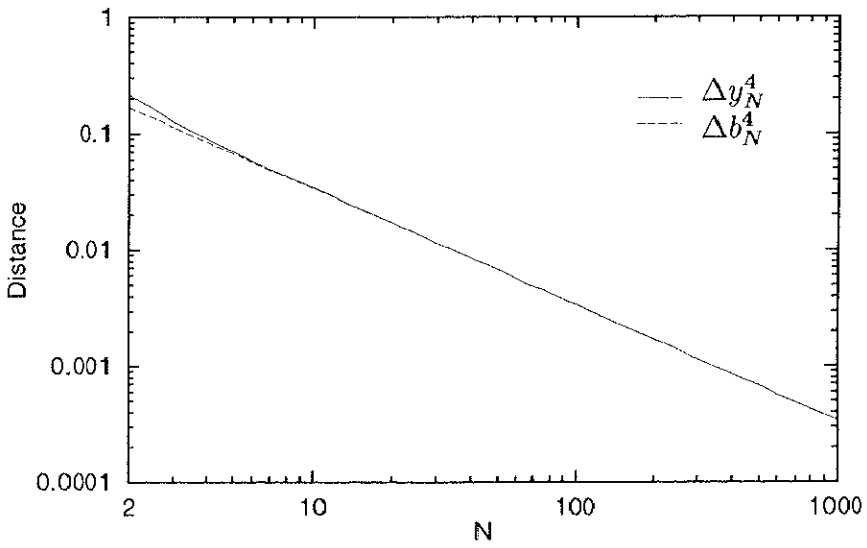
Figure 2.1 plots the relative distances Δy_N^m and Δb_N^m versus N. Figure 2.1 shows that the multifold RRS function gets as close to the original B–spline as the discrete B–splines in the sense of RMS error for N larger than ten in the cases $m = 3$ and 4. In this situation, the RRS function can do almost the same in substituting the original B–spline in signal approximation as the discrete B–spline. Besides, the RRS function can be evaluated by fewer operations because it is expressed simply by the multifold discrete convolution $b_N^1 * b_N^1 * \cdots * b_N^1$ in (2.8) that can be implemented by the fast RRS digital filters (Saramäki *et al.*, 1990) (costing m subtractions and mN additions per knot interval) while the discrete B–splines require an additional short linear-phase FIR filter ($(m-2)N$ additions and $\lfloor m/2 \rfloor N$ multiplications per knot interval) for $\cdot * b_1^m$ in (2.9).

2.4 Approximate B-spline Decomposition and Reconstruction

In this section, we start from the ideal scheme Fig.2.2(a) of B-spline decomposition and reconstruction to reach its practicable approximation Fig.2.2(e).



(a) in the case $m = 3$



(b) in the case $m = 4$

Figure 2.1: The distances Δy_N^m (solid line) and Δb_N^m (broken line).

2.4.1 Truncation of the Infinite Sequence

Let $f \in \boldsymbol{L}^2(\boldsymbol{R}) := \{f : \boldsymbol{R} \to \boldsymbol{R} | \int_{-\infty}^{\infty} |f(t)|^2 dt < \infty\}$ denote a given continuous-time signal. The ideal B-spline coefficients $\{c_k\}_{k \in \boldsymbol{Z}}$ for best approximating f are mathematically

expressed by

$$c_k = \int_{-\infty}^{\infty} f(t)\bar{\beta}^m(t-k)dt. \tag{2.23}$$

The ideal reconstruction s of f, which is reconstructed from the coefficients $\{c_k\}_{k\in Z}$, is expressed by

$$s(t) = \sum_{k=-\infty}^{\infty} c_k \beta^m(t-k). \tag{2.24}$$

The ideal decomposition (2.23) and reconstruction (2.24) are represented by the block diagram of Fig.2.2(a). The system $\boxed{*\beta^m}$ in Fig.2.2(a) is feasible [1, 10, 11], but $\boxed{\circledast\ \bar{\beta}^m}$ is not feasible because $\bar{\beta}^m$ is an ever oscillating function over the time axis.

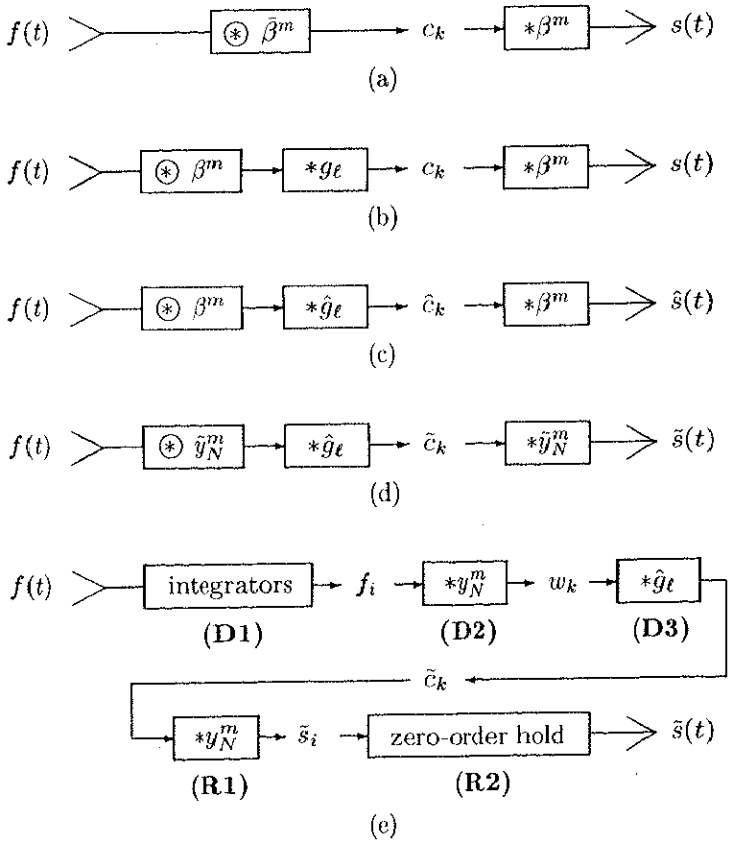


Figure 2.2: Scheme of B-spline decomposition and reconstruction.

Substituting (2.5) for $\bar{\beta}^m$, we have (2.23) rewritten as follows:

$$c_k = \int_{-\infty}^{\infty} f(t) \sum_{\ell=-\infty}^{\infty} g_\ell \beta^m(t-(k+\ell))dt$$
$$= \sum_{\ell=-\infty}^{\infty} g_\ell \int_{-\infty}^{\infty} f(t)\beta^m(t-(k+\ell))dt. \tag{2.25}$$

This formula means that the convolution integral $\boxed{\circledast\ \bar{\beta}^m}$ in Fig.2.2(a) is equivalent to the cascade of the convolution integral $\boxed{\circledast\ \beta^m}$ and the discrete convolution $\boxed{*\ g_\ell}$ as in Fig.2.2(b). The system $\boxed{\circledast\ \beta^m}$ in Fig.2.2(b) is feasible, but $\boxed{*\ g_\ell}$ is still not feasible because the coefficients $\{g_\ell\}_{\ell\in Z}$ remain nonzero throughout the time axis. We have to replace the infinite sequence $\{g_\ell\}_{\ell\in Z}$ by a finite one.

Replacing $\{g_\ell\}_{\ell\in Z}$ by its truncated version $\{\hat{g}_\ell\}_{\ell\in Z}$ of (2.7), we have a finite system $\boxed{*\ \hat{g}_\ell}$ in Fig.2.2(c) that yields approximate B-spline coefficients

$$\hat{c}_k := \sum_{\ell=-M}^{M} \hat{g}_\ell \int_{-\infty}^{\infty} f(t)\beta^m(t-(k+\ell))dt, \tag{2.26}$$

and an approximate reconstruction

$$\hat{s}(t) := \sum_{k=-\infty}^{\infty} \hat{c}_k \beta^m(t-k). \tag{2.27}$$

2.4.2 Substitution of B-splines by Multifold RRS Functions

The processes in Fig.2.2(c) could be implemented by analog circuits at the cost of much analog hardware complexity and corruption with internal analog noise. We shall take a different approach in this paper: we substitute the staircase interpolation $\tilde{y}_N^m$ of the RRS functions for the B-spline β^m in Fig.2.2(c).

This substitution rewrites Fig.2.2(c) as Fig.2.2(d) which yields approximate B-spline coefficients

$$\tilde{c}_k := \sum_{\ell=-M}^{M} \hat{g}_\ell \int_{-\infty}^{\infty} f(t)\tilde{y}_N^m(t-(k+\ell))dt, \tag{2.28}$$

and an approximate reconstruction

$$\tilde{s}(t) := \sum_{k=-\infty}^{\infty} \tilde{c}_k \tilde{y}_N^m(t-k). \tag{2.29}$$

The process $\boxed{* \hat{g}_\ell}$ in Fig.2.2(d) can be implemented by an ordinary FIR digital filter. On the other hand, we need further elaboration with $\boxed{\circledast\, \tilde{y}_N^m}$ and $\boxed{*\tilde{y}_N^m}$ of Fig.2.2(d), which will be respectively discussed in Subsections 2.4.3 and 2.4.4.

2.4.3 Decomposition Scheme

For the implementation of $\boxed{\circledast\, \tilde{y}_N^m}$, we rearrange (2.28) as follows:

$$\begin{aligned}
\tilde{c}_k &= \sum_{\ell=-M}^{M} \hat{g}_\ell \int_{-\infty}^{\infty} f(t)\ \tilde{y}_N^m(t-(k+\ell))dt \\
&= \sum_{\ell=-M}^{M} \hat{g}_\ell \int_{k+\ell}^{k+\ell+m} f(t)\ \tilde{y}_N^m(t-(k+\ell))dt, \quad \text{(by Lemma 2 of [2])} \\
&= \sum_{\ell=-M}^{M} \hat{g}_\ell \int_{k+\ell}^{k+\ell+m} f(t)\ y_N^m\left(\left\lfloor tN - \frac{m-1}{2} \right\rfloor - (k+\ell)N \right) dt, \quad \text{(by (2.15))} \\
&= \sum_{\ell=-M}^{M} \hat{g}_\ell \left[\sum_{i=0}^{mN-1} \left\{ \int_{k+\ell+i/N+(m-1)/2N}^{k+\ell+(i+1)/N+(m-1)/2N} f(t)\ y_N^m(i)dt \right\} \right] \\
&= \sum_{\ell=-M}^{M} \hat{g}_\ell \left[\sum_{i=0}^{mN-1} \left\{ \int_{k+\ell+i/N+(m-1)/2N}^{k+\ell+(i+1)/N+(m-1)/2N} f(t)dt \right\} y_N^m(i) \right].
\end{aligned}$$

The last formula can be divided into the three steps:

$$f_i := N \int_{i/N+(m-1)/2N}^{(i+1)/N+(m-1)/2N} f(t)dt, \tag{2.30}$$

$$w_\ell := \frac{1}{N} \sum_{i=0}^{mN-1} f_{i+\ell N}\ y_N^m(i), \tag{2.31}$$

$$\tilde{c}_k = \sum_{\ell=-M}^{M} g_\ell w_{k+\ell}, \tag{2.32}$$

which respectively corresponds to the following processes:

(D1) Averaged sampling of $f(t)$ to obtain $\{f_i\}_{i\in Z}$ by (2.30).

(D2) A downsampling digital filter to compute $\{w_\ell\}_{\ell\in Z}$ from $\{f_i\}_{i\in Z}$ by (2.31).

(D3) An ordinary FIR filter to compute the approximate B-spline coefficients $\{\tilde{c}_k\}_{k\in\boldsymbol{Z}}$ from $\{w_\ell\}_{\ell\in\boldsymbol{Z}}$ by (2.32).

Those three processes constitute the first half of Fig.2.2(e).

Process **(D1)** samples the average value of $f(t)$ for each sampling interval, which can be implemented by the twin integrators and an A-D converter as shown in Fig.2.3. Each of the twins averages $f(t)$ in turn for every two sampling intervals. After one integrator finishes integration, it holds the integrated value for some time for the A-D conversion and then discharges while the other is integrating. Process **(D2)** is a combination of RRS digital filters [19] and a downsampler. To make it work fast, the process **(D2)** is implemented by the technique [35] that works well with the RRS filters as shown in Fig.2.4. Process **(D3)** is implemented by an ordinary FIR filter. It should be noted that the FIR filter costs little computational complexity since it deals only with down-sampled data.

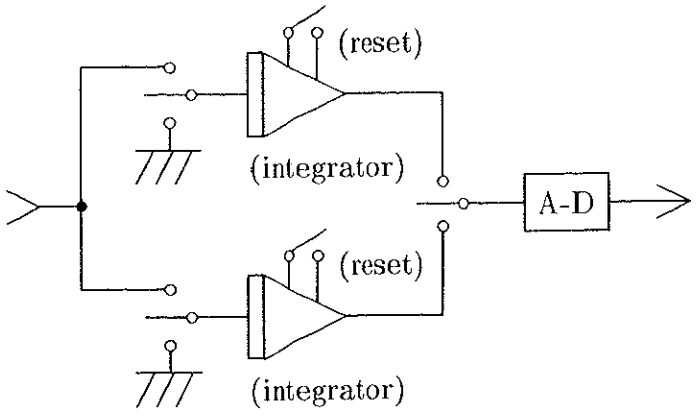


Figure 2.3: A construction of the process **(D1)** of the decomposition scheme.

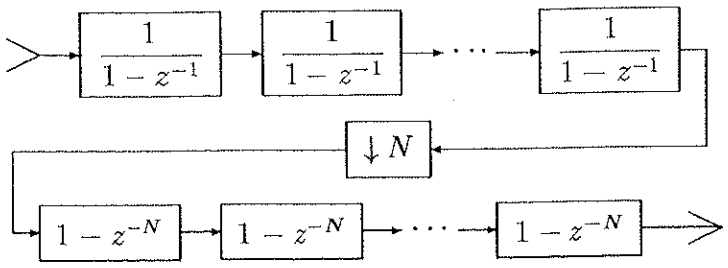


Figure 2.4: A construction of the process **(D2)** of the decomposition scheme.

2.4.4 Reconstruction Scheme

For the implementation of $\boxed{*\tilde{y}_N^m}$, we rearrange (2.29) as follows:

$$\begin{aligned}
\tilde{s}(t) &= \sum_{k=-\infty}^{\infty} \tilde{c}_k \tilde{y}_N^m(t-k) \\
&= \sum_{k=-\infty}^{\infty} \tilde{c}_k \tilde{y}_N^m\left(\frac{1}{N}\left\lfloor tN - \frac{m-1}{2}\right\rfloor + \frac{m-1}{2N} - k\right) \\
&= \sum_{k=-\infty}^{\infty} \tilde{c}_k y_N^m\left(\left\lfloor tN - \frac{m-1}{2}\right\rfloor - kN\right), \quad \text{(by (2.15))} \\
&= \sum_{k=\lfloor \lfloor tN-(m-1)/2\rfloor/N\rfloor - m+1}^{\lfloor \lfloor tN-(m-1)/2\rfloor/N\rfloor} \tilde{c}_k y_N^m\left(\left\lfloor tN - \frac{m-1}{2}\right\rfloor - kN\right), \quad \text{(by Lemma 2 of [2])}
\end{aligned}$$

The last formula can be divided into two steps:

$$\tilde{s}_i := \sum_{k=\lfloor i/N\rfloor - m+1}^{\lfloor i/N\rfloor} \tilde{c}_k \, y_N^m(i-kN) \tag{2.33}$$

$$\tilde{s}(t) = \tilde{s}_i, \quad \frac{i}{N} + \frac{m-1}{2N} \le t < \frac{i+1}{N} + \frac{m-1}{2N}, \tag{2.34}$$

which respectively correspond to the following two processes:

(R1) An upsampling digital filter to compute $\{\tilde{s}_i\}_{i\in Z}$ from B-spline coefficients $\{\tilde{c}_\ell\}_{\ell\in Z}$ by (2.33).

(R2) A Zero-order hold to obtain the continuous-time signal $\tilde{s}(t)$ from $\{\tilde{s}_i\}_{i\in Z}$ by (2.34).

Those two processes constitute the last half of Fig.2.2(e).

Process **(R1)** is a combination of an upsampler and RRS digital filters [35] as in Fig.2.5. Process **(R2)** needs only a zero-order hold circuit.

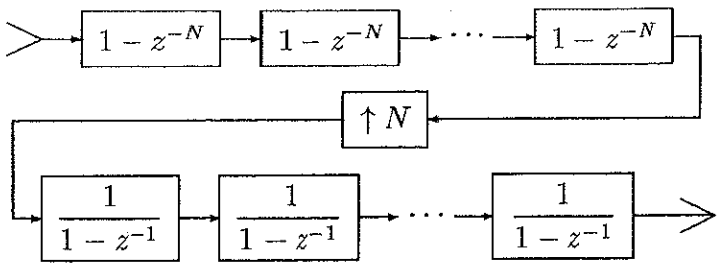


Figure 2.5: A construction of the process **(R1)** of the reconstruction scheme.

2.5 Analysis and Design Procedure

In this section, we analyze the relationship between the stability of the scheme, the maximum possible amplitude of the internal signals, the approximation error due to the two substitutions made in Section 2.4, the circuit parameters M, N, and the data representation format.

2.5.1 Stability and Maximum Possible Amplitude

2.5.1.1 Decomposition Scheme

The decomposition scheme is constructed by **(D1)**, **(D2)** and **(D3)** as shown in Subsection 2.4.3. Apparently **(D1)** and **(D3)** are stable because they are just averaging circuits. The process **(D2)** is externally stable but its structure internally includes the critically unstable factor $1/(1-z^{-1})$. This factor causes accumulation that may cause internal overflows. This problem can be settled by the special property [36, 26] of the RRS digital filter: The internal overflows do not matter if the amplitude of both input and output is mathematically less than some constant A and if we use the fixed-point 2's complement

representation that can express $\pm A$ for the internal computation. Thus the stability of **(D2)** can be easily checked by the external gain of **(D2)**. Now we evaluate the maximum possible amplitude of the output signals when the input f of **(D1)** satisfies $||f||_{L^\infty} < 1$.

Proposition 2.5 If the input signal f satisfies $||f||_{L^\infty} < 1$, the output of **(D1)**, **(D2)** and **(D3)** respectively satisfy

$$\|\{f_i\}_{i\in\boldsymbol{Z}}\|_{\ell^\infty} < 1, \quad \|\{w_\ell\}_{\ell\in\boldsymbol{Z}}\|_{\ell^\infty} < 1, \quad \|\{\tilde{c}_k\}_{k\in\boldsymbol{Z}}\|_{\ell^\infty} < \alpha,$$

where

$$\alpha := \frac{1}{N}\sum_{i=0}^{mN-1}\left|\sum_{\ell=-M}^{M}\hat{g}_\ell y_N^m(i-\ell N)\right|, \tag{2.35}$$

is a constant depending on m, M and N.

Proof Those inequalities are derived from (2.30), (2.31), (2.32) and the triangular inequality. ■

The parameter α can be numerically calculated by (2.35).

According to Proposition 2.5, we can implement **(D2)** by normalizing the input f to satisfy $||f||_{L^\infty} < 1$ and employing the fixed-point representation with only a sign bit on the left of the decimal point for **(D1)** and **(D2)**. There may be internal overflows with **(D2)**, but none of them affects the output.

When **(D3)** is constructed as an FIR filter of the transversal type, the internal signals

$$v_k^j := \sum_{\ell=-M}^{j-M} g_\ell w_{k+\ell}, \tag{2.36}$$

of **(D3)** must be represented properly.

Proposition 2.6 If the input signal f satisfies $||f||_{L^\infty} < 1$, the internal signals $\{v_k^j\}_{j=1,2,\ldots,2M-1}$ of **(D3)** satisfy $\|\{v_k^j\}_{j=1,2,\ldots,2M-1}\|_{\ell^\infty} < \alpha$.

Proof It follows from (2.36) and Proposition 2.5. ■

Proposition 2.6 guarantees that **(D3)** works properly without any internal overflows if we place a sign bit and Y bits on the left of the decimal point, where $Y \in \boldsymbol{N}$, $\log_2 \alpha \leq Y < 1 + \log_2 \alpha$.

The above implementation makes all of the decomposition scheme stable and free from any external overflows.

2.5.1.2 Reconstruction Scheme

The reconstruction scheme is constructed by **(R1)** and **(R2)** as shown in Subsection 2.4.4. The maximum possible amplitude $\|\tilde{s}\|_{L^\infty}$ of the approximate reconstruction $\tilde{s}$ is given by the following Proposition.

Proposition 2.7 If the input signal f of the decomposition scheme satisfies $\|f\|_{L^\infty} < 1$, the outputs of **(R1)** and **(R2)** respectively satisfy $\|\{\tilde{s}_i\}_{i\in Z}\|_{\ell^\infty} < \alpha$ and $\|\tilde{s}\|_{L^\infty} < \alpha$.

Proof It follows from (2.33), (2.34) and Proposition 1. ■

Proposition 2.7 guarantees the stability of **(R1)**, and means that we can implement **(R1)** and **(R2)** by employing the fixed-point representation with a sign bit and Y bits on the left of the decimal point.

The above implementation makes all the reconstruction scheme stable and free from any external overflows.

2.5.2 Approximation Error

The approximate reconstruction $\tilde{s}$ of Fig.2.2(e) is affected by two substitutions: the truncated B-spline dual coefficients and the RRS functions. We shall evaluate the approximation error due to the substitutions mathematically. For that purpose, we prepare the

following two Lemmas.

Lemma 2.8 The distance $\|\hat{s} - s\|_{\boldsymbol{L}^\infty}$ between the reconstruction s and the approximate reconstruction $\hat{s}$ satisfies

$$\|\hat{s} - s\|_{\boldsymbol{L}^\infty} \leq \|\{\hat{c}_k - c_k\}_{k\in\boldsymbol{Z}}\|_{\boldsymbol{\ell}^\infty} \leq \|f\|_{\boldsymbol{L}^\infty} \cdot \left\| \sum_{|\ell|>M} g_\ell \beta^m(t-\ell) \right\|_{\boldsymbol{L}^1}. \tag{2.37}$$

Besides, the approximate B-spline coefficients $\{\hat{c}_k\}_{k\in\boldsymbol{Z}}$ satisfy the following inequality.

$$\|\{\hat{c}_k\}_{k\in\boldsymbol{Z}}\|_{\boldsymbol{\ell}^\infty} \leq \|f\|_{\boldsymbol{L}^\infty} \cdot \left\| \sum_{\ell=-M}^{M} \hat{g}_\ell \beta^m(t-\ell) \right\|_{\boldsymbol{L}^1}. \tag{2.38}$$

Proof Inequality (2.37) follows from

$$\begin{aligned}
&\|\hat{s} - s\|_{\boldsymbol{L}^\infty} \\
&= \left\| \sum_{k=-\infty}^{\infty} (\hat{c}_k - c_k)\, \beta^m(t-k) \right\|_{\boldsymbol{L}^\infty}, \quad \text{(by (2.24) and (2.27))} \\
&\leq \left\| \sum_{k=-\infty}^{\infty} \left\{ \sup_{i\in\boldsymbol{Z}} |\hat{c}_i - c_i| \right\} \beta^m(t-k) \right\|_{\boldsymbol{L}^\infty} \\
&\leq \|\{\hat{c}_k - c_k\}_{k\in\boldsymbol{Z}}\|_{\boldsymbol{\ell}^\infty} \cdot \left\| \sum_{k=-\infty}^{\infty} \beta^m(t-k) \right\|_{\boldsymbol{L}^\infty} \\
&= \|\{\hat{c}_k - c_k\}_{k\in\boldsymbol{Z}}\|_{\boldsymbol{\ell}^\infty}, \qquad \text{(by (2.3))} \\
&= \sup_{k\in\boldsymbol{Z}} \left| \sum_{|\ell|>M} g_\ell \int_{-\infty}^{\infty} f(t)\beta^m(t-(k+\ell))dt \right|, \quad \text{(by (2.25) and (2.26))} \\
&= \sup_{k\in\boldsymbol{Z}} \left| \int_{-\infty}^{\infty} f(t) \sum_{|\ell|>M} g_\ell \beta^m(t-(k+\ell))dt \right| \\
&\leq \sup_{k\in\boldsymbol{Z}} \left[\int_{-\infty}^{\infty} \left\{ \sup_{t\in\boldsymbol{R}} |f(t)| \right\} \cdot \left| \sum_{|\ell|>M} g_\ell \beta^m(t-(k+\ell)) \right| dt \right] \\
&= \|f\|_{\boldsymbol{L}^\infty} \cdot \sup_{k\in\boldsymbol{Z}} \left[\int_{-\infty}^{\infty} \left| \sum_{|\ell|>M} g_\ell \beta^m(t-(k+\ell)) \right| dt \right] \\
&= \|f\|_{\boldsymbol{L}^\infty} \cdot \int_{-\infty}^{\infty} \left| \sum_{|\ell|>M} g_\ell \beta^m(t-\ell) \right| dt \\
&= \|f\|_{\boldsymbol{L}^\infty} \cdot \left\| \sum_{|\ell|>M} g_\ell \beta^m(t-\ell) \right\|_{\boldsymbol{L}^1}.
\end{aligned}$$

Inequality (2.38) is derived in a similar way. ■

Lemma 2.9 The distance $\|\tilde{s} - \hat{s}\|_{\boldsymbol{L}^\infty}$ between two approximate reconstructions $\hat{s}$ and $\tilde{s}$ satisfies

$$\begin{aligned}\|\tilde{s} - \hat{s}\|_{\boldsymbol{L}^\infty} \leq \|f\|_{\boldsymbol{L}^\infty} \cdot \left\| \sum_{\ell=-M}^{M} \hat{g}_\ell \{\tilde{y}_N^m(t-\ell) - \beta^m(t-\ell)\} \right\|_{\boldsymbol{L}^1} \\ + \|f\|_{\boldsymbol{L}^\infty} \cdot \left\| \sum_{\ell=-M}^{M} \hat{g}_\ell \beta^m(t-\ell) \right\|_{\boldsymbol{L}^1} \cdot \left\| \sum_{k=-\infty}^{\infty} |\tilde{y}_N^m(t-k) - \beta^m(t-k)| \right\|_{\boldsymbol{L}^\infty} \end{aligned} \tag{2.39}$$

Proof

$$\begin{aligned}&\|\tilde{s} - \hat{s}\|_{\boldsymbol{L}^\infty} \\ &= \left\| \sum_{k=-\infty}^{\infty} \tilde{c}_k \tilde{y}_N^m(t-k) - \sum_{k=-\infty}^{\infty} \hat{c}_k \beta^m(t-k) \right\|_{\boldsymbol{L}^\infty}, \quad \text{(by (2.27) and (2.29))} \\ &= \left\| \sum_{k=-\infty}^{\infty} (\tilde{c}_k - \hat{c}_k)\, \tilde{y}_N^m(t-k) + \sum_{k=-\infty}^{\infty} \hat{c}_k \{\tilde{y}_N^m(t-k) - \beta^m(t-k)\} \right\|_{\boldsymbol{L}^\infty} \\ &\leq \left\| \sum_{k=-\infty}^{\infty} (\tilde{c}_k - \hat{c}_k) \tilde{y}_N^m(t-k) \right\|_{\boldsymbol{L}^\infty} + \left\| \sum_{k=-\infty}^{\infty} \hat{c}_k \{\tilde{y}_N^m(t-k) - \beta^m(t-k)\} \right\|_{\boldsymbol{L}^\infty}. \end{aligned} \tag{2.40}$$

Here, the following inequality holds.

$$\begin{aligned}&\left\| \sum_{k=-\infty}^{\infty} (\tilde{c}_k - \hat{c}_k) \tilde{y}_N^m(t-k) \right\|_{\boldsymbol{L}^\infty} \\ &\leq \left\| \sum_{k=-\infty}^{\infty} \left\{ \sup_{i \in \boldsymbol{Z}} (\tilde{c}_i - \hat{c}_i) \right\} \cdot \tilde{y}_N^m(t-k) \right\|_{\boldsymbol{L}^\infty} \\ &= \|\{\tilde{c}_k - \hat{c}_k\}_{k \in \boldsymbol{Z}}\|_{\boldsymbol{\ell}^\infty} \cdot \left\| \sum_{k=-\infty}^{\infty} \tilde{y}_N^m(t-k) \right\|_{\boldsymbol{L}^\infty} \\ &= \|\{\tilde{c}_k - \hat{c}_k\}_{k \in \boldsymbol{Z}}\|_{\boldsymbol{\ell}^\infty}, \quad \text{(by (2.22))} \\ &= \left\| \sum_{\ell=-M}^{M} \hat{g}_\ell \int_{-\infty}^{\infty} f(t) \{\tilde{y}_N^m(t-(k+\ell)) - \beta^m(t-(k+\ell))\} dt \right\|_{\boldsymbol{\ell}^\infty}, \end{aligned}$$

(by (2.26) and (2.28))

$$
\begin{aligned}
&= \left\| \int_{-\infty}^{\infty} f(t+k) \sum_{\ell=-M}^{M} \hat{g}_\ell \{\tilde{y}_N^m(t-\ell) - \beta^m(t-\ell)\} dt \right\|_{\ell^\infty} \\
&\leq \|f\|_{L^\infty} \cdot \left\| \sum_{\ell=-M}^{M} \hat{g}_\ell \{\tilde{y}_N^m(t-\ell) - \beta^m(t-\ell)\} \right\|_{L^1}.
\end{aligned}
\tag{2.41}
$$

On the other hand, the following inequality also holds.

$$
\begin{aligned}
&\left\| \sum_{k=-\infty}^{\infty} \hat{c}_k \{\tilde{y}_N^m(t-k) - \beta^m(t-k)\} \right\|_{L^\infty} \\
&\quad \leq \left\| \sum_{k=-\infty}^{\infty} \left\{ \sup_{i \in \mathbf{Z}} |\hat{c}_i| \right\} \cdot |\tilde{y}_N^m(t-k) - \beta^m(t-k)| \right\|_{L^\infty} \\
&\quad = \|\{\hat{c}_k\}_{k \in \mathbf{Z}}\|_{\ell^\infty} \cdot \left\| \sum_{k=-\infty}^{\infty} |\tilde{y}_N^m(t-k) - \beta^m(t-k)| \right\|_{L^\infty} \\
&\quad \leq \|f\|_{L^\infty} \cdot \left\| \sum_{\ell=-M}^{M} \hat{g}_\ell \beta^m(t-\ell) \right\|_{L^1} \cdot \left\| \sum_{k=-\infty}^{\infty} |\tilde{y}_N^m(t-k) - \beta^m(t-k)| \right\|_{L^\infty}, \quad \text{(by (2.38))}.
\end{aligned}
\tag{2.42}
$$

The above (2.40)–(2.42) complete the proof of Lemma 2.9. ■

From (2.37) and (2.39), the following Theorem holds.

Theorem 2.10 The approximation error $\|\tilde{s} - s\|_{L^\infty} / \|f\|_{L^\infty}$ satisfies

$$
\frac{\|\tilde{s} - s\|_{L^\infty}}{\|f\|_{L^\infty}} \leq \frac{\|\tilde{s} - \hat{s} + \hat{s} - s\|_{L^\infty}}{\|f\|_{L^\infty}} \leq \frac{\|\tilde{s} - \hat{s}\|_{L^\infty}}{\|f\|_{L^\infty}} + \frac{\|\hat{s} - s\|_{L^\infty}}{\|f\|_{L^\infty}} \leq \varepsilon,
$$

where

$$
\begin{aligned}
\varepsilon := &\left\| \sum_{|\ell| > M} g_\ell \beta^m(t-\ell) \right\|_{L^1} + \left\| \sum_{\ell=-M}^{M} \hat{g}_\ell \{\tilde{b}_N^m(t-\ell) - \beta^m(t-\ell)\} \right\|_{L^1} \\
&+ \left\| \sum_{\ell=-M}^{M} \hat{g}_\ell \beta^m(t-\ell) \right\|_{L^1} \cdot \left\| \sum_{k=-\infty}^{\infty} |\tilde{y}_N^m(t-k) - \beta^m(t-k)| \right\|_{L^\infty}.
\end{aligned}
\tag{2.43}
$$

■

Figure 2.6 plots the upper bound ε in (2.43) evaluated numerically for finite M and N. Figure 2.6 illustrates that the normalized distance $\|\tilde{s} - s\|_{L^\infty} / \|f\|_{L^\infty}$ can be made less than any ε by choosing appropriate M and N.

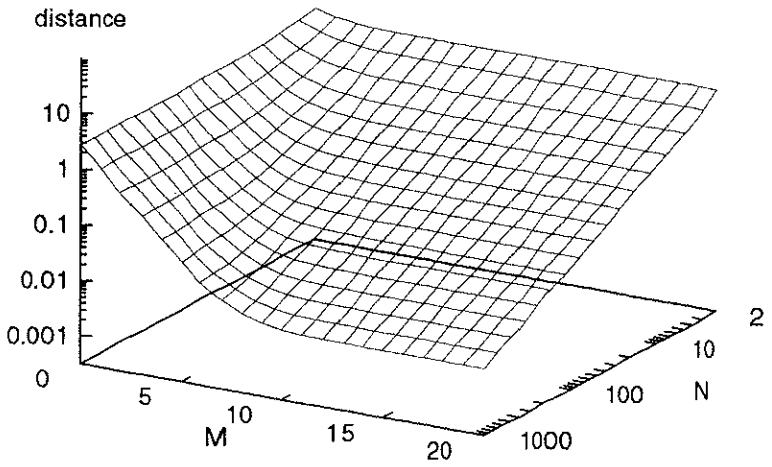


Figure 2.6: Upper bound ε of the approximation error versus parameters M and N in the case $m = 3$.

2.5.3 Determination of Circuit Parameters

We now summarize how to determine the circuit parameters.

First, M and N are chosen so as to make the upper bound ε smaller than required. We can choose M and N by looking at Fig.2.6.

We use the fixed-point 2's complement representation. For the A-D converter in **(D1)** and the RRS filters in **(D2)**, the maximum possible amplitude of the internal digital signals is less than one if the input f satisfies $\|f\|_{L^\infty} < 1$. So, we place only a sign bit on the left of the decimal point for **(D1)** and **(D2)**.

For the FIR filter in **(D3)**, the RRS filter in **(R1)** and the D-A converter in **(R2)**, the maximum amplitude of the internal digital signals is less than α if the input f satisfies

$\|f\|_{L^\infty} < 1$. So we place a sign bit and additional Y bits on the left of the decimal point for **(D3)**, **(R1)** and **(R2)**.

It is unreasonable to spend too many bits on the right side since the final precision is limited by the approximation error. So we place X bits on the right of the decimal point so that the round off error 2^{-X-1} gets as small as ε.

2.6 Example and Simulation

We present an example of the scheme in the case $m = 3$ with approximation error less than 1%, and its numerical simulation to confirm that it works.

2.6.1 Design Example

According to Fig.2.6, the upper bound ε of the approximation error is made less than 0.01 by choosing $M = 20$ and $N = 512$. Numerical evaluation of the constant α by (2.35) gives $\alpha = 2.387$. Assuming that the input f is normalized as $\|f\|_{L^\infty} < 1$, we employ the fixed-point data format with only the sign bit on the left of the decimal point for **(D1)** and **(D2)**, and that with the sign bit and two more bits on the left for **(D3)**, **(R1)** and **(R2)** in accordance with the discussion in 2.5. On the right of the decimal point, we place seven bits to have the round-off error of roughly 1%. The above circuit parameters are summarized in Table 2.1.

2.6.2 Simulation

The example scheme designed in 2.6.1 was simulated by a program that implements the specifications in Table 2.1. The ideal reconstruction s was numerically computed by discretizing the ideal mathematical formulae by a minute sampling interval.

Table 2.1: Specifications of the design example.

Order m of B–spline	3
Truncation length M of dual coefficients	20
Number of samples in [0,1), N	512
Bit length X on the right of the decimal point,	7
Bit length Y on the left of the decimal point,	2

Figure 2.7 plots the approximation error $\|\tilde{s} - s\|_{L^\infty} / \|f\|_{L^\infty}$ and the corresponding upper bound ε for sinusoidal input $f(t) = (127/128)\sin 2\pi t/T$ of various T. The error makes its largest peak around $T = 2$ which is close to the upper bound ε, but does not exceed ε. The period 2 is twice the sampling interval of the B-spline coefficients. This suggests that there must be some penetrating reason for the peak, but it is yet to be found.

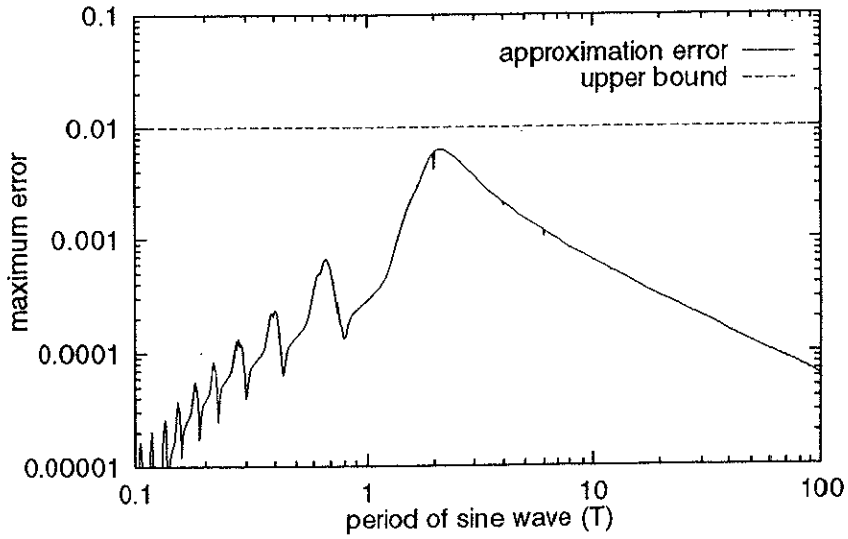


Figure 2.7: Approximation error $\|\tilde{s} - s\|_{L^\infty} / \|f\|_{L^\infty}$ (solid line) for sinusoidal input and the upper bound ε (dotted line) in the case $m = 3$.

For a special input

$$f(t) = \begin{cases} 127/128, & 0.781 \leq t < 2.211, \\ -127/128, & 0 \leq t < 0.781, \\ & 2.211 \leq t < 3, \\ 0, & \text{otherwise}, \end{cases}$$

the output $\{\tilde{c}_k\}_{k\in Z}$ of **(D3)** almost reaches the maximum possible amplitude $\alpha = 2.387$. Figure 2.8 plots the example input f and the corresponding output $\{\tilde{c}_k\}_{k\in Z}$. The output $\{\tilde{c}_k\}_{k\in Z}$ certainly reached α, as denoted above. The added Y bits work for such a case.

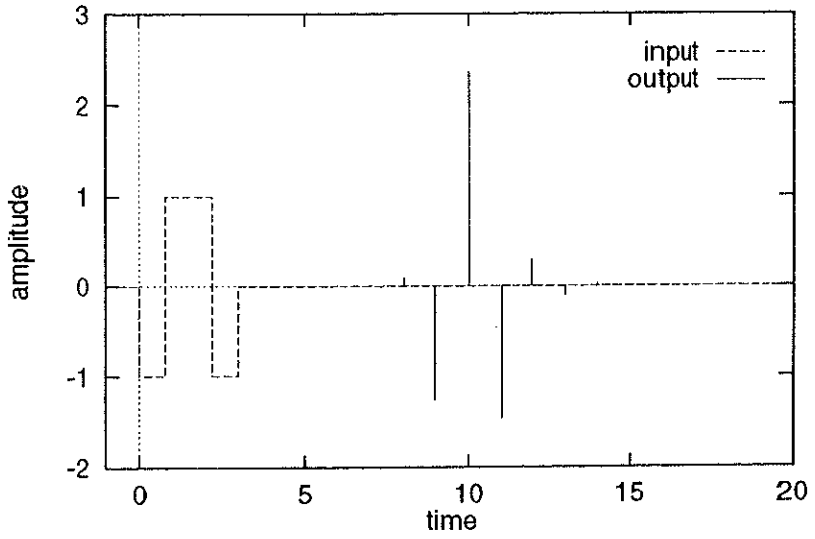


Figure 2.8: Example input s (dotted line) and output $\{\tilde{c}_k\}_{k\in Z}$ (solid line) that almost reaches α in the case $m = 3$.

Throughout the simulation, all the overflows detected were those expected inside the RRS digital filter. No unexpected overflows were detected.

2.7 Summary

In this chapter, a simple scheme of B-spline decomposition and reconstruction was studied. Its analysis yielded conditions for the circuit parameters to avoid instability and overflow, and to meet required precision.

As presented by the design example and the simulation, we need a high oversampling rate such as 512 to make the error around 1%. This situation is almost the same even if we use sampled values of the B-splines because they are almost the same as the RRS functions with respect to approximating the B-splines [2, 4, 14]. In the meantime, the present approach is computationally more advantageous.

It would be helpful for the design if we have a formula expressing the exact period T in Fig. 2.7 for which the approximation error gets largest. Derivation of that formula seems very difficult but is important for a more advanced design procedure to be developed in further studies.

It also remains as one of further studies to know which combination of $\{N, M, m, f(t)\}$ change the inequality (2.43) into an equality. Such combination seems to exist, however we still don't know if it really exists or not.

Chapter 3

Frequency Transformation Matrices and Their Properties

3.1 Introduction

Frequency transformations derive filters of various types from a filter of low-pass type. The transformation formulae have been established for analog filters in the s-domain [30, p.258], and for IIR digital filters in the z-domain [23],[30, p.260]. Furthermore, bilinear transformation [30, pp.219–224] derives IIR digital filters from analog filters, and also enables the inverse derivation. The transformation formulae are well-known and widely used. However the direct application of these transformations yields a complicated formula of the target transfer function, which has to be reduced by hand computation into the form of a rational polynomial. The yielded formula becomes more complicated for a transfer function with a higher order which can be used in practice to realize minute characteristics.

To dissolve this complexity, a bilinear transformation matrix [20] has been formulated which transforms coefficients of a polynomial in the s-domain to those in the z-domain. This matrix can replace the hand computation of the bilinear transformation for polynomials by an automatic procedure. Similarly, the hand computation of those transformations

could be replaced by an automatic procedure if the relation between the coefficients of the transfer functions is formulated.

This chapter first aims at deriving the explicit formulae which connect those coefficients by matrices. Subsections 3.2.1 and 3.2.2 give explicit formulae for frequency transformation matrices, which represent the frequency transformation for analog filters and that for IIR digital filters, respectively. Then in Subsection 3.2.3, the bilinear transformation matrix is summarized, and it is applied to the bilinear transformation of rational polynomials.

The matrices also enable automatic design of IIR digital filters from an analog low-pass filter. IIR digital filters are generally designed from an original analog low-pass filter by any combination of bilinear transformation, frequency transformation for analog filters and frequency transformation for IIR digital filters. Subsection 3.2.4 studies some conditions for designing IIR digital filters directly from analog low-pass filters. Subsection 3.2.5 shows some design examples of IIR digital filters which are designed directly from an analog low-pass filter.

The derived automatic design procedures still require a large number of operations, which include a lot of binomial coefficients. Here, it is mentioned that in computing the elements of the bilinear transformation matrix, a large number of operations are also required. To decrease the number of operations, Bose [20] studied properties of the bilinear transformation matrix. Similarly such properties can also be derived for frequency transformation matrices.

Section 3.3 studies the properties of frequency transformation matrices for IIR digital filters, and also develops some fast algorithms to compute the elements of the matrices.

Subsection 3.3.1 studies the properties of the matrices, and fast algorithms are developed in Subsection 3.3.2. Subsection 3.3.3 evaluates the fast algorithms in comparison to the definitions of matrices.

Finally, Section 3.4 summarizes this chapter.

3.2 Frequency Transformation Matrices

This section derives the explicit formulae of matrices which represent frequency transformations and bilinear transformation.

3.2.1 Frequency Transformation Matrices for Analog Filters

In the following discussion, frequency means angular frequency, and $\left\langle \begin{array}{c} a \\ b \end{array} \right\rangle$ denotes

$$\left\langle \begin{array}{c} a \\ b \end{array} \right\rangle := \begin{cases} 0, & b > a \quad \text{or} \quad b < 0 \quad \text{or} \\ & \quad b \text{ is not an integer}, \\ \dfrac{a!}{b!(a-b)!}, & \text{otherwise.} \end{cases}$$

Let the cutoff frequency of the original analog low-pass filter be $\Omega_0 = 1$, which is fixed throughout this investigation. The transfer function of this filter can be written as

$$H_1(s) := \frac{a_0 + a_1 s + \cdots + a_m s^m}{b_0 + b_1 s + \cdots + b_n s^n}, \quad (m \leq n).$$

The transformation formulae [30, p.258] of the frequency transformation for analog filters are shown in Table 3.1.

The transfer functions of the derived analog low-pass and high-pass filters can be written as

$$H_2(s) = \frac{c_0 + c_1 s + \cdots + c_n s^n}{d_0 + d_1 s + \cdots + d_n s^n}.$$

Table 3.1: Transformation formulae for frequency transformation of analog filters.

Filter type Cutoff frequency	Transformation formulae Parameters
Low-pass Ω_1	$s = \dfrac{s}{\Omega_1}$
High-pass Ω_1	$s = \dfrac{\Omega_1}{s}$
Band-pass Ω_1, Ω_2 $(\Omega_1 < \Omega_2)$	$s = \dfrac{s^2 + \Omega_a^2}{\Omega_b s}$ $\Omega_a = \sqrt{\Omega_1 \Omega_2}$, $\Omega_b = \Omega_2 - \Omega_1$
Band-stop Ω_1, Ω_2 $(\Omega_1 < \Omega_2)$	$s = \dfrac{\Omega_b s}{s^2 + \Omega_a^2}$ $\Omega_a = \sqrt{\Omega_1 \Omega_2}$, $\Omega_b = \Omega_2 - \Omega_1$

Those of the derived analog band-pass and band-stop filters can be written as

$$\tilde{H}_2(s) = \frac{c_0 + c_1 s + \cdots + c_n s^n + \cdots + c_{2n} s^{2n}}{d_0 + d_1 s + \cdots + d_n s^n + \cdots + d_{2n} s^{2n}}.$$

Then, matrices which transform the transfer function of the original low-pass filter to transfer functions of the derived filters are formulated as follows.

(a) Design of analog low-pass filters

The analog low-pass filter with the cutoff frequency Ω_1 is designed from the original analog low-pass filter. From the transformation formula in Table 3.1, the relation of

coefficients from $\{a_i\}_{i=0}^m$ and $\{b_i\}_{i=0}^n$ into $\{c_i\}_{i=0}^n$ and $\{d_i\}_{i=0}^n$ can easily be written as

$$c_i = \begin{cases} (\Omega_1)^{-i} a_i, & i = 0, 1, \ldots, m, \\ 0, & i = m+1, m+2, \ldots, n, \end{cases}$$
$$d_i = (\Omega_1)^{-i} b_i.$$

On the other hand, we define the following two matrices

$$\boldsymbol{X}_{nm}^{\mathrm{LP}} := \begin{bmatrix} \boldsymbol{I}_{m+1} \\ \boldsymbol{O}_{n-m,m+1} \end{bmatrix} \in \boldsymbol{R}^{(n+1)\times(m+1)},$$
$$\boldsymbol{S}_n^{\mathrm{LP}} := \begin{bmatrix} 1 & & & 0 \\ & (\Omega_1)^{-1} & & \\ & & \ddots & \\ 0 & & & (\Omega_1)^{-n} \end{bmatrix} \in \boldsymbol{R}^{(n+1)\times(n+1)},$$

where $\boldsymbol{I}_{m+1}$ is an identity matrix of dimensions $(m+1) \times (m+1)$, and $\boldsymbol{O}_{n-m,m+1}$ is a zero matrix of dimensions $(n-m) \times (m+1)$. Then, the relation of coefficient vectors can be rewritten as

$$\begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix} = \boldsymbol{X}_{nm}^{\mathrm{LP}} \boldsymbol{S}_m^{\mathrm{LP}} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{bmatrix}, \quad \begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_n \end{bmatrix} = \boldsymbol{S}_n^{\mathrm{LP}} \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{bmatrix}.$$

It remarks that the matrix $\boldsymbol{X}_{nm}^{\mathrm{LP}}$ equals to $\boldsymbol{I}_{m+1}$ in the case $m = n$.

(b) Analog high-pass filters

The analog high-pass filter with the cutoff frequency Ω_1 is designed from the original analog low-pass filter. From the transformation formula in Table 3.1, the relation of coefficients from $\{a_i\}_{i=0}^m$ and $\{b_i\}_{i=0}^n$ into $\{c_i\}_{i=0}^n$ and $\{d_i\}_{i=0}^n$ can easily be written as

$$c_i = \begin{cases} 0, & i = 0, 1, \ldots, n-m-1, \\ (\Omega_1)^{n-i} a_{n-i}, & i = n-m, n-m+1, \ldots, n, \end{cases}$$
$$d_i = (\Omega_1)^{n-i} b_{n-i}.$$

Then, the matrices which transform those coefficients can be obtained in the same manner as (a), *i.e.*,

$$\boldsymbol{X}_{nm}^{\mathrm{HP}} := \begin{bmatrix} \boldsymbol{O}_{n-m,m+1} \\ \boldsymbol{I}_{m+1} \end{bmatrix} \in \boldsymbol{R}^{(n+1)\times(m+1)},$$

$$\boldsymbol{S}_{n}^{\mathrm{HP}} := \begin{bmatrix} 0 & & (\Omega_1)^n \\ & \iddots & \\ & \Omega_1 & \\ 1 & & 0 \end{bmatrix} \in \boldsymbol{R}^{(n+1)\times(n+1)}.$$

The matrix $\boldsymbol{X}_{nm}^{\mathrm{HP}}$ equals $\boldsymbol{I}_{m+1}$ in the case $m = n$.

(c) *Analog band-pass filters*

The analog band-pass filter with the cutoff frequencies Ω_1, Ω_2 $(\Omega_1 < \Omega_2)$ is designed from the original analog low-pass filter. The matrices

$$\boldsymbol{X}_{nm}^{\mathrm{BP}} := (\Omega_b)^{n-m} \begin{bmatrix} \boldsymbol{O}_{n-m,2m+1} \\ \boldsymbol{I}_{2m+1} \\ \boldsymbol{O}_{n-m,2m+1} \end{bmatrix} \in \boldsymbol{R}^{(2n+1)\times(2m+1)},$$

$$\boldsymbol{S}_{n}^{\mathrm{BP}} := \left[s_{ij}^{\mathrm{BP}} \right] \in \boldsymbol{R}^{(2n+1)\times(n+1)},$$

where

$$s_{ij}^{\mathrm{BP}} = \left\langle \begin{matrix} i \\ (i+j-n)/2 \end{matrix} \right\rangle (\Omega_a)^{j-i+n} (\Omega_b)^{n-j},$$

transform $\{a_i\}_{i=0}^{m}$ and $\{b_i\}_{i=0}^{n}$ into $\{c_i\}_{i=0}^{2n}$ and $\{d_i\}_{i=0}^{2n}$ as

$$\begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \\ \vdots \\ c_{2n} \end{bmatrix} = \boldsymbol{X}_{nm}^{\mathrm{BP}} \boldsymbol{S}_{m}^{\mathrm{BP}} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{bmatrix}, \quad \begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_n \\ \vdots \\ d_{2n} \end{bmatrix} = \boldsymbol{S}_{n}^{\mathrm{BP}} \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{bmatrix}.$$

The matrix $\boldsymbol{X}_{nm}^{\mathrm{BP}}$ equals $\boldsymbol{I}_{2m+1}$ in the case $m = n$.

(*d*) *Analog band-stop filters*

The analog band-stop filter with the cutoff frequencies Ω_1, Ω_2 $(\Omega_1 < \Omega_2)$ is designed from the original analog low-pass filter. The matrices which transform $\{a_i\}_{i=0}^{m}$ and $\{b_i\}_{i=0}^{n}$ into $\{c_i\}_{i=0}^{2n}$ and $\{d_i\}_{i=0}^{2n}$ can be obtained in the same manner as (*c*), *i.e.*,

$$\boldsymbol{X}_{nm}^{\mathrm{BS}} := \left[x_{ij}^{\mathrm{BS}}\right] \in \boldsymbol{R}^{(2n+1)\times(2m+1)},$$

$$\boldsymbol{S}_{n}^{\mathrm{BS}} := \left[s_{ij}^{\mathrm{BS}}\right] \in \boldsymbol{R}^{(2n+1)\times(n+1)},$$

where

$$x_{ij}^{\mathrm{BS}} = \left\langle \begin{array}{c} n-m \\ (i-j)/2 \end{array} \right\rangle (\Omega_a)^{2n-2m+j-i},$$

$$s_{ij}^{\mathrm{BS}} = s_{i,n-j}^{\mathrm{BP}} = \left\langle \begin{array}{c} n-j \\ (i-j)/2 \end{array} \right\rangle (\Omega_a)^{2n-i+j}(\Omega_b)^{j}.$$

3.2.2 Frequency Transformation Matrices for IIR Digital Filters

This section derives the explicit formulae for matrices which represent the frequency transformation for IIR digital filters.

Let T denote the sampling interval of digital filters, and let the cutoff frequency of the original digital low-pass filter be ω_0. Assume that these IIR digital filters are derived from analog filters by the bilinear transformation. Then the transfer functions of the original digital low-pass filter can be obtained from [25] as

$$H_3(z) := \frac{e_0 + e_1 z^{-1} + \cdots + e_n z^{-n}}{f_0 + f_1 z^{-1} + \cdots + f_n z^{-n}},$$

of which the order of the numerator polynomial is the same as that of the denominator, as a result of the bilinear transformation from an analog low-pass filter. The transformation

formulae [23],[30, p.260] of the frequency transformation for IIR digital filters are shown in Table 3.2.

Table 3.2: Transformation formulae for frequency transformation of IIR digital filters.

Filter type Cutoff frequency	Transformation formulae Parameters
Low-pass ω_1	$z^{-1} = \dfrac{z^{-1} - \alpha}{1 - \alpha z^{-1}}$ $\alpha = \dfrac{\sin\left(\dfrac{\omega_0 - \omega_1}{2}\right)T}{\sin\left(\dfrac{\omega_0 + \omega_1}{2}\right)T}$
High-pass ω_1	$z^{-1} = -\dfrac{z^{-1} + \tilde{\alpha}}{1 + \tilde{\alpha} z^{-1}},$ $\tilde{\alpha} = -\dfrac{\cos\left(\dfrac{\omega_0 + \omega_1}{2}\right)T}{\cos\left(\dfrac{\omega_0 - \omega_1}{2}\right)T}$
Band-pass ω_1, ω_2 $(\omega_1 < \omega_2)$	$z^{-1} = -\dfrac{z^{-2} + vz^{-1} + u}{uz^{-2} + vz^{-1} + 1}$ $u = \dfrac{k-1}{k+1}, \quad k = \cot\left(\dfrac{\omega_2 - \omega_1}{2}\right)T \tan\dfrac{\omega_0}{2}T,$ $v = -\dfrac{2\alpha k}{k+1}, \quad \alpha = \dfrac{\cos\left(\dfrac{\omega_2 + \omega_1}{2}\right)T}{\cos\left(\dfrac{\omega_2 - \omega_1}{2}\right)T}$
Band-stop ω_1, ω_2 $(\omega_1 < \omega_2)$	$z^{-1} = \dfrac{z^{-2} + \tilde{v}z^{-1} + \tilde{u}}{\tilde{u}z^{-2} + \tilde{v}z^{-1} + 1}$ $\tilde{u} = -\dfrac{\tilde{k}-1}{\tilde{k}+1}, \quad \tilde{k} = \tan\left(\dfrac{\omega_2 - \omega_1}{2}\right)T \tan\dfrac{\omega_0}{2}T,$ $\tilde{v} = -\dfrac{2\tilde{\alpha}}{\tilde{k}+1}, \quad \tilde{\alpha} = \dfrac{\cos\left(\dfrac{\omega_2 + \omega_1}{2}\right)T}{\cos\left(\dfrac{\omega_2 - \omega_1}{2}\right)T}$

The transfer functions of the derived digital low-pass and high-pass filters can be reduced to

$$H_4(z) = \frac{g_0 + g_1 z^{-1} + \cdots + g_n z^{-n}}{h_0 + h_1 z^{-1} + \cdots + h_n z^{-n}}.$$

Those transfer functions of the derived digital band-pass and band-stop filters can be reduced to

$$\tilde{H}_4(z) = \frac{g_0 + g_1 z^{-1} + \cdots + g_n z^{-n} + \cdots + g_{2n} z^{-2n}}{h_0 + h_1 z^{-1} + \cdots + h_n z^{-n} + \cdots + h_{2n} z^{-2n}}.$$

Then, matrices which transform the transfer function of the original low-pass filter to transfer functions of the derived filters are formulated as follows.

(*a*) *Design of IIR digital filters of low-pass type*

The digital low-pass filter with the cutoff frequency ω_1 is designed from the original digital low-pass filter. The matrix

$$\boldsymbol{T}_n^{\mathrm{LP}} := \left[t_{ij}^{\mathrm{LP}}\right] \in \boldsymbol{R}^{(n+1)\times(n+1)},$$

where

$$t_{ij}^{\mathrm{LP}} = \sum_{k=0}^{i} (-1)^{j-i} \left\langle \begin{array}{c} j \\ i-k \end{array} \right\rangle \left\langle \begin{array}{c} n-j \\ k \end{array} \right\rangle \alpha^{j-i+2k},$$

transforms $\{e_i\}_{i=0}^n$ and $\{f_i\}_{i=0}^n$ into $\{g_i\}_{i=0}^n$ and $\{h_i\}_{i=0}^n$ as

$$\begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_n \end{bmatrix} = \boldsymbol{T}_n^{\mathrm{LP}} \begin{bmatrix} e_0 \\ e_1 \\ \vdots \\ e_n \end{bmatrix}, \quad \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_n \end{bmatrix} = \boldsymbol{T}_n^{\mathrm{LP}} \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{bmatrix}.$$

(*b*) *IIR Digital filters of high-pass type*

The digital high-pass filter with the cutoff frequency ω_1 is designed from the original digital low-pass filter. The matrix which transforms $\{e_i\}_{i=0}^n$ and $\{f_i\}_{i=0}^n$ into $\{g_i\}_{i=0}^n$ and

$\{h_i\}_{i=0}^{n}$ can be obtained in the same manner as (a), *i.e.*,

$$\begin{aligned}\boldsymbol{T}_n^{\mathrm{HP}} &:= \begin{bmatrix} 1 & & & 0 \\ & -1 & & \\ & & \ddots & \\ 0 & & & (-1)^n \end{bmatrix} \boldsymbol{T}_n^{\mathrm{LP}}\Big|_{\alpha=\tilde{\alpha}} \\ &= \left[t_{ij}^{\mathrm{HP}}\right] \in \boldsymbol{R}^{(n+1)\times(n+1)},\end{aligned}$$

where

$$\begin{aligned}t_{ij}^{\mathrm{HP}} &= (-1)^i t_{ij}^{\mathrm{LP}}\Big|_{\alpha=\tilde{\alpha}} \\ &= \sum_{k=0}^{i}(-1)^j \left\langle \begin{matrix} j \\ i-k \end{matrix} \right\rangle \left\langle \begin{matrix} n-j \\ k \end{matrix} \right\rangle \tilde{\alpha}^{j-i+2k}.\end{aligned}$$

(c) *IIR Digital filters of band-pass type*

The digital band-pass filter with the cutoff frequencies $\omega_1, \omega_2(\omega_1 < \omega_2)$ is designed from the original digital low-pass filter. The matrix

$$\boldsymbol{T}_n^{\mathrm{BP}} := \left[t_{ij}^{\mathrm{BP}}\right] \in \boldsymbol{R}^{(2n+1)\times(n+1)},$$

where

$$\begin{aligned}t_{ij}^{\mathrm{BP}} &= \sum_{\ell=0}^{i}(-1)^j \beta_{i-\ell,j}\gamma_{\ell,j}, \\ \beta_{\ell,j} &= \sum_{k=0}^{\lceil \ell/2 \rceil} \left\langle \begin{matrix} j \\ \ell-k \end{matrix} \right\rangle \left\langle \begin{matrix} \ell-k \\ k \end{matrix} \right\rangle u^{j-\ell+k} v^{\ell-2k}, \\ \gamma_{\ell,j} &= \sum_{k=0}^{\lceil \ell/2 \rceil} \left\langle \begin{matrix} n-j \\ \ell-k \end{matrix} \right\rangle \left\langle \begin{matrix} \ell-k \\ k \end{matrix} \right\rangle u^{k} v^{\ell-2k},\end{aligned}$$

transforms $\{e_i\}_{i=0}^{n}$ and $\{f_i\}_{i=0}^{n}$ into $\{g_i\}_{i=0}^{2n}$ and $\{h_i\}_{i=0}^{2n}$ as

$$\begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_n \\ \vdots \\ g_{2n} \end{bmatrix} = \boldsymbol{T}_n^{\mathrm{BP}} \begin{bmatrix} e_0 \\ e_1 \\ \vdots \\ e_n \end{bmatrix}, \quad \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_n \\ \vdots \\ h_{2n} \end{bmatrix} = \boldsymbol{T}_n^{\mathrm{BP}} \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{bmatrix}.$$

In the above formulation of $\beta_{\ell,j}$ and $\gamma_{\ell,j}$, $[\ell/2]$ means the maximum integer not exceeding $\ell/2$.

(d) IIR Digital filters of band-stop type

The digital band-stop filter with the cutoff frequencies $\omega_1, \omega_2 (\omega_1 < \omega_2)$ is designed from the original digital low-pass filter. The matrix which transforms $\{e_i\}_{i=0}^{n}$ and $\{f_i\}_{i=0}^{n}$ into $\{g_i\}_{i=0}^{2n}$ and $\{h_i\}_{i=0}^{2n}$ can be obtained in the same manner as *(c)*, *i.e.*,

$$\begin{aligned} \boldsymbol{T}_n^{\mathrm{BS}} &:= \boldsymbol{T}_n^{\mathrm{BP}}\Big|_{u=\tilde{u},v=\tilde{v}} \begin{bmatrix} 1 & & & 0 \\ & -1 & & \\ & & \ddots & \\ 0 & & & (-1)^n \end{bmatrix} \\ &= \left[t_{ij}^{\mathrm{BS}}\right] \in \boldsymbol{R}^{(2n+1)\times(n+1)}, \end{aligned}$$

where

$$\begin{aligned} t_{ij}^{\mathrm{BS}} &= (-1)^j t_{ij}^{\mathrm{BP}}\Big|_{u=\tilde{u},v=\tilde{v}} \\ &= \sum_{\ell=0,}^{i} \beta_{i-\ell,j}\gamma_{\ell,j}\Big|_{u=\tilde{u},v=\tilde{v}}. \end{aligned}$$

3.2.3 Bilinear Transformation Matrix for Rational Polynomials

Next, the transfer matrices are derived which represent the bilinear transformation for rational polynomials.

The transfer function of an analog filter is written as

$$H_1(s) := \frac{a_0 + a_1 s + \cdots + a_m s^m}{b_0 + b_1 s + \cdots + b_n s^n}, \quad (m \leq n).$$

The transformation formula [30, pp.219–224] of the bilinear transformation is given as

$$s = \frac{2}{T} \cdot \frac{1 - z^{-1}}{1 + z^{-1}}.$$

In designing filters, this is equivalent [25] to

$$s = \frac{1 - z^{-1}}{1 + z^{-1}}.$$

Then the transfer function of the derived digital filter can be reduced to

$$H_3(z) = \frac{e_0 + e_1 z^{-1} + \cdots + e_n z^{-n}}{f_0 + f_1 z^{-1} + \cdots + f_n z^{-n}}.$$

Then the matrices:

$$\boldsymbol{Y}_{nm} := [y_{ij}] \in \boldsymbol{R}^{(n+1)\times(m+1)},$$

$$\boldsymbol{Q}_n := [q_{ij}] \in \boldsymbol{R}^{(n+1)\times(n+1)},$$

where

$$y_{ij} = \left\langle \begin{array}{c} n-m \\ i-j \end{array} \right\rangle,$$

$$q_{ij} = \sum_{k=0}^{i} (-1)^{i-k} \left\langle \begin{array}{c} j \\ i-k \end{array} \right\rangle \left\langle \begin{array}{c} n-j \\ k \end{array} \right\rangle.$$

transform $\{a_i\}_{i=0}^m$ and $\{b_i\}_{i=0}^n$ into $\{e_i\}_{i=0}^n$ and $\{f_i\}_{i=0}^n$ as

$$\begin{bmatrix} e_0 \\ e_1 \\ \vdots \\ e_n \end{bmatrix} = \boldsymbol{Y}_{nm} \boldsymbol{Q}_m \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{bmatrix}, \quad \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{bmatrix} = \boldsymbol{Q}_n \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{bmatrix}.$$

The matrix $\boldsymbol{Y}_{nm}$ equals $\boldsymbol{I}_{m+1}$ in the case $m = n$. Furthermore, the recursive formula to obtain $\boldsymbol{Q}_n$ is shown in [20].

3.2.4 Conditions for Direct Design of IIR Digital Filters from an Analog Low-pass Filter

In this section, conditions to design IIR digital filters directly from an analog Low-pass filter are investigated.

In designing an IIR digital filter, two different methods can be considered as shown in Fig.3.1, *i.e.*,

(i) Frequency Transformation for Analog Filters first, and then Bilinear Transformation.

(ii) Bilinear Transformation first, and then Frequency Transformation for IIR Digital Filters.

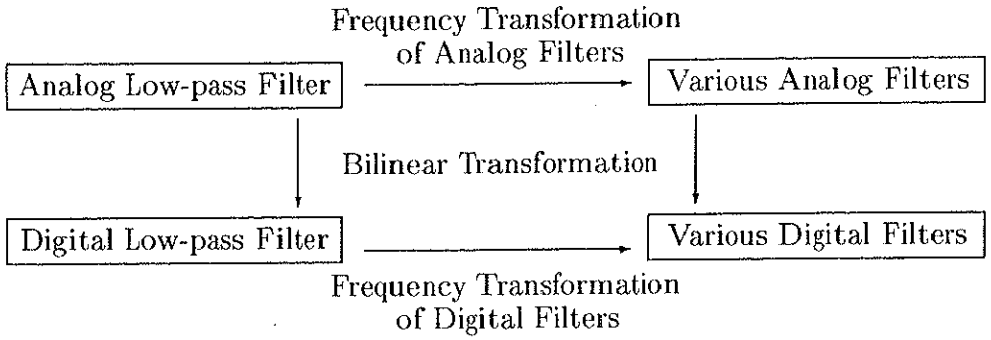


Figure 3.1: The design process for IIR digital filters.

Though they look quite different from each other at a glance, their transformation formulae is exactly the same so that the same digital filter is designed [30, p.261]. Hence, we can choose simpler one.

Since the frequency transformation formulae to design analog low-pass and high-pass filters can be simply defined, it appears more simple to design digital low-pass and high-pass filters by the method (i) than by the method (ii). It is not known which method is simpler to design digital band-pass and band-stop filters.

To design digital low-pass and high-pass filters with cutoff frequency ω_1 by the method **(i)**, the cutoff frequency Ω_1 of the analog filters must be

$$\Omega_1 = \tan\frac{\omega_1}{2}T.$$

To design band-pass and band-stop digital filters with cutoff frequencies ω_1, ω_2 $(\omega_1 < \omega_2)$ by the method **(i)**, the cutoff frequencies Ω_1, Ω_2 of the analog filters must be

$$\Omega_1 = \tan\frac{\omega_1}{2}T, \quad \Omega_2 = \tan\frac{\omega_2}{2}T.$$

To design any type of IIR digital filters by the method (ii), the cutoff frequency ω_0 of the original digital low-pass filter must be

$$\omega_0 = \frac{\pi}{2T}.$$

3.2.5 Design Examples of IIR Digital Filters

Some simple digital filters are designed to evaluate the effectiveness of the proposed matrices. By using the matrices, transfer functions of IIR digital filters of various types are derived directly and automatically from a transfer function of an analog low-pass filter.

(*a*) *Original analog filter of low-pass type*

Let the original analog low-pass filter be a second-order normalized Butterworth low-pass filter. The transfer function is given as

$$H_1(s) := \frac{1}{1 + \sqrt{2}s + s^2}.$$

Coefficient vectors are

$$[a_0] = [1], \quad \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} 1 \\ \sqrt{2} \\ 1 \end{bmatrix}.$$

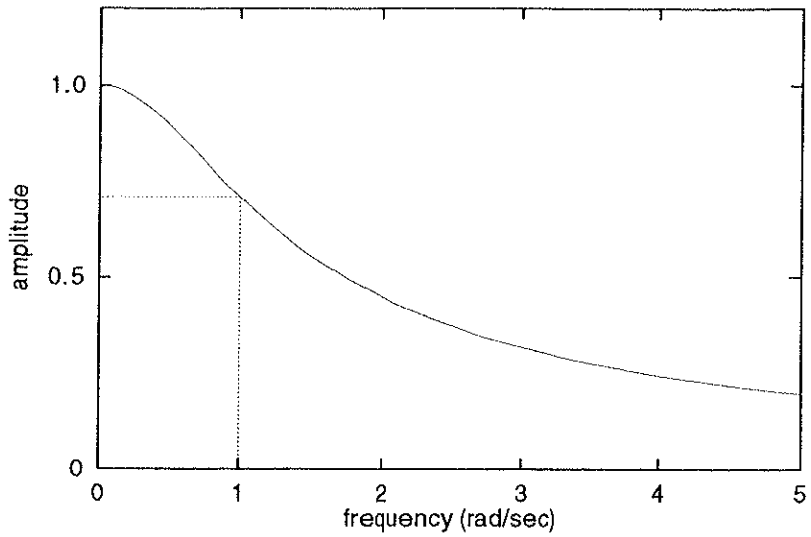


Figure 3.2: Amplitude characteristics of the original analog low-pass filter.

The amplitude characteristics of this filter is shown in Fig.3.2.

(b) Design of IIR digital filters of low-pass type

The digital low-pass filter with the cutoff frequency $\omega_1 = 0.2\pi/T$ is designed from the original analog low-pass filter. This filter is designed by the method **(i)**, which is the simpler method.

The cutoff frequency of the analog low-pass filter, which is derived by the frequency transformation for analog filters, is given as

$$\Omega_1 = \tan\frac{\omega_1}{2}T = \tan 0.1\pi \simeq 0.3249.$$

The transfer matrices $\boldsymbol{Q}_2, \boldsymbol{X}_{2,0}^{\mathrm{LP}}, \boldsymbol{S}_0^{\mathrm{LP}}$ and $\boldsymbol{S}_2^{\mathrm{LP}}$ are given as

$$\boldsymbol{Q}_2 = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 0 & -2 \\ 1 & -1 & 1 \end{bmatrix}, \quad \boldsymbol{X}_{2,0}^{\mathrm{LP}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix},$$

$$\boldsymbol{S}_0^{\mathrm{LP}} = [1], \quad \boldsymbol{S}_2^{\mathrm{LP}} = \begin{bmatrix} 1.0000 & 0.0000 & 0.0000 \\ 0.0000 & 3.0779 & 0.0000 \\ 0.0000 & 0.0000 & 9.4732 \end{bmatrix}.$$

Then the coefficient vectors of the derived digital low-pass filter are obtained as

$$\begin{bmatrix} g_0 \\ g_1 \\ g_2 \end{bmatrix} = \boldsymbol{Q}_2 \boldsymbol{X}_{2,0}^{\mathrm{LP}} \boldsymbol{S}_0^{\mathrm{LP}} [a_0] = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix},$$

$$\begin{bmatrix} h_0 \\ h_1 \\ h_2 \end{bmatrix} = \boldsymbol{Q}_2 \boldsymbol{S}_2^{\mathrm{LP}} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} 14.8260 \\ -16.9464 \\ 6.1204 \end{bmatrix}.$$

The derived transfer function of this filter is written as

$$H_4(z) = \frac{1 + 2z^{-1} + z^{-2}}{14.8260 - 16.9464z^{-1} + 6.1204z^{-2}}.$$

The amplitude characteristics of this filter are shown in Fig.3.3. The designed digital filter realizes low-pass characteristics.

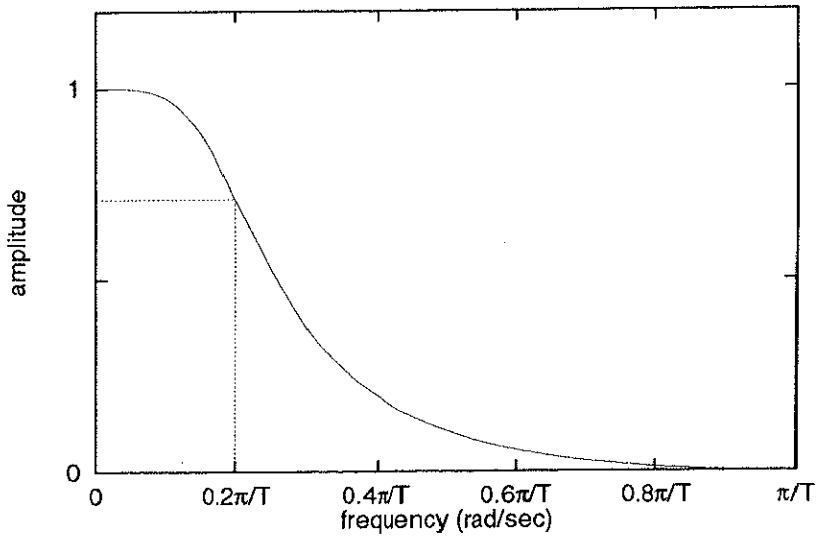


Figure 3.3: Amplitude characteristics of the derived digital low-pass filter.

(b) Design of IIR digital filter of high-pass type

The digital high-pass filter with the cutoff frequency $\omega_1 = 0.2\pi/T$ is designed from the original analog low-pass filter. This filter is designed by the method (i), which is the simpler method.

The cutoff frequency of the analog high-pass filter, which is derived by the frequency transformation for analog filters, is given as

$$\Omega_1 = \tan\frac{\omega_1}{2}T = \tan 0.1\pi \simeq 0.3249.$$

The transfer matrices $\boldsymbol{Q}_2, \boldsymbol{X}_{2,0}^{\mathrm{HP}}, \boldsymbol{S}_0^{\mathrm{HP}}$ and $\boldsymbol{S}_2^{\mathrm{HP}}$ are given as

$$\boldsymbol{Q}_2 = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 0 & -2 \\ 1 & -1 & 1 \end{bmatrix}, \quad \boldsymbol{X}_{2,0}^{\mathrm{HP}} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},$$

$$\boldsymbol{S}_0^{\mathrm{HP}} = [1], \quad \boldsymbol{S}_2^{\mathrm{HP}} = \begin{bmatrix} 0.0000 & 0.0000 & 0.1056 \\ 0.0000 & 0.3249 & 0.0000 \\ 1.0000 & 0.0000 & 0.0000 \end{bmatrix}.$$

The coefficient vectors of the derived digital high-pass filter are obtained as

$$\begin{bmatrix} g_0 \\ g_1 \\ g_2 \end{bmatrix} = \boldsymbol{Q}_2\boldsymbol{X}_{2,0}^{\mathrm{HP}}\boldsymbol{S}_0^{\mathrm{HP}}[a_0] = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix},$$

$$\begin{bmatrix} h_0 \\ h_1 \\ h_2 \end{bmatrix} = \boldsymbol{Q}_2\boldsymbol{S}_2^{\mathrm{HP}} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} 1.5651 \\ -1.7888 \\ 0.6461 \end{bmatrix}.$$

The derived transfer function of this filter is written as

$$H_4(z) = \frac{1 - 2z^{-1} + z^{-2}}{1.5651 - 1.7888z^{-1} + 0.6461z^{-2}}.$$

The amplitude characteristics of this filter are shown in Fig.3.4. The designed digital filter realizes high-pass characteristics.

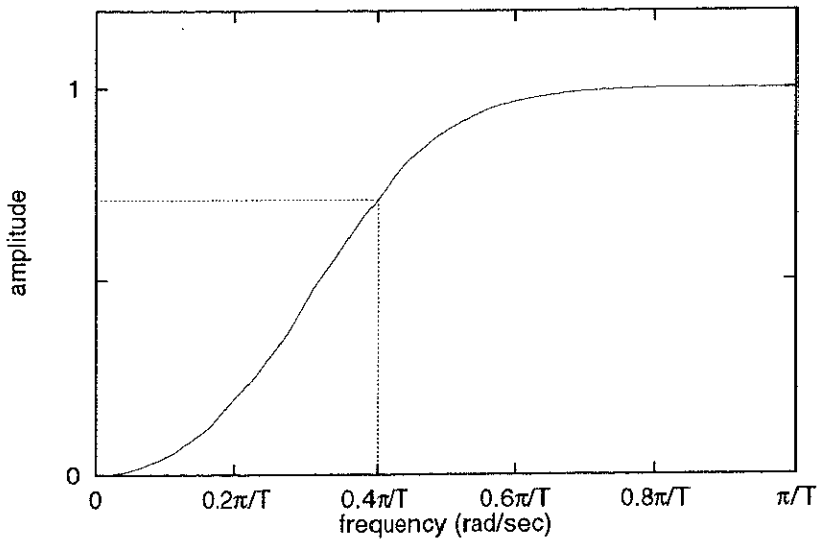


Figure 3.4: Amplitude characteristics of the derived digital high-pass filter.

(d) IIR digital filter of band-pass type

The digital band-pass filter with the cutoff frequencies $\omega_1 = 0.2\pi/T, \omega_2 = 0.6\pi/T$ is designed from the original analog low-pass filter. This filter is designed by both the methods (i) and (ii).

(d-i) the method (i)

The cutoff frequencies of the analog band-pass filter, which is derived by the frequency transformation for analog filters, are given as

$$\Omega_1 = \tan\frac{\omega_1}{2}T = \tan 0.1\pi \simeq 0.3249,$$

$$\Omega_2 = \tan\frac{\omega_2}{2}T = \tan 0.3\pi \simeq 1.3764.$$

The transfer matrices $\boldsymbol{Q}_4, \boldsymbol{X}_{2,0}^{\mathrm{BP}}, \boldsymbol{S}_0^{\mathrm{BP}}$ and $\boldsymbol{S}_2^{\mathrm{BP}}$ are given as

$$\boldsymbol{Q}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 4 & 2 & 0 & -2 & -4 \\ 6 & 0 & -2 & 0 & 6 \\ 4 & -2 & 0 & 2 & -4 \\ 1 & -1 & 1 & -1 & 1 \end{bmatrix}, \quad \boldsymbol{X}_{2,0}^{\mathrm{BP}} = \begin{bmatrix} 0.0000 \\ 0.0000 \\ 1.1056 \\ 0.0000 \\ 0.0000 \end{bmatrix},$$

$$\boldsymbol{S}_0^{\mathrm{BP}} = [1], \quad \boldsymbol{S}_2^{\mathrm{BP}} = \begin{bmatrix} 0.0000 & 0.0000 & 0.2000 \\ 0.0000 & 0.4702 & 0.0000 \\ 1.1056 & 0.0000 & 0.8944 \\ 0.0000 & 1.0515 & 0.0000 \\ 0.0000 & 0.0000 & 1.0000 \end{bmatrix}.$$

Then, the coefficient vectors of the derived digital high-pass filter are obtained as

$$\begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \\ g_4 \end{bmatrix} = \boldsymbol{Q}_4 \boldsymbol{X}_{2,0}^{\mathrm{BP}} \boldsymbol{S}_0^{\mathrm{BP}} [a_0] = \begin{bmatrix} 1.1056 \\ 0.0000 \\ -2.2112 \\ 0.0000 \\ 1.1056 \end{bmatrix},$$

$$\begin{bmatrix} h_0 \\ h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix} = \boldsymbol{Q}_4 \boldsymbol{S}_2^{\mathrm{BP}} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} 5.3520 \\ -4.8442 \\ 3.2000 \\ -1.5558 \\ 1.0480 \end{bmatrix}.$$

The derived transfer function of this filter is written as

$$\tilde{H}_4(z) = \frac{1.1056 - 2.2112z^{-2} + 1.1056z^{-4}}{5.3520 - 4.8442z^{-1} + 3.2000z^{-2} - 1.5558z^{-3} + 1.0480z^{-4}}.$$

The amplitude characteristics of this filter are shown in Fig.3.5. The designed digital filter realizes band-pass characteristics.

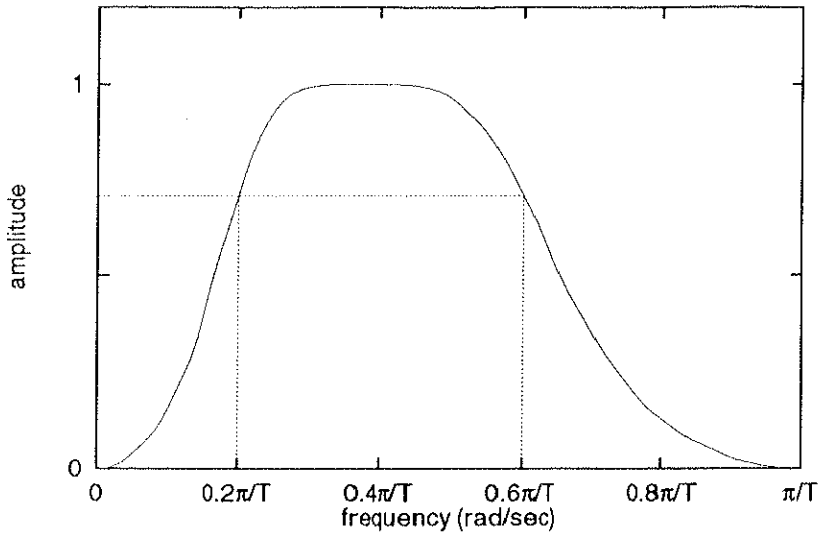


Figure 3.5: Amplitude characteristics of the derived digital band-pass filter.

(*d-ii*) *the method* (ii)

The transfer matrices $\boldsymbol{T}_2^{\mathrm{BP}}$, $\boldsymbol{Y}_{2,0}$ and $\boldsymbol{Q}_0$ are given as

$$\boldsymbol{T}_2^{\mathrm{BP}} = \begin{bmatrix} 1.0000 & -0.1584 & 0.0251 \\ -0.8850 & 0.5126 & -0.1402 \\ 0.5126 & -1.2209 & 0.5126 \\ -0.1402 & 0.5126 & -0.8850 \\ 0.0251 & -0.1584 & 1.0000 \end{bmatrix},$$

$$\boldsymbol{Y}_{2,0} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}, \quad \boldsymbol{Q}_0 = [1].$$

Then, the coefficient vectors of the derived digital high-pass filter are obtained as

$$\begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \\ g_4 \end{bmatrix} = \boldsymbol{T}_2^{\mathrm{BP}} \boldsymbol{Y}_{2,0} \boldsymbol{Q}_0 [a_0] = \begin{bmatrix} 0.7083 \\ 0.0000 \\ -1.4166 \\ 0.0000 \\ 0.7083 \end{bmatrix},$$

$$\begin{bmatrix} h_0 \\ h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix} = \boldsymbol{T}_2^{\mathrm{BP}} \boldsymbol{Q}_2 \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} 3.4289 \\ -3.1037 \\ 2.0503 \\ -0.9970 \\ 0.6713 \end{bmatrix}.$$

These coefficients are different from those obtained by the method (i), but the derived transfer functions become same. Therefore, the same digital filter was designed.

3.3 Properties of Frequency Transformation Matrices for IIR Digital Filters

3.3.1 Properties

In this section, significant properties of the frequency transformation matrices are investigated. Let $\left\langle \begin{matrix} a \\ b \end{matrix} \right\rangle$ denote that

$$\left\langle \begin{matrix} a \\ b \end{matrix} \right\rangle = \begin{cases} 0, & b > a \text{ or } b < 0, \\ \dfrac{a!}{b!(a-b)!}, & \text{otherwise}, \end{cases}$$

for any integers a and b. We recall the transfer function $H_3(z)$ of the original low-pass filter with the cutoff frequency ω_0:

$$H_3(z) = \frac{e_0 + e_1 z^{-1} + \cdots + e_n z^{-n}}{f_0 + f_1 z^{-1} + \cdots + f_n z^{-n}}, \tag{3.1}$$

of which the order of the numerator polynomial is the same as that of the denominator, as a result of the bilinear transformation from an analog low-pass filter. The transformation formulae of the frequency transformation for IIR digital filters were shown in Table 3.2.

3.3.1.1 Properties of the Matrices to Design Low-pass Filters

The matrix $\boldsymbol{T}_n^{\mathrm{LP}}$ to design low-pass filters was obtained as

$$\boldsymbol{T}_n^{\mathrm{LP}} := \boldsymbol{T}_n^{\mathrm{LP}}(\alpha) = \left[t_{i,j}^{\mathrm{LP},n}(\alpha) \right] \in \boldsymbol{R}^{(n+1)\times(n+1)},$$

$$\text{where} \quad t_{i,j}^{\mathrm{LP},n} := t_{i,j}^{\mathrm{LP},n}(\alpha) = \sum_{k=0}^{i} (-1)^{j-i} \left\langle \begin{array}{c} j \\ i-k \end{array} \right\rangle \left\langle \begin{array}{c} n-j \\ k \end{array} \right\rangle \alpha^{j-i+2k}. \tag{3.2}$$

The transfer function $H_4(z)$ of the derived low-pass filter can be represented by

$$H_4(z) = \frac{g_0 + g_1 z^{-1} + \cdots + g_n z^{-n}}{h_0 + h_1 z^{-1} + \cdots + h_n z^{-n}}. \tag{3.3}$$

Here, the relations between the coefficients of (3.1) and those of (3.3) are as follows:

$$\begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_n \end{bmatrix} = \boldsymbol{T}_n^{\mathrm{LP}} \begin{bmatrix} e_0 \\ e_1 \\ \vdots \\ e_n \end{bmatrix}, \quad \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_n \end{bmatrix} = \boldsymbol{T}_n^{\mathrm{LP}} \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{bmatrix}. \tag{3.4}$$

Since the coefficients $\{g_i\}_{i=0}^n$ of (3.3) can be obtained by (3.4), the numerator polynomial $N_4(z)$ of $H_4(z)$ can be written as

$$N_4(z) = \sum_{i=0}^{n} g_i z^{-i} = \sum_{i=0}^{n} \left(\sum_{j=0}^{n} t_{i,j}^{\mathrm{LP},n} e_j \right) z^{-i} = \sum_{j=0}^{n} \left(\sum_{i=0}^{n} t_{i,j}^{\mathrm{LP},n} z^{-i} \right) e_j. \tag{3.5}$$

On the other hand, since the transfer function $H_4(z)$ can be obtained as a result of the frequency transformation of $H_3(z)$, $N_4(z)$ can also be written as

$$\begin{aligned} N_4(z) &= \sum_{i=0}^{n} g_i z^{-i} = \left(1 - \alpha z^{-1}\right)^n \sum_{j=0}^{n} e_j \left(\frac{z^{-1} - \alpha}{1 - \alpha z^{-1}} \right)^j \\ &= \sum_{j=0}^{n} \left\{ \left(1 - \alpha z^{-1}\right)^{n-j} \left(z^{-1} - \alpha\right)^j \right\} e_j. \end{aligned} \tag{3.6}$$

From (3.5) and (3.6), we have

$$\left(1 - \alpha z^{-1}\right)^{n-j} \left(z^{-1} - \alpha\right)^j = \sum_{i=0}^{n} t_{i,j}^{\mathrm{LP},n} z^{-i}. \tag{3.7}$$

The following three propositions can be obtained.

Proposition 3.1 The elements $\left\{t_{i,j}^{\mathrm{LP},n}\right\}$ of the matrix $\boldsymbol{T}_n^{\mathrm{LP}}$ satisfy the following properties.

(i) $$t_{0,j}^{\mathrm{LP},n} = (-\alpha)^j, \quad j = 0, 1, \ldots, n, \tag{3.8}$$

(ii) $$t_{i,0}^{\mathrm{LP},n} = (-\alpha)^i \left\langle \begin{array}{c} n \\ i \end{array} \right\rangle, \quad i = 0, 1, \ldots, n, \tag{3.9}$$

(iii) $$t_{i,j}^{\mathrm{LP},n} = t_{n-i,n-j}^{\mathrm{LP},n}, \quad i, j = 0, 1, \ldots, n, \tag{3.10}$$

(iv) $$t_{n-i,j}^{\mathrm{LP},n} = (-\alpha)^n \cdot t_{i,j}^{\mathrm{LP},n}\left(\frac{1}{\alpha}\right), \quad i, j = 0, 1, \ldots, n, \tag{3.11}$$

(v) $$t_{i,j}^{\mathrm{LP},n} = t_{i-1,j-1}^{\mathrm{LP},n} + \alpha \cdot t_{i-1,j}^{\mathrm{LP},n} - \alpha \cdot t_{i,j-1}^{\mathrm{LP},n}, \quad i, j = 0, 1, \ldots, n, \tag{3.12}$$

where $$t_{i,j}^{\mathrm{LP},n} = 0, \quad i < 0 \text{ or } j < 0 \text{ or } i > n \text{ or } j > n.$$

Proof The proofs of (i) and (ii) easily follow from the definition (3.2).

Proof of (iii): Replacing j in (3.7) by $(n - j)$, we have

$$\left(1 - \alpha z^{-1}\right)^j \left(z^{-1} - \alpha\right)^{n-j} = \sum_{i=0}^{n} t_{i,n-j}^{\mathrm{LP},n} z^{-i}. \tag{3.13}$$

Next, replacing z^{-1} in (3.13) by z, and then multiplying z^{-n} on both sides, we have

$$\left(z^{-1} - \alpha\right)^j \left(1 - \alpha z^{-1}\right)^{n-j} = \sum_{i=0}^{n} t_{n-i,n-j}^{\mathrm{LP},n} z^{-i}. \tag{3.14}$$

The left side of (3.14) is equal to that of (3.7).

Proof of (iv): Replacing α in (3.7) by $1/\alpha$, and then multiplying $(-\alpha)^n$ to both sides, we have

$$\left(1 - \alpha z^{-1}\right)^j \left(z^{-1} - \alpha\right)^{n-j} = \sum_{i=0}^{n} (-\alpha)^n \cdot \left\{ t_{i,j}^{\mathrm{LP},n}\left(\frac{1}{\alpha}\right) \right\} z^{-i}. \tag{3.15}$$

The left side of (3.15) is equal to that of (3.13). Therefore, (3.11) holds.

Proof of (v): Replacing j in (3.7) by $(j - 1)$, we have

$$\left(1 - \alpha z^{-1}\right)^{n-j+1} \left(z^{-1} - \alpha\right)^{j-1} = \sum_{i=0}^{n} t_{i,j-1}^{\mathrm{LP},n} z^{-i}. \tag{3.16}$$

From (3.7) and (3.16), we have

$$(1 - \alpha z^{-1}) \sum_{i=0}^{n} t_{i,j}^{\mathrm{LP},n} z^{-i} = (z^{-1} - \alpha) \sum_{i=0}^{n} t_{i,j-1}^{\mathrm{LP},n} z^{-i}. \tag{3.17}$$

Using (3.8)–(3.10), (3.17) can be rewritten as follows:

$$\sum_{i=1}^{n}\left(t_{i,j}^{\mathrm{LP},n}-t_{i-1,j-1}^{\mathrm{LP},n}-\alpha\cdot t_{i-1,j}^{\mathrm{LP},n}+\alpha\cdot t_{i,j-1}^{\mathrm{LP},n}\right)z^{-i}=0,$$

which implies (3.12). ■

Proposition 3.2 The relations between the elements $\left\{t_{i,j}^{\mathrm{LP},n}\right\}$ of $\boldsymbol{T}_n^{\mathrm{LP}}$ and $\left\{t_{i,j}^{\mathrm{LP},n-1}\right\}$ of $\boldsymbol{T}_{n-1}^{\mathrm{LP}}$ are as follows.

$$\text{(i)}\qquad t_{i,j}^{\mathrm{LP},n}=t_{i,j}^{\mathrm{LP},n-1}-\alpha\cdot t_{i-1,j}^{\mathrm{LP},n-1},\quad i,j=0,1,\ldots,n, \tag{3.18}$$

$$\text{(ii)}\qquad t_{i,j}^{\mathrm{LP},n}=t_{i-1,j-1}^{\mathrm{LP},n-1}-\alpha\cdot t_{i,j-1}^{\mathrm{LP},n-1},\quad i,j=0,1,\ldots,n. \tag{3.19}$$

Proof Proof of (i): Replacing n in (3.7) by $(n-1)$, we have

$$\left(1-\alpha z^{-1}\right)^{n-1-j}\left(z^{-1}-\alpha\right)^{j}=\sum_{i=0}^{n-1}t_{i,j}^{n-1}z^{-i}. \tag{3.20}$$

From (3.7) and (3.20), we have

$$\begin{aligned}\sum_{i=0}^{n}t_{i,j}^{\mathrm{LP},n}z^{-i}&=\left(1-\alpha z^{-1}\right)\sum_{i=0}^{n-1}t_{i,j}^{n-1}z^{-i}\\&=\sum_{i=0}^{n-1}t_{i,j}^{n-1}z^{-i}-\alpha\sum_{i=1}^{n}t_{i-1,j}^{n-1}z^{-i}\\&=\sum_{i=0}^{n}\left(t_{i,j}^{n-1}-\alpha\cdot t_{i-1,j}^{n-1}\right)z^{-i}.\end{aligned}$$

Therefore, (3.18) holds. The proof of (ii) follows from (3.12) and (3.18). ■

Proposition 3.3 The following equalities hold.

$$\boldsymbol{T}_n^{\mathrm{LP}}(\alpha)\cdot\boldsymbol{T}_n^{\mathrm{LP}}(-\alpha)=\boldsymbol{T}_n^{\mathrm{LP}}(-\alpha)\cdot\boldsymbol{T}_n^{\mathrm{LP}}(\alpha)=(1-\alpha^2)^n\boldsymbol{I}_{n+1}, \tag{3.21}$$

where $\boldsymbol{I}_{n+1}$ is an identity matrix of dimensions $(n+1)\times(n+1)$.

Proof The matrix $\boldsymbol{T}_n^{\mathrm{LP}}(\alpha)$ describes the transformation from the coefficients of the original low-pass filter (with the cutoff frequency ω_0) into those of the derived low-pass filter (with the cutoff frequency ω_1).

Here, we consider the inverse transformation from the derived low-pass filter into the original. The transformation parameter α' in this case is given as

$$\alpha' = \frac{\sin\left(\frac{\omega_1 - \omega_0}{2}\right)T}{\sin\left(\frac{\omega_1 + \omega_0}{2}\right)T} = -\frac{\sin\left(\frac{\omega_0 - \omega_1}{2}\right)T}{\sin\left(\frac{\omega_0 + \omega_1}{2}\right)T} = -\alpha,$$

which means that the inverse transformation matrix is given by $\boldsymbol{T}_n^{\mathrm{LP}}(-\alpha)$. Hence, there exists $p \in \boldsymbol{R}$ which satisfies

$$\boldsymbol{T}_n^{\mathrm{LP}}(\alpha) \cdot \boldsymbol{T}_n^{\mathrm{LP}}(-\alpha) = \boldsymbol{T}_n^{\mathrm{LP}}(-\alpha) \cdot \boldsymbol{T}_n^{\mathrm{LP}}(\alpha) = p \cdot \boldsymbol{I}_{n+1}. \tag{3.22}$$

The parameter p in (3.22) can be obtained by the multiplication of the first-row elements of $\boldsymbol{T}_n^{\mathrm{LP}}(\alpha)$ and the first-column elements of $\boldsymbol{T}_n^{\mathrm{LP}}(-\alpha)$, *i.e.*,

$$p = \sum_{i=0}^{n} \left\{ t_{0,i}^{\mathrm{LP},n}(\alpha) \cdot t_{i,0}^{\mathrm{LP},n}(-\alpha) \right\} = \sum_{i=0}^{n} (-\alpha)^i \cdot \alpha^i \left\langle \begin{array}{c} n \\ i \end{array} \right\rangle$$

$$= \sum_{i=0}^{n} (-\alpha^2)^i \left\langle \begin{array}{c} n \\ i \end{array} \right\rangle = (1 - \alpha^2)^n.$$

Therefore, (3.21) holds. ■

3.3.1.2 Properties of the Matrices to Design High-pass Filters

The transformation formula to design high-pass filters can be obtained from the formula to design low-pass filters by replacing $\alpha, H_0(z)$ with $-\tilde{\alpha}, H_0(-z)$, respectively. From this, the following properties of the high-pass design matrix $\boldsymbol{T}_n^{\mathrm{HP}}$ are similarly derived.

Proposition 3.4 The elements $\left\{t_{i,j}^{\mathrm{HP},n}\right\}$ of the matrix $\boldsymbol{T}_n^{\mathrm{HP}}$ satisfy the following properties.

(i) $$t_{0,j}^{\mathrm{HP},n} = (-\tilde{\alpha})^j, \quad j = 0, 1, \ldots, n,$$

(ii) $$t_{i,0}^{\mathrm{HP},n} = \tilde{\alpha}^i \left\langle \begin{array}{c} n \\ i \end{array} \right\rangle, \quad i = 0, 1, \ldots, n,$$

(iii) $$t_{i,j}^{\mathrm{HP},n} = t_{n-i,n-j}^{\mathrm{HP},n}, \quad i, j = 0, 1, \ldots, n,$$

(iv) $$t_{n-i,j}^{\mathrm{HP},n} = (-\tilde{\alpha})^n \cdot t_{i,j}^{\mathrm{HP},n}\left(\frac{1}{\tilde{\alpha}}\right), \quad i, j = 0, 1, \ldots, n,$$

(v) $$t_{i,j}^{\mathrm{HP},n} = -\left(t_{i-1,j-1}^{\mathrm{HP},n} + \tilde{\alpha} \cdot t_{i-1,j}^{\mathrm{HP},n} + \tilde{\alpha} \cdot t_{i,j-1}^{\mathrm{HP},n}\right), \quad i, j = 0, 1, \ldots, n,$$

where $$t_{i,j}^{\mathrm{HP},n} = 0, \quad i < 0 \text{ or } j < 0 \text{ or } i > n \text{ or } j > n.$$ ■

Proposition 3.5 The relations between the elements $\left\{t_{i,j}^{\mathrm{HP},n}\right\}$ of $\boldsymbol{T}_n^{\mathrm{HP}}$ and $\left\{t_{i,j}^{\mathrm{HP},n-1}\right\}$ of $\boldsymbol{T}_{n-1}^{\mathrm{HP}}$ are as follows.

(i) $$t_{i,j}^{\mathrm{HP},n} = t_{i,j}^{\mathrm{HP},n-1} + \tilde{\alpha} \cdot t_{i-1,j}^{\mathrm{HP},n-1}, \quad i, j = 0, 1, \ldots, n,$$

(ii) $$t_{i,j}^{\mathrm{HP},n} = t_{i-1,j-1}^{\mathrm{HP},n-1} + \tilde{\alpha} \cdot t_{i,j-1}^{\mathrm{HP},n-1}, \quad i, j = 0, 1, \ldots, n.$$ ■

Proposition 3.6 A square of the matrix $\boldsymbol{T}_n^{\mathrm{HP}}(\tilde{\alpha})$ equals $(1 - \tilde{\alpha}^2)^n$ times an identity matrix, that is,

$$\boldsymbol{T}_n^{\mathrm{HP}}(\tilde{\alpha}) \cdot \boldsymbol{T}_n^{\mathrm{HP}}(\tilde{\alpha}) = (1 - \tilde{\alpha}^2)^n \boldsymbol{I}_{n+1}.$$

Proof The matrix $\boldsymbol{T}_n^{\mathrm{HP}}(\tilde{\alpha})$ describes the transformation from the coefficients of the original low-pass filter (with the cutoff frequency ω_0) into those of the derived high-pass filter (with the cutoff frequency ω_1).

Here, we consider the inverse transformation from the derived high-pass filter into the original low-pass filter. The transformation parameter $\tilde{\alpha}'$ of this case is given as

$$\tilde{\alpha}' = -\frac{\cos\left(\frac{\omega_1+\omega_0}{2}\right)T}{\cos\left(\frac{\omega_1-\omega_0}{2}\right)T} = -\frac{\cos\left(\frac{\omega_0+\omega_1}{2}\right)T}{\cos\left(\frac{\omega_0-\omega_1}{2}\right)T} = \tilde{\alpha},$$

which means that the inverse transformation matrix is also given by $\boldsymbol{T}_n^{\mathrm{HP}}(\tilde{\alpha})$. ■

3.3.1.3 Properties of the Matrices to Design Band-pass Filters

The matrix $\boldsymbol{T}_n^{\mathrm{BP}}$ for designing band-pass filters was obtained as

$$\boldsymbol{T}_n^{\mathrm{BP}} := \left[t_{i,j}^{\mathrm{BP},n}(u,v)\right] \in \boldsymbol{R}^{(2n+1)\times(n+1)},$$

$$\text{where} \quad t_{i,j}^{\mathrm{BP},n} := t_{i,j}^{\mathrm{BP},n}(u,v) = \sum_{\ell=0}^{i}(-1)^j \beta_{i-\ell,j}\gamma_{\ell,j}, \tag{3.23}$$

$$\beta_{\ell,j} = \sum_{k=0}^{\lfloor \ell/2 \rfloor} \left\langle \begin{matrix} j \\ \ell-k \end{matrix} \right\rangle \left\langle \begin{matrix} \ell-k \\ k \end{matrix} \right\rangle u^{j-\ell+k} v^{\ell-2k}, \tag{3.24}$$

$$\gamma_{\ell,j} = \sum_{k=0}^{\lfloor \ell/2 \rfloor} \left\langle \begin{matrix} n-j \\ \ell-k \end{matrix} \right\rangle \left\langle \begin{matrix} \ell-k \\ k \end{matrix} \right\rangle u^{k} v^{\ell-2k}. \tag{3.25}$$

The transfer function $\tilde{H}_4(z)$ of the derived filter was written as

$$\tilde{H}_4(z) = \frac{g_0 + g_1 z^{-1} + \cdots + g_n z^{-n} + \cdots + g_{2n} z^{-2n}}{h_0 + h_1 z^{-1} + \cdots + h_n z^{-n} + \cdots + h_{2n} z^{-2n}}. \tag{3.26}$$

Here, the relations between the coefficients of (3.1) and those of (3.26) are as follows:

$$\begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_n \\ \vdots \\ e_{2n} \end{bmatrix} = \boldsymbol{T}_n^{\mathrm{BP}} \begin{bmatrix} e_0 \\ e_1 \\ \vdots \\ e_n \end{bmatrix}, \quad \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_n \\ \vdots \\ f_{2n} \end{bmatrix} = \boldsymbol{T}_n^{\mathrm{BP}} \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{bmatrix}. \tag{3.27}$$

Since the coefficients $\{g_i\}_{i=0}^{2n}$ of (3.26) can be obtained by (3.27), the numerator poly-

nomial $\tilde{N}_4(z)$ of $\tilde{H}_4(z)$ can be written as

$$\tilde{N}_4(z) = \sum_{i=0}^{2n} g_i z^{-i} = \sum_{i=0}^{2n} \left(\sum_{j=0}^{n} t_{i,j}^{\mathrm{BP},n} e_j \right) z^{-i} = \sum_{j=0}^{n} \left(\sum_{i=0}^{2n} t_{i,j}^{\mathrm{BP},n} z^{-i} \right) e_j. \tag{3.28}$$

On the other hand, since the transfer function $\tilde{H}_4(z)$ can be obtained as a result of the frequency transformation of $H_3(z)$, $\tilde{N}_4(z)$ can also be written as

$$\begin{aligned} \tilde{N}_4(z) &= \sum_{i=0}^{2n} g_i z^{-i} \\ &= \left(uz^{-2} + vz^{-1} + 1 \right)^n \sum_{j=0}^{n} e_j \left(-\frac{z^{-2} + vz^{-1} + u}{uz^{-2} + vz^{-1} + 1} \right)^j \\ &= \sum_{j=0}^{n} \left\{ (-1)^j \left(uz^{-2} + vz^{-1} + 1 \right)^{n-j} \left(z^{-2} + vz^{-1} + u \right)^j \right\} e_j. \end{aligned} \tag{3.29}$$

From (3.28) and (3.29), we have

$$(-1)^j \left(uz^{-2} + vz^{-1} + 1 \right)^{n-j} \left(z^{-2} + vz^{-1} + u \right)^j = \sum_{i=0}^{2n} t_{i,j}^{\mathrm{BP},n} z^{-i}. \tag{3.30}$$

Then, the following two propositions can be obtained.

Proposition 3.7 The elements $\left\{ t_{i,j}^{\mathrm{BP},n} \right\}$ of the matrix $\boldsymbol{T}_n^{\mathrm{BP}}$ satisfy the following properties.

(i) $$t_{0,j}^{\mathrm{BP},n} = (-u)^j, \quad j = 0, 1, \ldots, n, \tag{3.31}$$

(ii) $$t_{i,0}^{\mathrm{BP},n} = \sum_{k=0}^{i} \left\langle \begin{matrix} n \\ k \end{matrix} \right\rangle \left\langle \begin{matrix} k \\ i-k \end{matrix} \right\rangle u^{i-k} v^{2k-i}, \quad i = 0, 1, \ldots, 2n, \tag{3.32}$$

(iii) $$t_{i,j}^{\mathrm{BP},n} = (-1)^n \cdot t_{2n-i,n-j}^{\mathrm{BP},n}, \quad i = 0, 1, \ldots, 2n, \quad j = 0, 1, \ldots, n, \tag{3.33}$$

(iv) $$t_{n-i,j}^{\mathrm{BP},n} = (-u)^n \cdot t_{i,j}^{\mathrm{BP},n} \left(\frac{1}{u}, \frac{v}{u} \right), \quad i = 0, 1, \ldots, 2n, \quad j = 0, 1, \ldots, n, \tag{3.34}$$

(v) $$t_{i,j}^{\mathrm{BP},n} = -\left\{ t_{i-2,j-1}^{\mathrm{BP},n} + v \left(t_{i-1,j-1}^{\mathrm{BP},n} + t_{i-1,j}^{\mathrm{BP},n} \right) + u \left(t_{i,j-1}^{\mathrm{BP},n} + t_{i-2,j}^{\mathrm{BP},n} \right) \right\},$$
$$i = 0, 1, \ldots, 2n, \quad j = 0, 1, \ldots, n, \tag{3.35}$$

where $t_{i,j}^{\mathrm{BP},n} = 0, \quad i < 0$ or $j < 0$ or $i > 2n$ or $j > n$.

Proof The proofs of (i) and (ii) easily follow from (3.23) and (3.30), respectively.

Proof of (iii): Replacing j in (3.30) by $(n-j)$, we have

$$(-1)^{n-j}\left(uz^{-2}+vz^{-1}+1\right)^{j}\left(z^{-2}+vz^{-1}+u\right)^{n-j}=\sum_{i=0}^{2n}t_{i,n-j}^{\mathrm{BP},n}z^{-i}. \tag{3.36}$$

Next, replacing z^{-1} in (3.36) by z, and then multiplying $(-z)^{-n}$ on both sides, we have

$$(-1)^{j}\left(z^{-2}+vz^{-1}+u\right)^{j}\left(uz^{-2}+vz^{-1}+1\right)^{n-j}=\sum_{i=0}^{2n}\left\{(-1)^{n}\,t_{n-i,n-j}^{\mathrm{BP},n}\right\}z^{-i}. \tag{3.37}$$

The left side of (3.37) is equal to that of (3.30). Thus, (3.33) holds.

Proof of (iv): Replacing u and v in (3.30) by $1/u$ and $1/v$ respectively, and then multiplying $(-u)^{n}$ on both sides, we have

$$(-1)^{n-j}\left(z^{-2}+vz^{-1}+u\right)^{n-j}\left(uz^{-2}+vz^{-1}+1\right)^{j}=\sum_{i=0}^{2n}(-u)^{n}\cdot\left\{t_{i,j}^{\mathrm{BP},n}\left(\frac{1}{u},\frac{v}{u}\right)\right\}z^{-i}. \tag{3.38}$$

The left side of (3.38) is equal to that of (3.36).

Proof of (v): Replacing j in (3.30) by $(j-1)$, we have

$$(-1)^{j-1}\left(uz^{-2}+vz^{-1}+1\right)^{n-j+1}\left(z^{-2}+vz^{-1}+u\right)^{j-1}=\sum_{i=0}^{2n}t_{i,j-1}^{\mathrm{BP},n}z^{-i}. \tag{3.39}$$

From (3.30) and (3.39), we have

$$\left(uz^{-2}+vz^{-1}+1\right)\sum_{i=0}^{2n}t_{i,j}^{\mathrm{BP},n}z^{-i}=-\left(z^{-2}+vz^{-1}+u\right)\sum_{i=0}^{2n}t_{i,j-1}^{\mathrm{BP},n}z^{-i}. \tag{3.40}$$

Using (3.31)-(3.33), (3.40) can be rewritten as follows:

$$\sum_{i=1}^{n}\left\{t_{i,j}^{\mathrm{BP},n}+t_{i-2,j-1}^{\mathrm{BP},n}+v\left(t_{i-1,j-1}^{\mathrm{BP},n}+t_{i-1,j}^{\mathrm{BP},n}\right)+u\left(t_{i,j-1}^{\mathrm{BP},n}+t_{i-2,j}^{\mathrm{BP},n}\right)\right\}z^{-i}=0,$$

which implies (3.35). ■

Proposition 3.8 The relations between the elements $\left\{t_{i,j}^{\mathrm{BP},n}\right\}$ of $\boldsymbol{T}_{n}^{\mathrm{BP}}$ and $\left\{t_{i,j}^{\mathrm{BP},n-1}\right\}$ of $\boldsymbol{T}_{n-1}^{\mathrm{BP}}$ are as follows.

(i) $\quad t_{i,j}^{\mathrm{BP},n}=u\cdot t_{i-2,j}^{\mathrm{BP},n-1}+v\cdot t_{i-1,j}^{\mathrm{BP},n-1}+t_{i,j}^{\mathrm{BP},n-1}$,

$$i = 0, 1, \ldots, 2n, \quad j = 0, 1, \ldots, n, \tag{3.41}$$

(ii) $$t_{i,j}^{\mathrm{BP},n} = -\left(u \cdot t_{i,j}^{\mathrm{BP},n-1} + v \cdot t_{i-1,j}^{\mathrm{BP},n-1} + t_{i-2,j}^{\mathrm{BP},n-1}\right),$$

$$i = 0, 1, \ldots, 2n, \quad j = 0, 1, \ldots, n. \tag{3.42}$$

Proof Proof of (i): Replacing n in (3.30) by $(n-1)$, we have

$$(-1)^j \left(uz^{-2} + vz^{-1} + 1\right)^{n-1-j} \left(z^{-2} + vz^{-1} + u\right)^j = \sum_{i=0}^{2n-2} t_{i,j}^{\mathrm{BP},n-1} z^{-i}. \tag{3.43}$$

From (3.30) and (3.43), we have

$$\begin{aligned}
\sum_{i=0}^{2n} t_{i,j}^{\mathrm{BP},n} z^{-i} &= \left(uz^{-2} + vz^{-1} + 1\right) \sum_{i=0}^{2n-2} t_{i,j}^{\mathrm{BP},n-1} z^{-i} \\
&= u \sum_{i=0}^{2n-2} t_{i,j}^{\mathrm{BP},n-1} z^{-(i+2)} + v \sum_{i=0}^{2n-2} t_{i,j}^{\mathrm{BP},n-1} z^{-(i+1)} + \sum_{i=0}^{2n-2} t_{i,j}^{\mathrm{BP},n-1} z^{-i} \\
&= u \sum_{i=2}^{2n} t_{i-2,j}^{\mathrm{BP},n-1} z^{-i} + v \sum_{i=1}^{2n-1} t_{i-1,j}^{\mathrm{BP},n-1} z^{-i} + \sum_{i=0}^{2n-2} t_{i,j}^{\mathrm{BP},n-1} z^{-i} \\
&= \sum_{i=0}^{2n} \left(u \cdot t_{i-2,j}^{\mathrm{BP},n-1} + v \cdot t_{i-1,j}^{\mathrm{BP},n-1} + t_{i,j}^{\mathrm{BP},n-1}\right) z^{-i}.
\end{aligned}$$

Thus, (3.41) holds.

The proof of (ii) follows from (3.35) and (3.41). ■

3.3.1.4 Properties of the Matrices to Design Band-stop Filters

The transformation formula to design band-stop filters can be obtained from the formula to design band-pass filters by replacing $u, v, H_0(z)$ with $\tilde{u}, \tilde{v}, H_0(-z)$, respectively. From this, the following properties of the band-pass design matrix $\boldsymbol{T}_n^{\mathrm{BS}}$ are similarly derived.

Proposition 3.9 The elements $\left\{t_{i,j}^{\mathrm{BS},n}\right\}$ of the matrix $\boldsymbol{T}_n^{\mathrm{BS}}$ satisfy the following properties.

(i) $$t_{0,j}^{\mathrm{BS},n} = \tilde{u}^j, \quad j = 0, 1, \ldots, n,$$

(ii) $$t_{i,0}^{\mathrm{BS},n} = \sum_{k=0}^{i} \left\langle \begin{matrix} n \\ k \end{matrix} \right\rangle \left\langle \begin{matrix} k \\ i-k \end{matrix} \right\rangle \tilde{u}^{i-k} \tilde{v}^{2k-i}, \quad i = 0, 1, \ldots, 2n,$$

(iii) $$t_{i,j}^{\mathrm{BS},n} = t_{2n-i,n-j}^{\mathrm{BS},n}, \quad i = 0,1,\ldots,2n, \quad j = 0,1,\ldots,n.$$

(iv) $$t_{n-i,j}^{\mathrm{BS},n} = \tilde{u}^n \cdot t_{i,j}^{\mathrm{BS},n}\left(\frac{1}{\tilde{u}}, \frac{\tilde{v}}{\tilde{u}}\right), \quad i = 0,1,\ldots,2n, \quad j = 0,1,\ldots,n.$$

(v) $$t_{i,j}^{\mathrm{BS},n} = t_{i-2,j-1}^{\mathrm{BS},n} + \tilde{v}\left(t_{i-1,j-1}^{\mathrm{BS},n} - t_{i-1,j}^{\mathrm{BS},n}\right) + \tilde{u}\left(t_{i,j-1}^{\mathrm{BS},n} - t_{i-2,j}^{\mathrm{BS},n}\right),$$
$$i = 0,1,\ldots,2n, \quad j = 0,1,\ldots,n,$$

where

$$t_{i,j}^{\mathrm{BS},n} = 0, \quad i < 0 \text{ or } j < 0 \text{ or } i > 2n \text{ or } j > n.$$

■

Proposition 3.10 The relations between the elements $\left\{t_{i,j}^{\mathrm{BS},n}\right\}$ of $\boldsymbol{T}_n^{\mathrm{BS}}$ and $\left\{t_{i,j}^{\mathrm{BS},n-1}\right\}$ of $\boldsymbol{T}_{n-1}^{\mathrm{BS}}$ are as follows.

(i) $$t_{i,j}^{\mathrm{BS},n} = \tilde{u} \cdot t_{i-2,j}^{\mathrm{BS},n-1} + \tilde{v} \cdot t_{i-1,j}^{\mathrm{BS},n-1} + t_{i,j}^{\mathrm{BS},n-1},$$
$$i = 0,1,\ldots,2n, \quad j = 0,1,\ldots,n,$$

(ii) $$t_{i,j}^{\mathrm{BS},n} = -\left(\tilde{u} \cdot t_{i,j}^{\mathrm{BS},n-1} + \tilde{v} \cdot t_{i-1,j}^{\mathrm{BS},n-1} + t_{i-2,j}^{\mathrm{BS},n-1}\right),$$
$$i = 0,1,\ldots,2n, \quad j = 0,1,\ldots,n.$$

■

3.3.2 Fast Algorithms

In this subsection, some fast algorithms for computing the elements of the frequency transformation matrices are proposed.

3.3.2.1 Fast Design of Low-pass Filters

To compute all the elements of the matrix $\boldsymbol{T}_n^{\mathrm{LP}}$ used to design low-pass filters, the following two algorithms can be considered.

Algorithm 3.11 All the elements $\left\{t_{i,j}^{\mathrm{LP},n}\right\}$ of the matrix $\boldsymbol{T}_n^{\mathrm{LP}}$ are computed by the following five steps.

[a] The element $t_{0,0}^{\mathrm{LP},n}$ is one for any n.

[b] From (3.8), the following relation holds.

$$t_{0,j}^{\mathrm{LP},n} = (-\alpha) \cdot t_{0,j-1}^{\mathrm{LP},n}, \quad j = 1, 2, \ldots, n. \tag{3.44}$$

Based on the relation (3.44), the first-row elements $\left\{t_{0,j}^{\mathrm{LP},n}\right\}_{j=1}^{n}$ are determined.

[c] From (3.9), the following relation holds.

$$t_{i,0}^{\mathrm{LP},n} = t_{0,i}^{\mathrm{LP},n} \left\langle \begin{array}{c} n \\ i \end{array} \right\rangle, \quad i = 1, 2, \ldots, n. \tag{3.45}$$

Based on the relation (3.45), the first-column elements $\left\{t_{i,0}^{\mathrm{LP},n}\right\}_{i=1}^{\lfloor n/2 \rfloor}$ are determined.

[d] Based on (3.12), the elements $\left\{t_{i,j}^{\mathrm{LP},n}\right\}_{i=1}^{\lfloor n/2 \rfloor}$ are determined for $j = 1, 2, \ldots, n$.

[e] Based on (3.10), the elements $\left\{t_{i,j}^{\mathrm{LP},n}\right\}_{i=\lfloor n/2 \rfloor+1}^{n}$ are determined for $j = 0, 1, \ldots, n$. ■

Algorithm 3.12 The elements $\left\{t_{i,j}^{\mathrm{LP},n}\right\}$ of the matrix $\boldsymbol{T}_n^{\mathrm{LP}}$ can also be computed by the following six steps.

[a] Same as [a] of Algorithm 3.11.

[b] Same as [b] of Algorithm 3.11.

[c] Same as [c] of Algorithm 3.11.

[d] Based on (3.18), the elements $\left\{t_{i,j}^{\mathrm{LP},n}\right\}_{i=1}^{\lfloor n/2 \rfloor}$ are determined for $j = 1, 2, \ldots, n-1$. In this case, the elements of the $(n-1)$th-order matrix $\boldsymbol{T}_{n-1}^{\mathrm{LP}}$ are required. They are computed recursively.

[e] Based on (3.19), the $(n+1)$th-column elements $\left\{t_{i,n}^{\mathrm{LP},n}\right\}_{i=1}^{\lfloor n/2 \rfloor}$ are determined.

[f] Same as [e] of Algorithm 3.11. ■

In this paper, three algorithms are compared by the number of additions (including subtractions) and multiplications (including divisions), respectively. The three algorithms to be compared are as follows.

(a) Algorithm based on the definition (3.2).

(b) Algorithm 3.11.

(c) Algorithm 3.12.

Similar algorithms can be considered for the matrix $\boldsymbol{T}_n^{\mathrm{HP}}$ to design high-pass filters.

3.3.2.2 Fast Design of Band-pass Filters

To compute all the elements of the matrix $\boldsymbol{T}_n^{\mathrm{BP}}$ to design band-pass filters, the following two algorithms can also be considered.

Algorithm 3.13 All the elements $\left\{t_{i,j}^{\mathrm{BP},n}\right\}$ of the matrix $\boldsymbol{T}_n^{\mathrm{BP}}$ are computed by the following five steps.

[a] The first-row first-column element $t_{0,0}^{\mathrm{BP},n}$ is one for any n.

[b] From (3.31), the following relation holds.

$$t_{0,j}^{\mathrm{BP},n} = (-u) \cdot t_{0,j-1}^{\mathrm{BP},n}, \quad j = 1, 2, \ldots, n. \tag{3.46}$$

Based on the relation (3.46), the first-row elements $\left\{t_{0,j}^{\mathrm{BP},n}\right\}_{j=1}^{n}$ are determined.

[c] Based on (3.32), the first-column elements $\left\{t_{i,0}^{\mathrm{BP},n}\right\}_{i=1}^{n}$ are determined.

[d] Based on (3.35), the elements $\left\{t_{i,j}^{\mathrm{BP},n}\right\}_{i=1}^{n}$ are determined for $j = 1, 2, \ldots, n$.

[e] Based on (3.33), the elements $\left\{t_{i,j}^{\mathrm{BP},n}\right\}_{i=n+1}^{2n}$ are determined for $j = 0, 1, \ldots, n$. ■

Algorithm 3.14 The elements $\left\{t_{i,j}^{\mathrm{BP},n}\right\}$ of the matrix $\boldsymbol{T}_n^{\mathrm{BP}}$ can also be computed by the following six steps.

[a] Same as [a] of Algorithm 3.13.

[b] Same as [b] of Algorithm 3.13.

[c] Same as [c] of Algorithm 3.13.

[d] Based on (3.41), the elements $\left\{t_{i,j}^{\mathrm{BP},n}\right\}_{i=1}^{n}$ are determined for $j = 1, 2, \ldots, n-1$. In this case, the elements of the $(n-1)$th-order matrix $\boldsymbol{T}_{n-1}^{\mathrm{BP}}$ must be computed. They are computed recursively.

[e] Based on (3.42), the $(n+1)$th-column elements $\left\{t_{i,n}^{\mathrm{BP},n}\right\}_{i=1}^{n}$ are determined.

[f] Same as [e] of Algorithm 3.13. ■

Just as in Subsection 3.1, three algorithms are compared by the number of computations of addition (including subtraction) and multiplication (including division), respectively. The three algorithms to be compared are as follows.

(a) Algorithm based on the definition (3.23).

(b) Algorithm 3.13.

(c) Algorithm 3.14.

Similar algorithms can be considered for the matrix $\boldsymbol{T}_n^{\mathrm{HP}}$ to design band-stop filters.

3.3.3 Evaluation

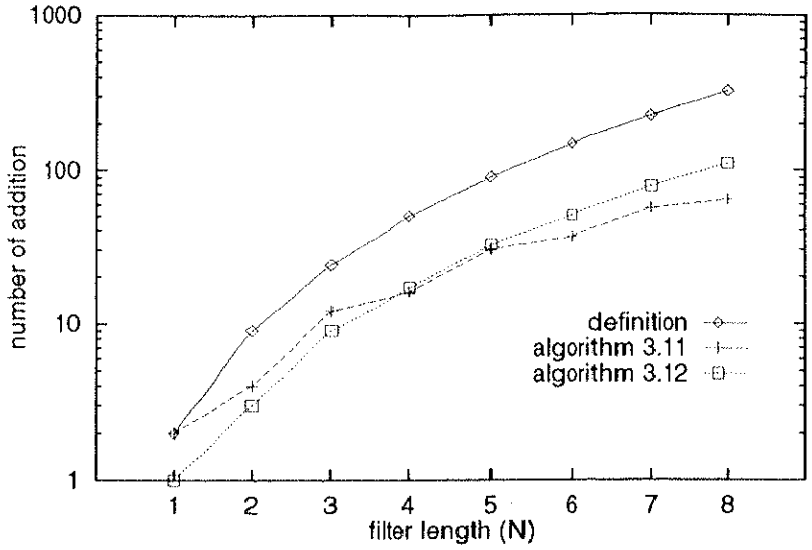
The algorithms are compared with those based on the definitions by the number of additions and multiplications. Based on the comparison, the best algorithms will be chosen.

Figure 3.6 respectively show the number of additions and multiplications needed to compute all the elements of the matrix $\boldsymbol{T}_n^{\mathrm{LP}}$ for the above three algorithms. Comparing the algorithm (a) with the algorithms (b) and (c), the latter two can compute the elements of the matrix $\boldsymbol{T}_n^{\mathrm{LP}}$ with much fewer operation than (a). Comparing (b) with (c), (b) seems to work as same as (c) in the case $n \leq 3$, and better than (c) in the case $n \geq 4$. Therefore, the algorithm (b) can be considered the best algorithm needed to compute the elements of the matrix $\boldsymbol{T}_n^{\mathrm{LP}}$. Though a computer program is shown in [21] to compute the coefficients of the transfer function derived by the frequency transformation, the program is just based on the definition of the transformation formulae. It can be concluded that the number of operations of the program is more than that of the algorithm (b).

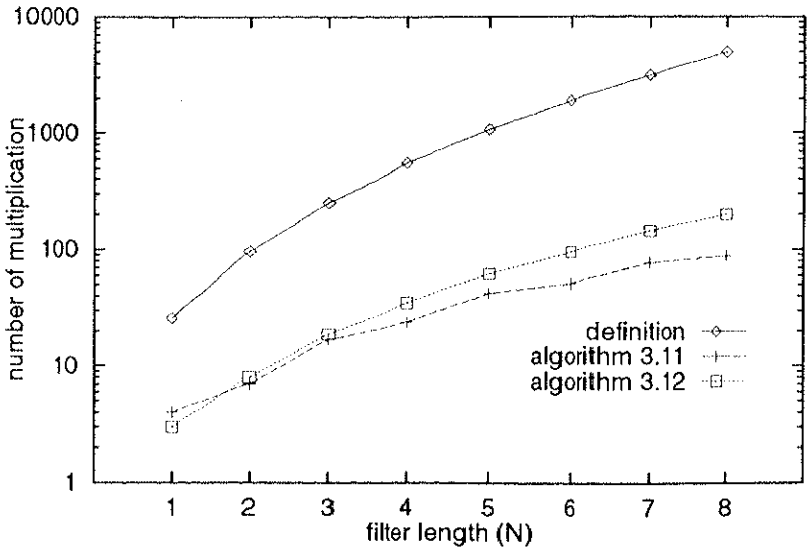
Figure 3.7 shows the number of additions and multiplications to compute all the elements of the matrix $\boldsymbol{T}_n^{\mathrm{BP}}$ for the above three algorithms. Comparing the algorithm (a) with the algorithms (b) and (c), the latter two can compute the elements of the matrix $\boldsymbol{T}_n^{\mathrm{BP}}$ with much fewer operation than (a). Comparing (b) with (c), (b) seems to work better than the algorithm 3.14 for any n. Therefore, the algorithm (b) can be considered the best algorithm to compute the elements of the matrix $\boldsymbol{T}_n^{\mathrm{BP}}$ in this case as well.

3.4 Summary

In this chapter, frequency transformation matrices were first proposed for analog filters and IIR digital filters. The proposed matrices could replace hand computation in those



(a) number of addition

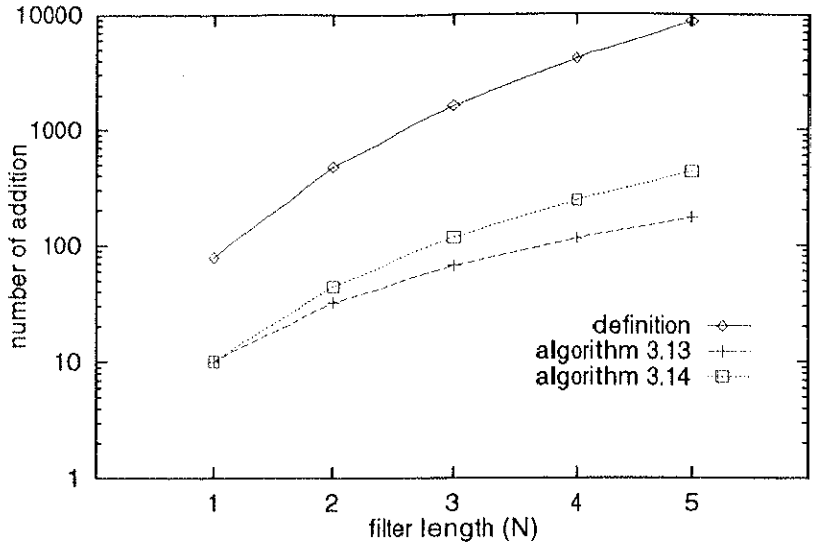


(b) number of multiplication

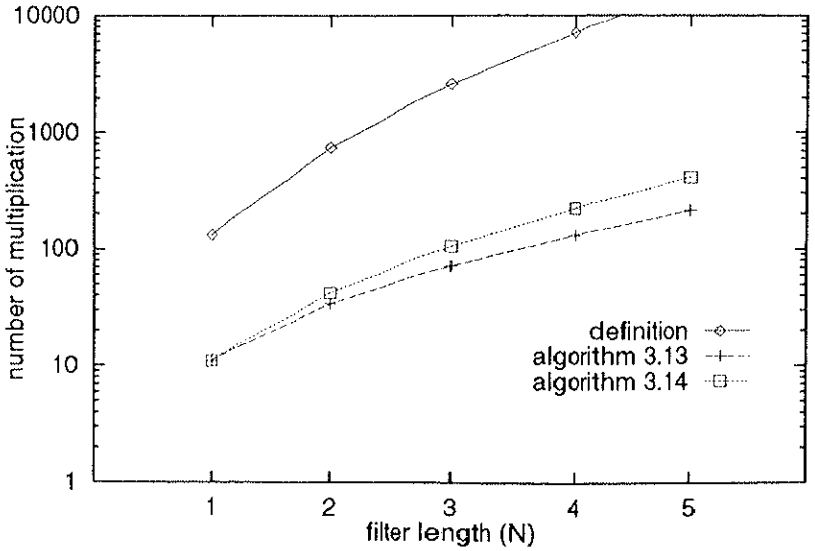
Figure 3.6: Comparison of the number of operations needed to compute all the elements of the matrix $\boldsymbol{T}_n^{\mathrm{LP}}$ (linear-to-log scale).

transformations by automatic procedures. Furthermore, an automatic procedure was established to design various types of IIR digital filters directly from an analog low-pass filter.

This chapter also developed properties of the frequency transformation matrices for



(a) number of addition



(b) number of multiplication

Figure 3.7: Comparison of the number of operations needed to compute all the elements of the matrix $\boldsymbol{T}_n^{\mathrm{BP}}$ (linear-to-log scale).

IIR digital filters. Fast algorithms to compute the elements of these matrices were also developed. It was shown that the proposed fast algorithms required quite fewer operations than the algorithms based on the definitions of matrices.

It would be helpful for the design if we know the required precision to the original

low-pass filter from the desired precision of the derived filters. Generally we prepare the original filter with enough accurate precision to meet any desired precision of the derived filters, although it is usually wasteful. Properties of the transfer matrices may enable to know the required precision to the original low-pass filter in advance of the design. the properties would be studied further.

Chapter 4

Accurate Estimation of Minimum Filter Length for Optimum FIR Digital Filters

4.1 Introduction

Shorter filters (filters with shorter filter length) have advantages over longer filters in that they have fewer circuit elements in a hardware implementation or less computational cost in a software implementation. Therefore, any filter design problem can be considered to be an optimization problem to find a filter satisfying the given specifications with a minimum filter length.

Especially for designing optimum (minimax criterion based) linear-phase FIR digital filters, an iterative optimizing algorithm using Remez exchange method [34] has been established for FIRs with odd filter length [28], and also for those with even filter length [31]. This algorithm is the most widely used for the design of linear-phase FIRs because of its flexible and efficient performance. However the algorithm requires the filter length of the designed filter to be known in advance, and optimizes the amplitude characteristics in the minimax sense for a specified filter length.

Suppose the case to design an FIR digital filter of low-pass type. Specifications of a

target filter are generally given by four parameters: passband edge frequency f_p, stopband edge frequency f_s ($f_p < f_s$), passband ripple δ_p and stopband ripple δ_s (usually $\delta_p \geq \delta_s$). In many practical cases, the above four parameters of a target filter is first specified, and then many filters are designed so as to see how long the minimum filter length N must be. It is hard to know the exact value of the minimum filter length N which satisfies given specifications.

To conjecture an appropriate filter length from given specifications in advance, two estimation formulae have been proposed by Herrmann *et al.* [24, 29] and by Kaiser [27] for the design of optimum FIR low-pass filters. However these formulae cannot achieve enough accuracy because of lack of some considerations. Since it takes much time and troubles to establish more accurate formula, the conventional formulae are still used in practice for the estimation. Such situation must be improved.

In this research, we aim to more accurately estimate the minimum filter length of optimum FIR low-pass filters. The proposed estimation formula, described in Section 4.3, is mainly based on the observations from our experimental results. The formula is first established for FIR low-pass filters with the identical pass-band and stop-band ripples in Subsection 4.3.1. Then it is expanded for every FIR low-pass filter in 4.3.2, and finally developed for high-pass, band-pass and band-stop filters in 4.3.3.

The accuracy of the proposed formula is evaluated by some design examples and quantitative distances in comparison with those of the conventional formulae in Section 4.4.

The results of this chapter are summarized in Section 4.5.

4.2 Conventional Formulae and Their Problems

4.2.1 Conventional Formulae

In this Subsection, $\langle a \rangle$ and $\lceil a \rceil$ denote the nearest odd integer from a and the minimum odd integer not less than a, respectively.

In Refs. [24, 29], Herrmann *et al.* proposed the following estimation formula:

$$\hat{N}_1(\Delta F, \delta_p, \delta_s) = \left\langle \frac{D_\infty(\delta_p, \delta_s)}{\Delta F} - f(\delta_p, \delta_s) \cdot \Delta F + 1 \right\rangle, \tag{4.1}$$

where

$$\begin{aligned} D_\infty(\delta_p, \delta_s) &= \left\{ a_1 (\log_{10} \delta_p)^2 + a_2 \log_{10} \delta_p + a_3 \right\} \log_{10} \delta_s \\ &\quad + \left\{ a_4 (\log_{10} \delta_p)^2 + a_5 \log_{10} \delta_p + a_6 \right\}, \\ f(\delta_p, \delta_s) &= b_1 + b_2 (\log_{10} \delta_p - \log_{10} \delta_s), \\ a_1 &= 5.309 \times 10^{-3}, \quad a_2 = 7.114 \times 10^{-2}, \quad a_3 = -4.761 \times 10^{-1}, \\ a_4 &= -2.66 \times 10^{-3}, \quad a_5 = -5.941 \times 10^{-1}, \quad a_6 = -4.278 \times 10^{-1}, \\ b_1 &= 11.01217, \quad b_2 = 0.51244. \end{aligned}$$

In (4.1), ΔF denotes the transition width $(f_s - f_p)$. In addition, Kaiser [27] also proposed the following formula independently:

$$\hat{N}_2(\Delta F, \delta_p, \delta_s) = \left\lceil \frac{-20 \log_{10} \sqrt{\delta_p \delta_s} - 13}{14.6 \Delta F} + 1 \right\rceil. \tag{4.2}$$

In the case $\delta_p = \delta_s$ $(= \delta)$, (4.1) and (4.2) are rearranged as functions of two parameters ΔF and δ:

$$\hat{N}_1(\Delta F, \delta) = \left\langle \frac{D_\infty(\delta, \delta)}{\Delta F} - b_1 \cdot \Delta F + 1 \right\rangle, \tag{4.3}$$

$$\hat{N}_2(\Delta F, \delta) = \left\lceil \frac{-20 \log_{10} \delta - 13}{14.6 \Delta F} + 1 \right\rceil. \tag{4.4}$$

4.2.2 Problems

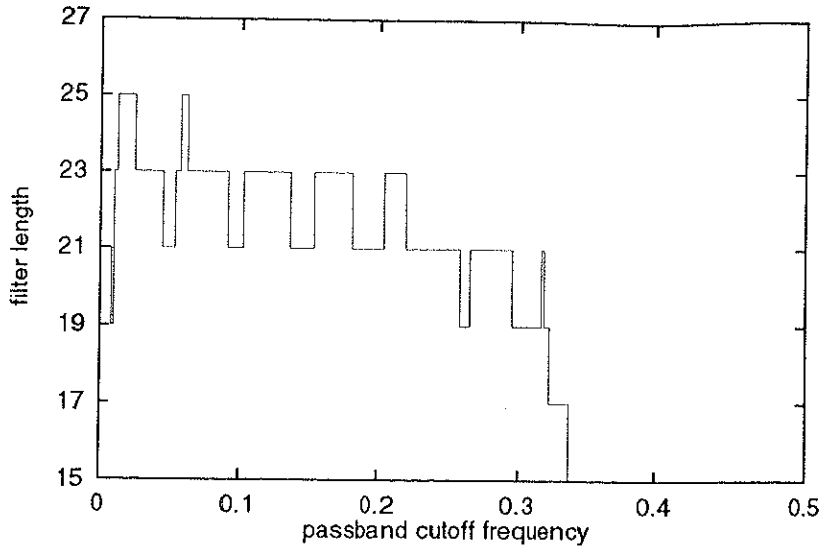
4.2.2.1 Historical Problems

As the estimated value $\hat{N}_1$ in (4.1) is rounded to the nearest odd integer, it is very particular to the FIRs of odd filter length, and does not consider those of even filter length. Similarly, $\hat{N}_2$ in (4.2) is rounded up to the nearest odd integer. This means that the estimation formula (4.2) also does not include the FIRs of even filter length. Figure 4.1 shows the behavior of the minimum odd filter length N_{odd} as a function of f_p, which is obtained by actually designing the example filters used in Fig. 14 and 15 of [24]. Figure 4.1(a) shows the behavior for the case $\delta_p = 0.01$, $\delta_s = 0.0001$ and $\Delta F = 0.158$, and Fig. 4.1(b) for the case $\delta_p = 0.01$, $\delta_s = 0.0001$ and $\Delta F = 0.032$. The estimation formulae (4.1) and (4.2) were made based on the data of Fig. 4.1 in 1973 and 1974, respectively. However, a design algorithm for FIRs of even filter length [31] was proposed in 1973. Figure 4.2 shows the behavior of the minimum filter length N as a function of f_p for the same specifications of Fig. 4.1, but N can be even by the algorithm [31]. From Figs. 4.1 and 4.2, we can see that

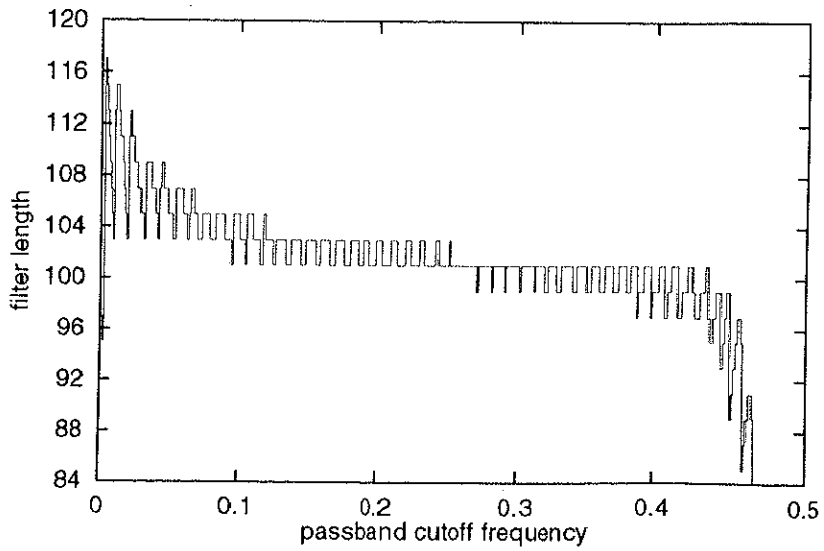
$$N \leq N_{\text{odd}}, \quad \forall f_p \in [0, 0.5 - \Delta F].$$

Since the design algorithm has already been established for FIRs of both odd and even filter length, the estimation formulae for filter length should be newly considered based on the data of Fig. 4.2 to make the algorithm more useful.

Furthermore, when the formulae (4.1) and (4.2) were established, FIRs of longer filter length (approximately more than 150) were not feasible, hence the formulae must have been formulated only for the FIRs of short filter length. Actually, as shown later, the estimation accuracies of the formulae (4.1) and (4.2) become worse as the filter length



(a) in the case $\delta_p = 0.01$, $\delta_s = 0.0001$ and $\Delta F = 0.158$

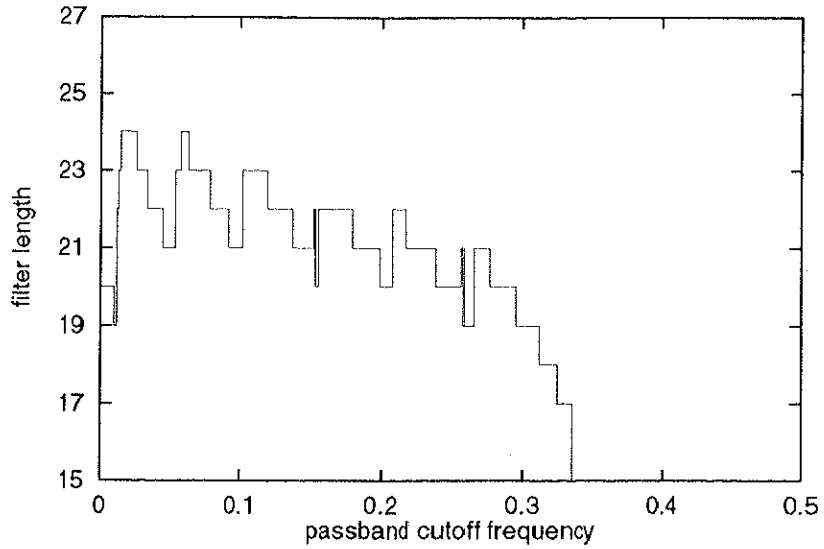


(b) in the case $\delta_p = 0.01$, $\delta_s = 0.0001$ and $\Delta F = 0.032$

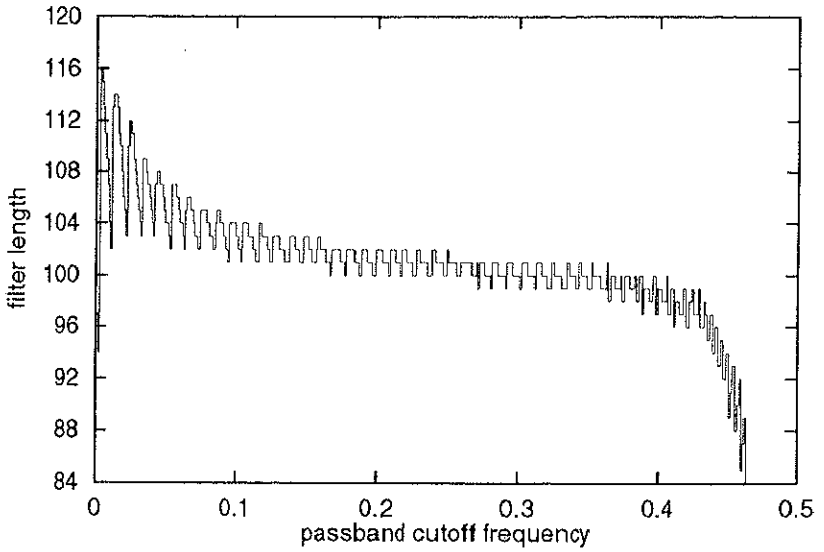
Figure 4.1: Behavior of the minimum odd filter length N_{odd} as a function of f_p.

becomes long. Now that FIRs of longer filter length can be designed, the accuracy for longer filter length should be improved.

From the above discussions, we can summarize the historical problems of the conventional formulations (4.1) and (4.2) as follows.



(a) in the case $\delta_p = 0.01$, $\delta_s = 0.0001$ and $\Delta F = 0.158$



(b) in the case $\delta_p = 0.01$, $\delta_s = 0.0001$ and $\Delta F = 0.032$

Figure 4.2: Behavior of the minimum (integer) filter length N as a function of f_p.

Problem 4.1 The formulae (4.1) and (4.2) are made based on the data of FIRs of odd filter length only, and do not mention those of even filter length.

Problem 4.2 The formulae (4.1) and (4.2) do not correspond to the FIRs of long filter length.

4.2.2.2 Problems on Formulation

Since specifications of a filter are given by four parameters: f_p, f_s, δ_p and δ_s, the minimum filter length N can be a function of those four variables. In the case using ΔF $(= f_s - f_p)$ instead of f_s, N can be rewritten as a function of f_p, ΔF, δ_p and δ_s. However in (4.1) and (4.2), the estimated minimum filter length $\hat{N}_1$ and $\hat{N}_2$ are written as functions of three-variables: ΔF, δ_p and δ_s. This means that they are constant irrespective of f_p.

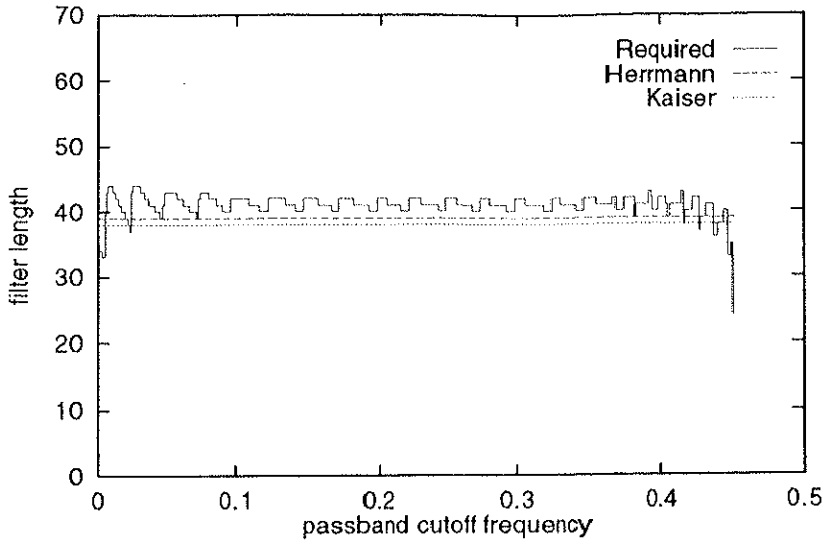
If N is really independent of f_p, the graph of N versus f_p must be drawn as a horizontal straight line. However, as shown in Figs. 4.1 and 4.2, N becomes small as f_p increases. This result leads to the fact that the minimum filter length N depends on all of the four variables including f_p.

Hereafter, $[a]$ and $\lceil a \rceil$ newly denote the nearest integer from a and the minimum integer not less than a respectively, to deal with all the integers.

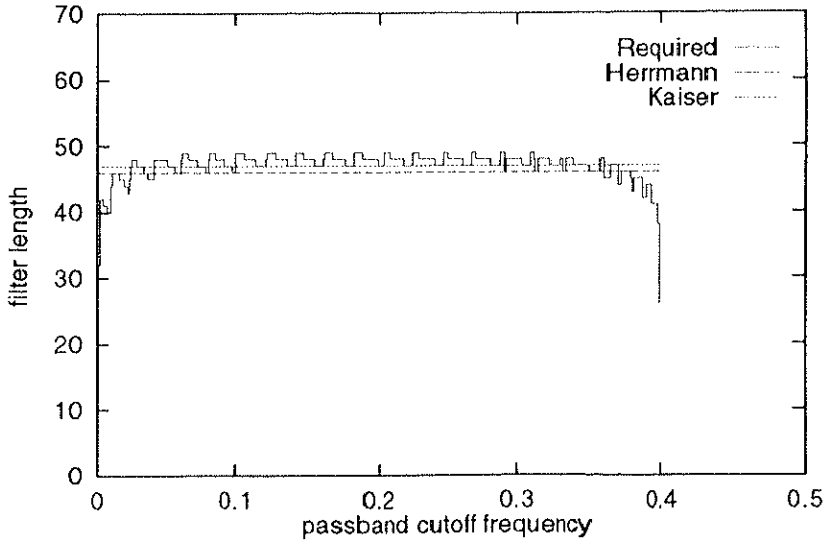
The case $\delta_p = \delta_s$ $(= \delta)$ is studied first. Figure 4.3 shows the behavior of the minimum filter length N as a function of f_p. Figure 4.3(a) shows the behavior for the case $\delta = 0.001$ and $\Delta F = 0.05$, and Fig.4.3(b) for the case $\delta = 0.0001$ and $\Delta F = 0.1$, where the solid, broken and dotted lines denote the required minimum filter length N by trial and error experiments, the estimated filter length $\hat{N}_1$ by (4.3) and the estimated filter length $\hat{N}_2$ by (4.4), respectively.

From Fig.4.3, we can see that the conventional formulae (4.3) and (4.4) in the case $\delta_p = \delta_s$ have the following problems.

Problem 4.3 For any ΔF and δ, the minimum filter length N becomes shorter as f_p gets close to 0 or $0.5 - \Delta F$.



(a) In the case $\delta_p = 0.1$, $\delta_s = 0.001$ and $\Delta F = 0.03$



(b) In the case $\delta_p = 0.01$, $\delta_s = 0.0001$ and $\Delta F = 0.05$

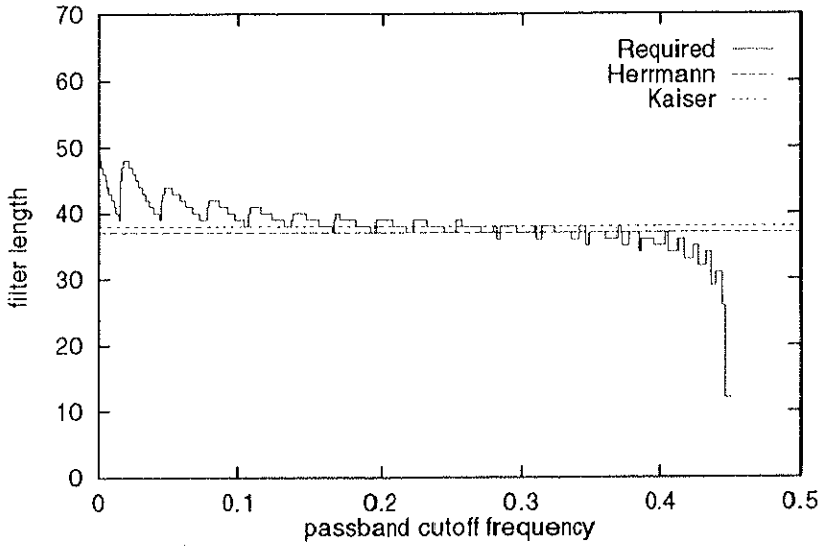
Figure 4.3: Behavior of the minimum filter length N and the estimated filter lengths $\hat{N}_1$ and $\hat{N}_2$ as a function of f_p.

Problem 4.4 Let f_c denote the center frequency in the interval $[0, 0.5 - \Delta F]$, *i.e.*, $f_c = (0.5 - \Delta F)/2$. For any ΔF and δ, if f_p is placed around f_c, the formulae (4.3) and (4.4) tend to give the shorter filter length than actually required.

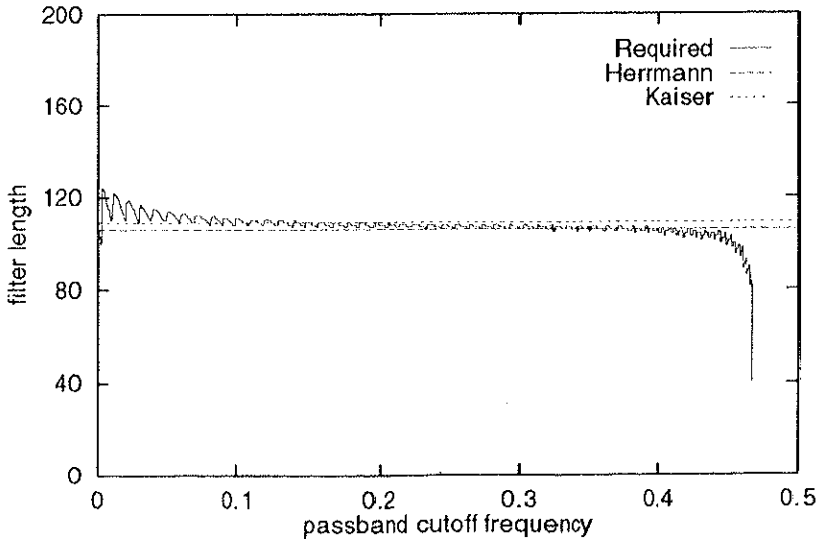
The above problems of the formulae (4.1) and (4.2) are due to the lack of consideration

of some significant theoretical properties, which are mentioned later.

Next the case $\delta_p \neq \delta_s$ is studied. Figure 4.4 shows the behavior of the minimum filter length N as a function of f_p. From Fig.4.4, we can see that the conventional formulae (4.1) and (4.2) have the following problem.



(a) In the case $\delta = 0.01$ and $\Delta F = 0.05$



(b) In the case $\delta = 0.0001$, $\Delta F = 0.1$

Figure 4.4: Behavior of the minimum filter length N and the estimated filter lengths $\hat{N}_1$ and $\hat{N}_2$ as a function of f_p.

Problem 4.5 For any ΔF and any $\delta_p > \delta_s$, the minimum filter length N becomes longer as f_p approaches to 0, and becomes shorter as f_p approaches $0.5 - \Delta F$, but the formulae (4.1) and (4.2) are constant irrespective of f_p.

Since specifications of low-pass filters are given by four-parameters, estimation formulae must be written as four-variable functions. However both of the formulae (4.1) and (4.2) are three-variable functions of $\Delta F, \delta_p$, and δ_s.

4.3 Proposed Formula

4.3.1 For Low-pass filters with Identical Pass-band and Stop-band Ripples

An accurate estimation formula $\hat{N}_3(f_p, \Delta F, \delta)$ is proposed in this section. First the relations between filter parameters are studied based on experimental results and theoretical considerations. Then the formula $\hat{N}_3$ is formulated.

4.3.1.1 Relations between Filter Parameters

In the case $\delta_p = \delta_s = \delta$, the required minimum filter length N_3 is determined by three parameters: f_p, f_s and δ. Hence, we express N by

$$N_3 = F_1(f_p, f_s, \delta).$$

Since f_s can be represented by f_p and ΔF, N can also be expressed by

$$N_3 = F_2(f_p, \Delta F, \delta) = F_1(f_p, f_p + \Delta F, \delta). \tag{4.5}$$

(*a*) *Relation between N_c and δ*

Let N_c denote the minimum filter length N for fixed $f_p = f_c$. First, the relation between N_c and δ is studied for various ΔF.

Figure 4.5 shows a behavior of the minimum filter length N_c as a function of ripple δ for some ΔF. Here, the following proposition 4.6 theoretically holds.

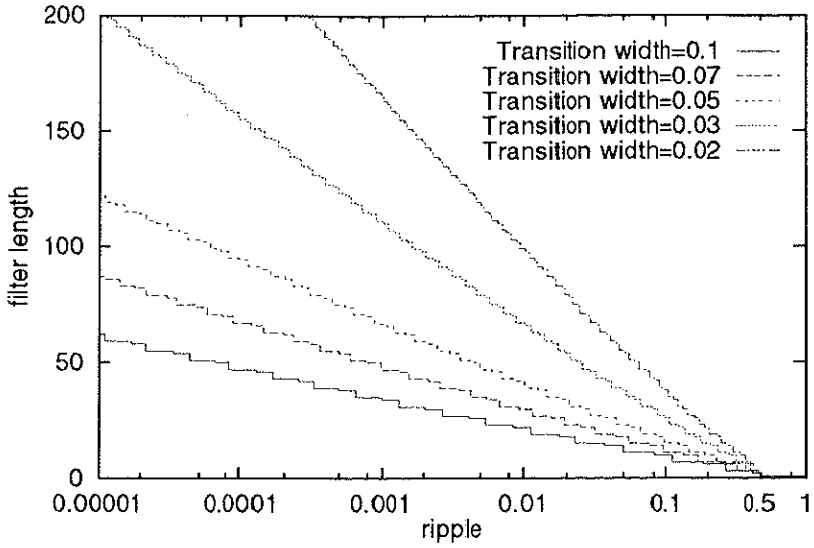


Figure 4.5: Behavior of the minimum filter length N_c as a function of δ (log-to-linear scale).

Proposition 4.6 For any ΔF, the minimum filter length N_c is one if $\delta = 0.5$.

Proof Consider a filter $H(z) = 0.5$ of the minimum filter length one. For any ΔF, the amplitude of the ripples caused by the filter in both the passband and the stopband is 0.5. This means that $H(z)$ is a filter satisfying $\delta = 0.5$. Therefore, the minimum filter length is one for any ΔF when $\delta = 0.5$. ■

The fact of Proposition 4.6 is also observed experimentally in Fig.4.5. The conventional formulae (4.3) and (4.4) lacked consideration to this fact.

From Fig. 4.5, we also have the following observation.

Observation 4.7 For any ΔF, $(N_c - 1)$ is almost a linear function of $\log_{10}\delta$.

This observation seems to be noticed in Kaiser's formula (4.4), but it is missed in Herrmann's formula (4.3).

(*b*) *Relation between* N_c *and* ΔF

Next, the relation between N_c and ΔF is studied for various δ.

Figure 4.6 illustrates the behavior of $(N_c - 1)$ as a function of ΔF for some values of δ. From Fig. 4.6, the following observation seems to hold.

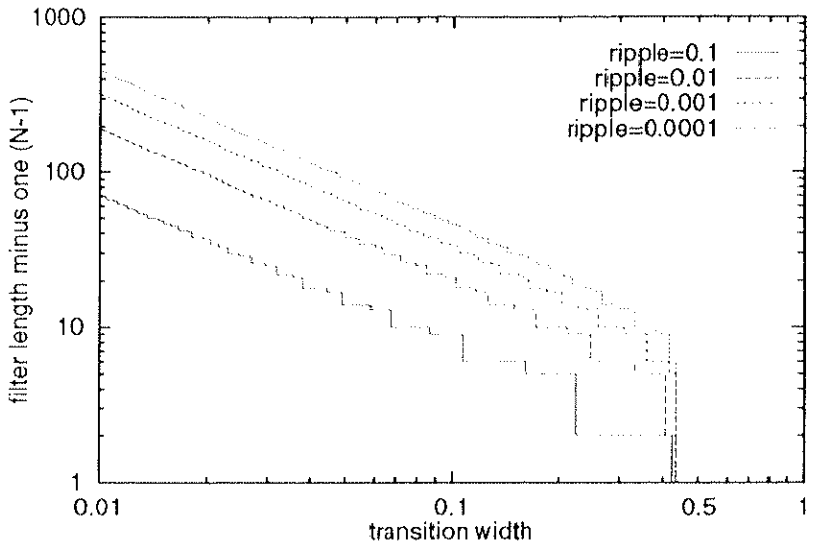


Figure 4.6: Behavior of the minimum filter length N_c as a function of ΔF (log-to-log scale).

Observation 4.8 For any δ, it seems that $\log_{10}(N_c - 1)$ is a linear function of $\log_{10} \Delta F$ with the gradient -1 when $N_c \geq 10$.

This is equivalent to the following observation:

Observation 4.9 For any δ, $(N_c - 1)$ is nearly in inverse proportion to ΔF when $N_c \geq 10$.

These are also mentioned in the Kaiser's formula (4.4), but the consideration to this observation in the Herrmann's formula (4.3) is not in enough detail.

(c) *Relation between N and f_p*

Next, the relation between N and f_p are studied for some values of ΔF and δ. As seen later in Fig. 4.3, the following seems to hold.

Observation 4.10 For any ΔF and δ, the graph of N versus f_p resembles the shape of an arch. In other words, the minimum filter length N is shorter than N_c when $f_p \simeq 0$ or $f_p \simeq 0.5 - \Delta F$.

4.3.1.2 Formulation

(a) *Estimation of N_c for $f_p = f_c$*

First, the estimated value $\hat{N}_c$ of the minimum filter length N_c for $f_p = f_c$ is formulated.

From Proposition 4.6 and Observation 4.7, $\hat{N}_c$ can be written as a function of δ as follows:

$$\begin{aligned} \hat{N}_c(\delta) &= \left\lceil a\left\{-\log_{10}\delta + \log_{10} 0.5\right\}^b + 1 \right\rceil \\ &= \left\lceil a\left\{-\log_{10}(2\delta)\right\}^b + 1 \right\rceil, \quad b \simeq 1. \end{aligned} \tag{4.6}$$

From Observations 4.8 and 4.9, $\hat{N}_c$ should also satisfy, as a function of ΔF, that

$$\log_{10}\left(\hat{N}_c(\Delta F) - 1\right) = c\left\{\log_{10}(\Delta F)\right\} + d, \quad c \simeq -1,$$

namely,

$$\hat{N}_c(\Delta F) = \left\lceil 10^d \cdot (\Delta F)^c + 1 \right\rceil. \tag{4.7}$$

Equations (4.6) and (4.7) lead to one possible expression of $\hat{N}_c$:

$$\hat{N}_c(\Delta F, \delta) = \left\lceil p\left\{-\log_{10}(2\delta)\right\}^q (\Delta F)^r + 1 \right\rceil. \tag{4.8}$$

Here, the parameters p, q and r in (4.8) are determined based on the following LMS approximation criterion for thousands of combinations of $\{\Delta F, \delta, N_c\}$:

$$\sum \left| \hat{N}_c - N_c \right|^2 \rightarrow \min .$$

Consequently, the estimation formula $\hat{N}_c$ is obtained as

$$\hat{N}_c(\Delta F, \delta) = \left\lceil \frac{1.101 \left\{ -\log_{10}(2\delta) \right\}^{1.1}}{\Delta F} + 1 \right\rceil . \tag{4.9}$$

(b) The proposed estimation formula $\hat{N}_3$ for $\delta_p = \delta_s = \delta$

Finally, the approximation function for the arch-like curves is found so as to determine a new estimation formula $\hat{N}_3$.

From the experiments of fitting various elementary functions to the arch-like curves, we found that the following modified arctangent function (4.10) is most suitable for approximating them.

$$g(f_p, \Delta F, \delta) := \frac{2}{\pi} \arctan \left\{ v(\Delta F, \delta) \cdot \left(\frac{1}{f_p} - \frac{1}{(0.5 - \Delta F)} \right) \right\}, \tag{4.10}$$

$$v(\Delta F, \delta) := 2.325 \cdot (-\log_{10} \delta)^{-0.445} \cdot (\Delta F)^{-1.39}. \tag{4.11}$$

The coefficients in (4.11) are also determined based on LMS approximation criterion. In this case, several hundred thousands of combinations of $\{f_p, \Delta F, \delta, N\}$ were used to determine the coefficients. This approximation problem is formulated as

$$\sum \left| \hat{N}_3 - N \right|^2 \rightarrow \min .$$

Finally we obtain the following estimation formula $\hat{N}_3$:

$$\hat{N}_3(f_p, \Delta F, \delta) := \left\lceil \hat{N}_c(\Delta F, \delta) \cdot \frac{g(f_p, \Delta F, \delta) + g(0.5 - \Delta F - f_p, \Delta F, \delta) + 1}{3} \right\rceil . \tag{4.12}$$

The formula (4.12) is a function of three-variables: f_p, ΔF and δ.

4.3.2 For any Low-pass Filter

Using (4.12), we expand a new estimation formula $\hat{N}_4$ which corresponds to the case $\delta_p \neq \delta_s$.

4.3.2.1 Relations between Filter Parameters

In the case $\delta_p \neq \delta_s$, the minimum filter length N_4 is determined by four parameters: f_p, f_s, δ_p and δ_s. Hence, we express N_4 by

$$N_4 = G_1(f_p, f_s, \delta_p, \delta_s).$$

Since f_s can be represented by f_p and ΔF, N can also be expressed by

$$N_4 = G_2(f_p, \Delta F, \delta_p, \delta_s) = G_1(f_p, f_p + \Delta F, \delta_p, \delta_s). \tag{4.13}$$

We assume that the minimum filter length N_4 for $\delta_p \neq \delta_s$ is represented in a form of an addition of the formula N_3 for $\delta_p = \delta_s$ in (4.5) and the distance DN, *i.e.*,

$$N_4(f_p, \Delta F, \delta_p, \delta_s) = N_3(f_p, \Delta F, \delta_p) + DN(f_p, \Delta F, \delta_p, \delta_s), \tag{4.14}$$

First we aim to formulate the approximation $D\hat{N}$ for the distance DN.

4.3.2.2 Formulation of $D\hat{N}$

We do not optimize the approximate distance $D\hat{N}$ for DN, but optimize the whole filter length $\hat{N}_4 = \hat{N}_3 + D\hat{N}$ for the required filter length N_4. This approximation problem can be formulated as

$$\sum \left| \hat{N}_4 - N \right|^2 \rightarrow \min . \tag{4.15}$$

Furthermore, the approximation $D\hat{N}$ must satisfy

$$D\hat{N} = 0, \quad \delta_p = \delta_s.$$

so as to be $N_4 = N_3$.

Now we study the behavior of the distance DN. For example, Figure 4.7 shows the behavior of the required filter length N_4 for the case $\Delta F = 0.05, \delta_p = 0.01, \delta_s = 0.00001$, the required filter length N_3 for the case $\Delta F = 0.05, \delta_p = \delta_s = 0.01$, and their distance DN as a function of f_p. From Fig. 4.7, we have the following observation.

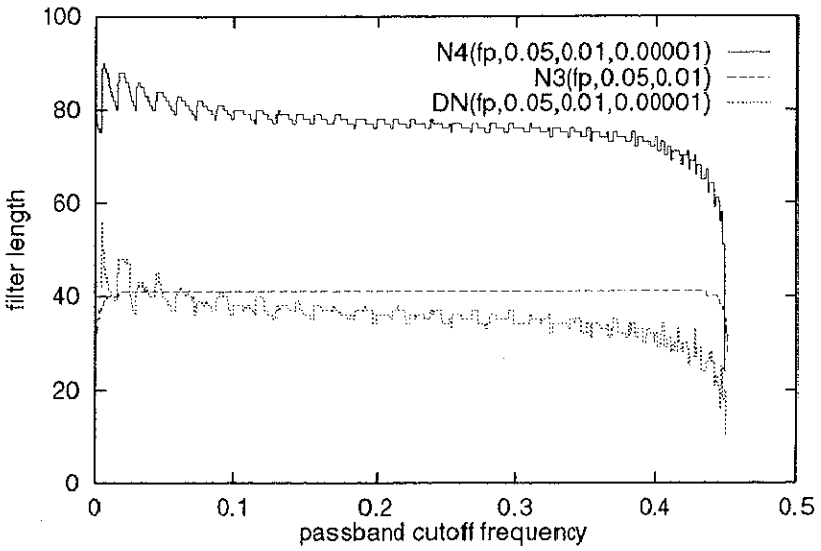


Figure 4.7: Behavior of the minimum filter length N_4 for the case $\Delta F = 0.05$, $\delta_p = 0.01$, $\delta_s = 0.00001$, the minimum filter length N_3 for the case $\Delta F = 0.05$, $\delta_p = \delta_s = 0.01$, and their distance DN as a function of f_p .

Observation 4.11 The distance DN gets smaller as f_p become larger. Especially when f_p get close to $0.5 - \Delta F$, the distance DN decreases rapidly. This behavior of DN resembles the behavior of N_3 if f_p is larger than f_c.

It would be mentioned here that the similar behavior were observed for the other specifications. This observation suggests that the modified arctangent function like g of (4.10) may be suited for approximating DN. From the experiments of fitting various elementary functions to the curves of DN, we found that the following modified arctangent function

h is most suitable. These approximate functions can be written as

$$D\hat{N}(f_p, \Delta F, \delta_p, \delta_s) := \left\lceil \hat{N}_m(\Delta F, \delta_p, \delta_s) \cdot \left\{ h(f_p, \Delta F, 1.1) - \frac{h(0.5 - \Delta F - f_p, \Delta F, 0.29) - 1}{2} \right\} \right\rceil, \tag{4.16}$$

$$h(f_p, \Delta F, c) := \frac{2}{\pi} \cdot \arctan\left\{ \frac{c}{\Delta F} \cdot \left(\frac{1}{f_p} - \frac{1}{0.5 - \Delta F} \right) \right\},$$

4.3.2.3 Formulation of N_m

The function $\hat{N}_m(\Delta F, \delta_p, \delta_s)$ in (4.16) is an approximation for N_m which is nearly the same as $DN(f_c, \Delta F, \delta_p, \delta_s)$. This approximation $\hat{N}_m$ works like $\hat{N}_c$ for the formula $\hat{N}_3$ in (4.12). Now the behavior of N_m is studied, and its approximation $\hat{N}_m$ is formulated.

(a) Relation between N_m and ΔF

First, the relation between N_m and ΔF is studied for various δ_p and δ_s.

Figure 4.8 illustrates the behavior of N_m as a function of ΔF for some values of δ_p and δ_s. From Fig. 4.8, the following observation seems to hold.

Observation 4.12 For any δ_p and δ_s, it seems that $\log_{10} N_m$ is a linear function of $\log_{10} \Delta F$ with the gradient -1.

This is equivalent to the following observation:

Observation 4.13 For any δ_p and δ_s, N_m is nearly in inverse proportion to ΔF.

(b) Relation between N_m and $\log(\delta_p/\delta_s)$

Next, the relation between N_m and a function $\log(\delta_p/\delta_s)$ is studied for various ΔF, δ_p and δ_s.

Figure 4.8 illustrates the behavior of N_m as a function of ΔF for some values of δ_p and ΔF. From Fig. 4.9, we have the following observation.

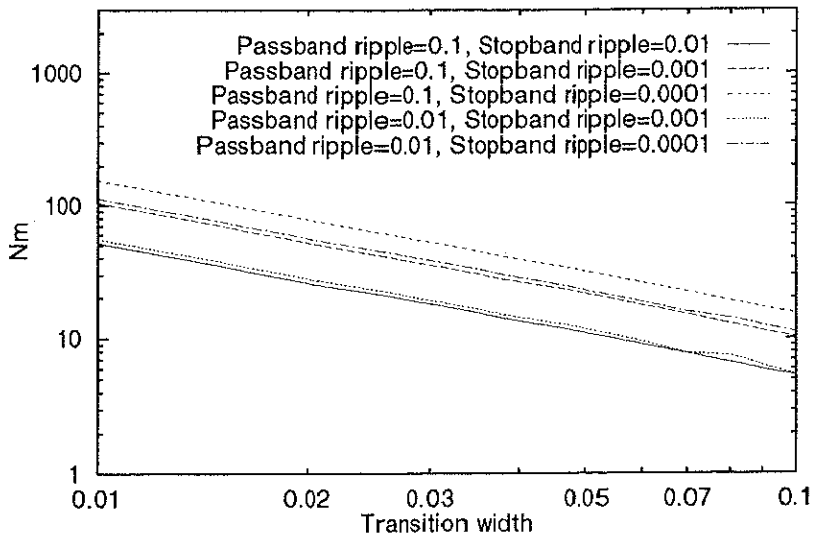


Figure 4.8: Behavior of the function N_m for some δ_p and δ_s as a function of $1/\Delta F$ (log-to-log scale).

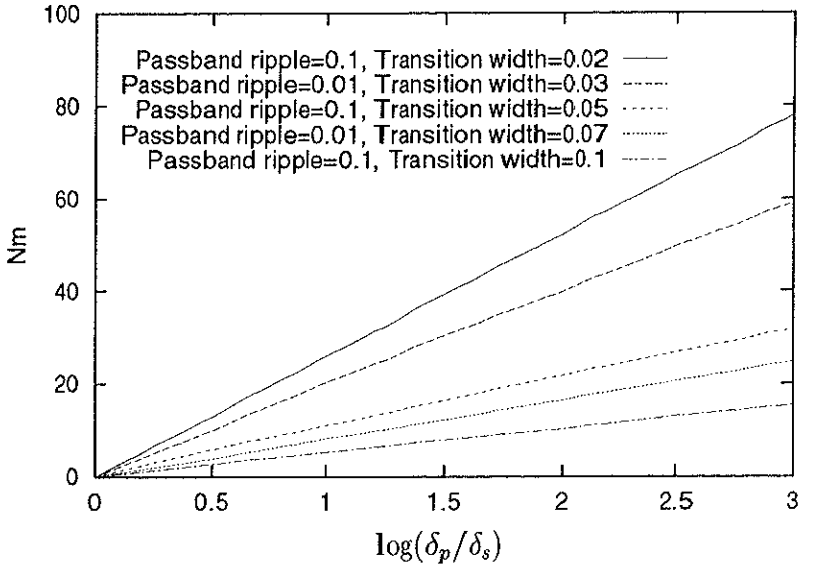


Figure 4.9: Behavior of the function N_m as a function of $\log(\delta_p/\delta_s)$.

Observation 4.14 For any δ_p, δ_s and ΔF, N_m is almost a linear function of $\log(\delta_p/\delta_s)$.

Based on Observations 4.13, 4.14 and some more experiments, we formulate the ap-

proximation $\hat{N}_m$ as

$$\hat{N}_m(\Delta F, \delta_p, \delta_s) = p \cdot \frac{\log_{10}(\delta_p/\delta_s)}{\Delta F} \cdot (-\log_{10}\delta_p)^q. \tag{4.17}$$

The coefficients p and q in (4.17) are also determined based on LMS approximation criterion in (4.15). In this case, millions of combinations of $\{f_p, \Delta F, \delta_p, \delta_s, N\}$ are used to determine the coefficients. Finally we have

$$p = 0.52, \quad q = 0.17.$$

4.3.2.4 The Proposed Estimation Formula $\hat{N}_4$ for $\delta_p \neq \delta_s$

As a result, the proposed estimation formula $\hat{N}_4$ can be summarized as follows.

$$\begin{aligned}
\hat{N}_4(f_p, \Delta F, \delta_p, \delta_s) &:= \hat{N}_3(f_p, \Delta F, \delta_p) + D\hat{N}(f_p, \Delta F, \delta_p, \delta_s), \\
D\hat{N}(f_p, \Delta F, \delta_p, \delta_s) &:= \left\lceil \hat{N}_m(\Delta F, \delta_p, \delta_s) \cdot \left\{ h(f_p, \Delta F, 1.1) \right.\right. \\
&\qquad \left.\left. - \frac{h(0.5 - \Delta F - f_p, \Delta F, 0.29) - 1}{2} \right\} \right\rceil, \\
\hat{N}_m(\Delta F, \delta_p, \delta_s) &:= 0.52 \cdot \frac{\log_{10}(\delta_p/\delta_s)}{\Delta F} \cdot (-\log_{10}\delta_p)^{0.17}, \\
h(f_p, \Delta F, c) &:= \frac{2}{\pi} \cdot \arctan\left\{ \frac{c}{\Delta F} \cdot \left(\frac{1}{f_p} - \frac{1}{0.5 - \Delta F} \right) \right\},
\end{aligned}$$

The formula $\hat{N}_4$ is a four-variable function of $f_p, \Delta F, \delta_p$ and δ_s.

4.3.3 For High-pass, Band-pass and Band-stop Filters

4.3.3.1 Design of High-pass Filters

In designing FIR high-pass filters, the proposed estimation formula (4.12) can be applied as follows.

Let F_s, F_p, δ_s and δ_p denote the stopband edge frequency, passband edge frequency ($0 \leq F_s < F_p \leq 0.5$), the stopband ripple and the passband ripple, respectively. In this

case, the estimation $\hat{N}_H$ of minimum filter length of the high-pass filter is given by:

$$\hat{N}_H = \hat{N}_4(0.5 - F_p, \Delta F_H, \delta_p, \delta_s)$$

where $\Delta F_H = F_p - F_s$. This result is based on the fact that the high-pass amplitude characteristics become the low-pass ones if the right side of the characteristics is turned to the left side.

4.3.3.2 Design of Band-pass Filters

In designing FIR band-pass filters, the proposed estimation formula (4.12) can be also applied as follows.

Suppose that the specifications of a band-pass filter are given in Table 4.1, where $0 \leq f_{s_1} < f_{p_1} \leq f_{p_2} < f_{s_2} \leq 0.5$ and $0 < \delta_s < \delta_p$. In this case, the estimation $\hat{N}_{BP}$ of the minimum filter length of the band-pass filter is given by:

$$\hat{N}_{BP} = \max\left\{\hat{N}_4(f_{p_2}, \Delta f_2, \delta_p, \delta_s),\ \hat{N}_4(0.5 - f_{p_1}, \Delta f_1, \delta_p, \delta_s)\right\} \tag{4.18}$$

where $\Delta f_1 = f_{p_1} - f_{s_1}$ and $\Delta f_2 = f_{s_2} - f_{p_2}$. This is based on our experimental results.

4.3.3.3 Design of Band-stop Filters

Specifications of a band-stop filter are given in Table 4.1, where $0 \leq F_{p_1} < F_{s_1} \leq F_{s_2} < F_{p_2} \leq 0.5$ and $0 < \delta_s < \delta_p$. In this case, the estimation $\hat{N}_{BS}$ of the minimum filter length of the band-stop filter is given by:

$$\hat{N}_{BS} = \max\left\{\hat{N}_4(F_{p_1}, \Delta F_1, \delta_p, \delta_s),\ \hat{N}_4(0.5 - F_{p_2}, \Delta F_2, \delta_p, \delta_s)\right\} \tag{4.19}$$

where $\Delta F_1 = F_{s_1} - F_{p_1}$ and $\Delta F_2 = F_{p_2} - F_{s_2}$. This is also based on our experimental results.

Table 4.1: Specifications of band-pass/stop filters.

	Band-pass	Band-stop
passband	$[f_{p_1}, f_{p_2}]$	$[0, F_{p_1}]$, $[F_{p_2}, 0.5]$
stopband	$[0, f_{s_1}]$, $[f_{s_2}, 0.5]$	$[F_{s_1}, F_{s_2}]$
passband ripple	δ_p	δ_p
stopband ripple	δ_s	δ_s

4.4 Evaluation

4.4.1 For Low-pass Filters with Identical Pass-band and Stop-band Ripples

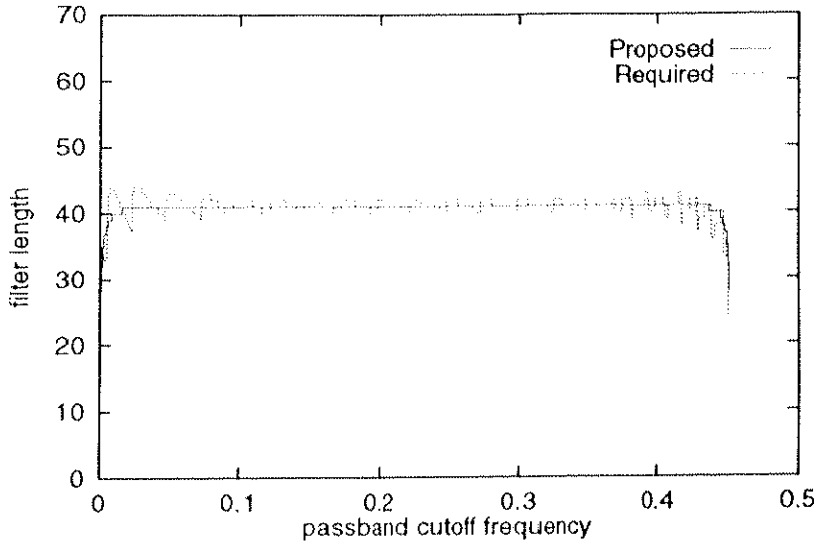
The proposed estimation formula (4.12) is evaluated in comparison with the conventional formulae (4.3) and (4.4). Figure 4.10 shows the behavior of the minimum filter length N and the estimated values $\hat{N}_i$ $(i = 1, 2, 3)$ as a function of f_p. Figure 4.10(a) shows the behavior for the case $\delta = 0.001$ and $\Delta F = 0.05$, and Fig. 4.10(b) for the case $\delta = 0.0001$ and $\Delta F = 0.1$. Figure 4.10 demonstrates that the estimated filter length $\hat{N}_3$ is better than the conventional estimations by (4.3) and (4.4) which are irrespective of f_p.

The estimation performance of (4.3) and (4.4) is evaluated by the following distance ΔN_i $(i = 1, 2, 3)$ defined by the normalized $\boldsymbol{L}^1$ norm:

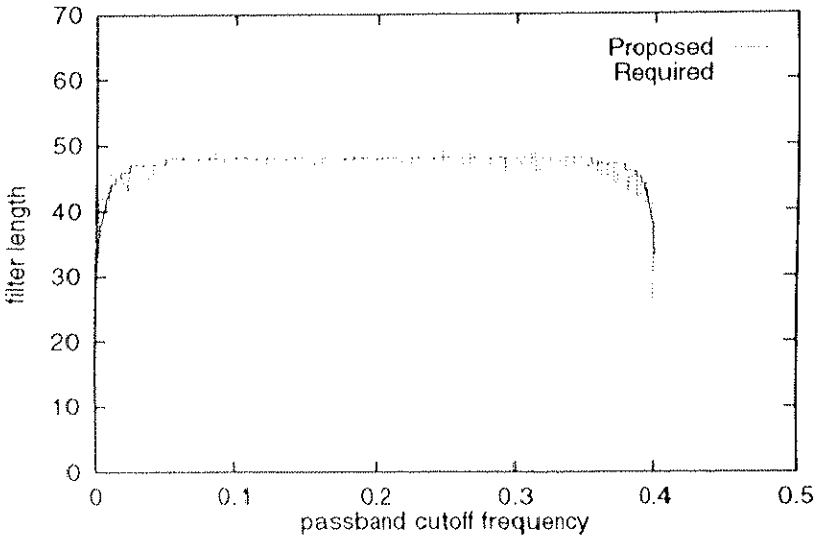
$$\Delta N_i(\Delta F, \delta) := \frac{1}{0.5 - \Delta F} \int_0^{0.5-\Delta F} |\hat{N}_i - N| \, df_p, \quad (i = 1, 2, 3). \tag{4.20}$$

In (4.20), ΔN_i means the average distance with respect to $f_p \in [0, 0.5 - \Delta F]$, which is a function of ΔF and δ.

Figure 4.11 shows the behavior of the distances ΔN_i $(i = 1, 2, 3)$ as a function of δ for some values of ΔF. In Fig. 4.11, we can see that the distance ΔN_3 stays around one and is smaller than both ΔN_1 and ΔN_2 for any ΔF and δ.



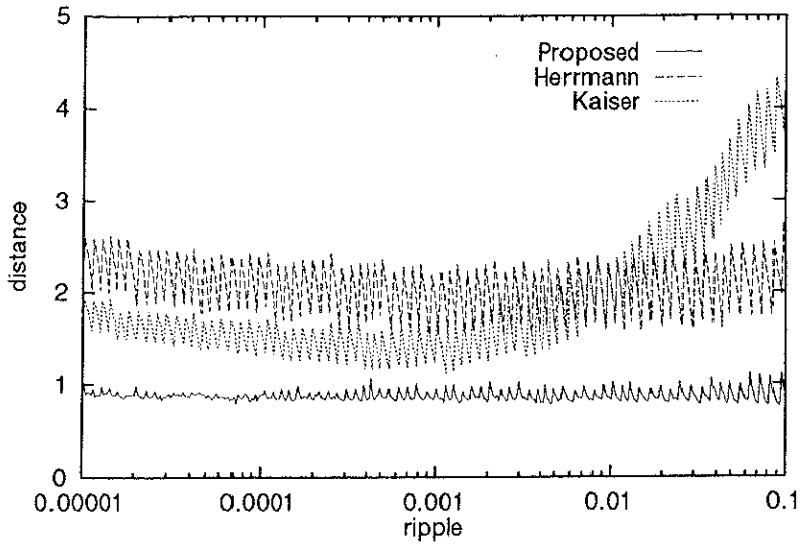
(a) In the case $\delta = 0.01$ and $\Delta F = 0.05$



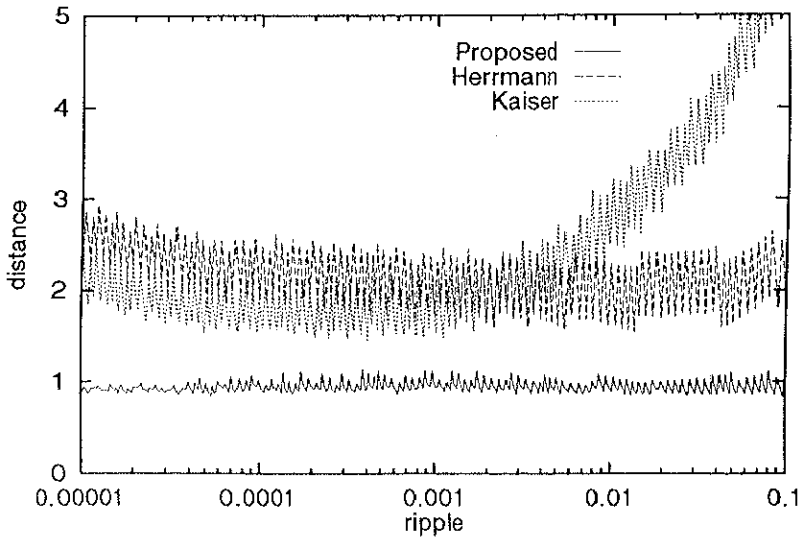
(b) In the case $\delta = 0.0001$ and $\Delta F = 0.1$

Figure 4.10: Behavior of the minimum filter length N and the estimated filter length $\hat{N}_3$ as a function of f_p.

Especially in the case where both ΔF and δ get smaller, in other words, when the minimum filter length N gets longer, the distances ΔN_1 and ΔN_2 become worse. This is because the conventional formulae (4.3) and (4.4) do not correspond to the FIRs of long filter length and some significant theoretical considerations were missed. Since the dis-



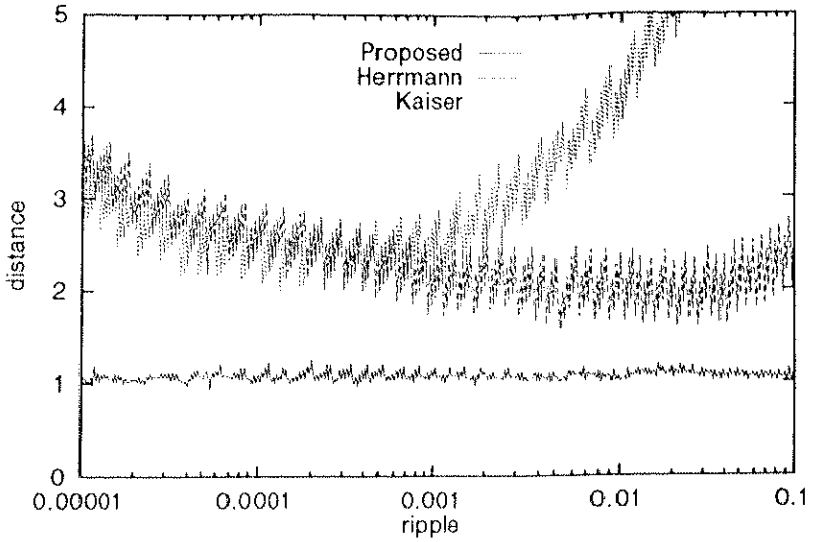
(a) In the case $\Delta F = 0.07$



(b) in the case $\Delta F = 0.05$

Figure 4.11: Behavior of the distances ΔN_i $(i = 1, 2, 3)$ as a function of δ (log-to-linear scale).

tance ΔN_3 stays around one for any ΔF and δ, it is obvious that the proposed estimation formulae (4.12) makes a good estimation for any case.



(c) in the case $\Delta F = 0.03$

Figure 4.11: (Continued).

4.4.2 For any Low-pass Filter

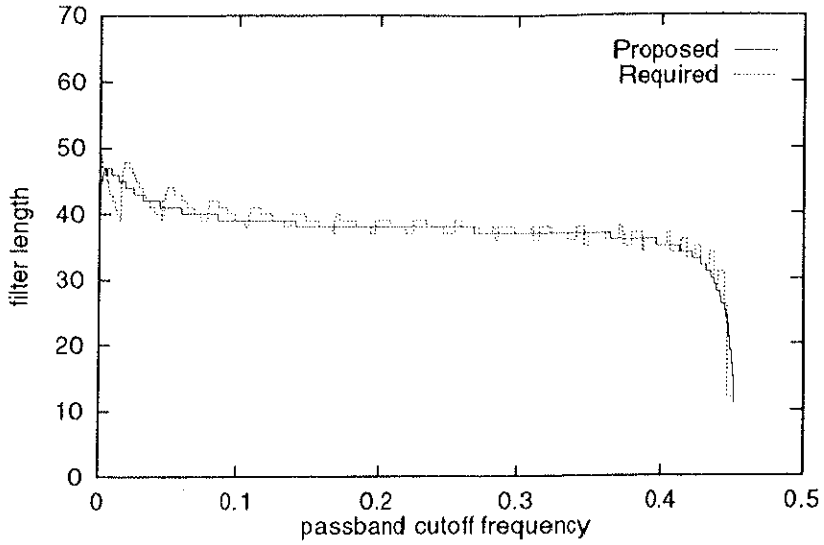
The proposed estimation formula (4.12) is evaluated in comparison with the conventional formulae (4.1) and (4.2).

Figure 4.12 shows the behavior of the estimated filter length $\hat{N}_4$ by (4.12) and the required minimum filter length N as a function of f_p. Specifications of Figs.4.12(a) and 4.12(b) are the same as those of Figs.4.4(a) and 4.4(b), respectively. Figures 4.4 and 4.12 demonstrate that the proposed estimation formula $\hat{N}_4$ is better than the conventional estimations by (4.1) and (4.2) which are constant irrespective to f_p.

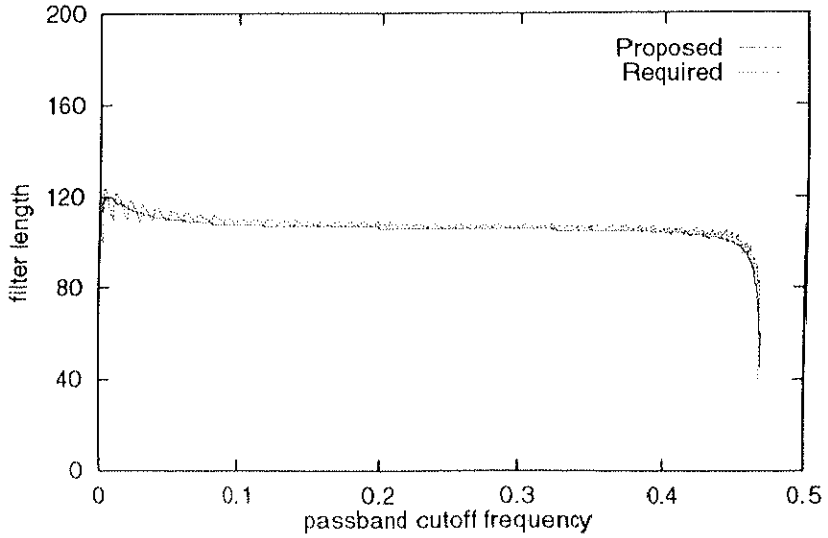
The estimation performance of (4.1) and (4.2) is evaluated by the following distance ΔN_i $(i = 1, 2, 4)$ defined by the normalized L^1 norm:

$$\Delta N_i(\Delta F, \delta_p, \delta_s) := \frac{1}{0.5 - \Delta F} \int_0^{0.5-\Delta F} |\hat{N}_i - N| \, df_p,$$

$$(i = 1, 2, 4). \tag{4.21}$$



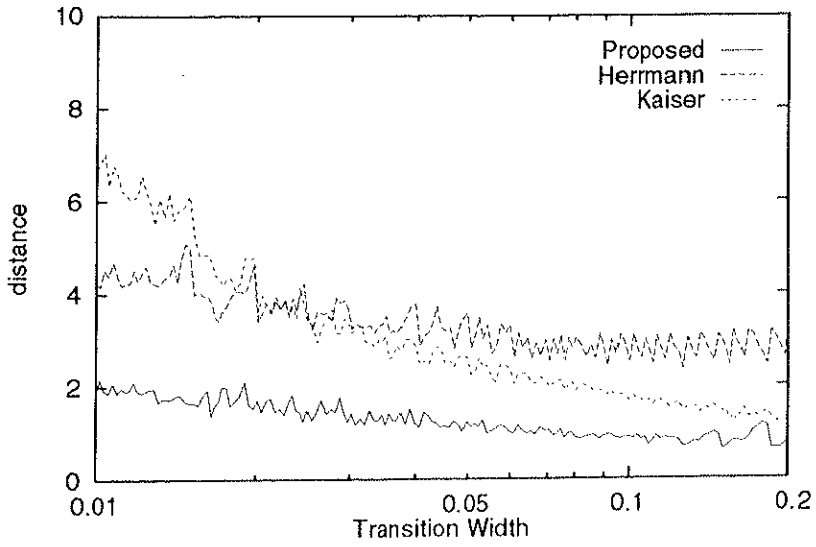
(a) In the case $\delta_p = 0.1$, $\delta_s = 0.001$ and $\Delta F = 0.03$



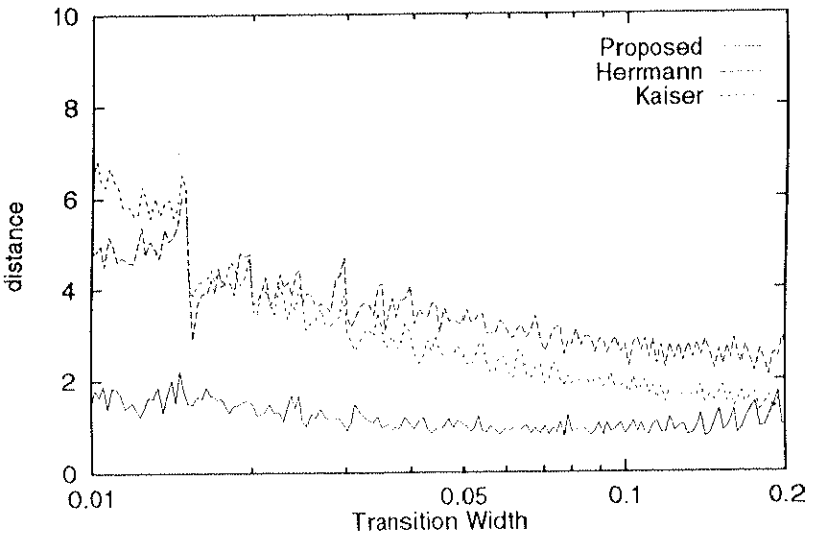
(b) In the case $\delta_p = 0.01$, $\delta_s = 0.0001$ and $\Delta F = 0.05$

Figure 4.12: Behavior of the estimated filter length $\hat{N}_4$ (real line) and the required minimum filter length N (broken line) as a function of f_p.

In (4.21), ΔN_i means the average distance with respect to $f_p \in [0, 0.5 - \Delta F]$. Figure 4.13 shows the behavior of the distances ΔN_i $(i = 1, 2, 4)$ as a function of ΔF for some values of δ_p and δ_s. In Fig.4.13, we can see that the distance ΔN_4 stays less than two and is smaller than both ΔN_1 and ΔN_2. This means that the proposed formula $\hat{N}_4$ makes



(a) In the case $\delta_p = 0.1$ and $\delta_s = 0.001$



(b) In the case $\delta_p = 0.01$ and $\delta_s = 0.0001$

Figure 4.13: Behavior of the distance ΔN_4 of the proposed estimation formula (solid line), the distance ΔN_1 of Herrmann's formula (broken line) and the distance ΔN_2 of Kaiser's formula (dotted line) as a function of ΔF.

better estimation than both $\hat{N}_1$ and $\hat{N}_2$.

4.4.3 For High-pass, Band-pass and Band-stop Filters

Since the conventional estimation formulae correspond only to the estimation for low-pass filter design, they are not applied to the design of other types of filters. Hence, the proposed estimation formula for high-pass, band-pass and band-stop filters are evaluated by comparing the estimated minimum filter length with the required one for some design examples.

4.4.3.1 For High-pass Filters

Consider designing the high-pass filter for which specifications are given in Table 4.2. The estimation $\hat{N}_H$ by the proposed formula is 33, and the required N is 35. The distance is only two in this case.

4.4.3.2 For Band-pass Filters

Specifications of the desired band-pass filter are given as Table 4.2. The estimation $\hat{N}_{BP}$ by the proposed formula (4.18) is $\max\{38, 91\} = 91$, and the required N is 92. The distance is only one in this case.

4.4.3.3 For Band-stop Filters

We next design the band-stop filter of which specifications are given in Table 4.2. The estimation $\hat{N}_{BS}$ by the proposed formula (4.19) is $\max\{106, 107\} = 107$, and the required N is 105. The distance is two in this case.

4.5 Summary

In this chapter, accurate estimation formulae were proposed for minimum filter length of optimum FIR digital filters. The formula was first established for low-pass filters with

Table 4.2: Specifications of high-pass, band-pass and band-stop filters.

	High-pass	Band-pass	Band-stop
passband	$[0.2, 0.5]$	$[0.10, 0.15]$	$[0, 0.17]$, $[0.25, 0.5]$
stopband	$[0, 0.1]$	$[0, 0.08]$, $[0.2, 0.5]$	$[0.2, 0.22]$
passband ripple	0.01	0.1	0.01
stopband ripple	0.0001	0.001	0.0001

identical pass-band and stop-band ripples, and then it was expanded for any low-pass filter. Moreover, the formula was applied to the design of high-pass, band-pass and band-stop filters.

It was shown that the proposed estimation formulae really perform with good accuracy. For designing low-pass filters, the proposed formula was evaluated in comparison with the conventional estimation formulae and the proposed one could make much smaller estimation distances than those of conventional ones. Also for designing other types of filters, it was confirmed that the proposed formula well approximate the required minimum filter length.

The estimation formulae (4.18) and (4.19) for band-pass and band-stop filters still have problems which may lead to wrong estimation. If we estimate the minimum filter length of a band-pass filter with the narrow passband (or, a band-stop filter with the narrow stopband), the proposed estimation formulae (4.18) and (4.19) tend to give longer filter length than required.

For example, we design the band-pass filter with the narrow passband for which specifications are given in Table 4.3. The estimation $\hat{N}_{BP}$ by the proposed formula (4.19) is $\max\{33, 82\} = 82$, and the required filter length N is 69. The estimation is 13 longer

than the required.

The band-stop filter with the narrow stopband is also designed. Specifications of the band-stop filter are given in Table 4.3. The estimation $\hat{N}_{BS}$ by the proposed formula (4.19) is $\max\{66, 83\} = 83$, and the required N is 65. The distance is 18 in this case.

Behavior of the minimum filter length of those filters would be studied further, and the accuracy of the estimation must be improved.

Table 4.3: Specifications of band-pass and band-stop filters.

	Band-pass	Band-stop
passband	$[0.20, 0.21]$	$[0, 0.2]$, $[0.3, 0.5]$
stopband	$[0, 0.10]$, $[0.25, 0.5]$	$[0.25, 0.26]$
passband ripple	0.01	0.01
stopband ripple	0.0001	0.0001

Chapter 5

Conclusions

5.1 Consideration of Results of the Present Study

A simple scheme of B-spline decomposition and reconstruction of continuous-time signals is studied in Chapter 2. The proposed scheme takes an important role of A–D and D–A conversions of continuous-time signals based on wavelet transforms. The scheme is designed mainly by digital circuits, employing only a few analog devices: two integrators and a zero-order hold circuit. Its analysis yielded conditions for the circuit parameters to avoid instability and overflow, and to meet the required precision. Moreover, the proposed scheme works as theoretically predicted in a computer simulation.

Frequency transformation matrices were proposed for analog and IIR digital filters in Chapter 3. The proposed matrices could replace the hand computation in those transformations by automatic procedures. Furthermore, an automatic procedure was established to design various types of IIR digital filters directly from an analog low-pass filter.

This chapter also studied some properties of the frequency transformation matrices for IIR digital filters. Some recursive relationships of the elements of the matrices were presented, and they helped deriving the fast algorithm to compute all the elements. The proposed fast algorithms required much fewer operations than the algorithms based on

the definitions of matrices.

The accurate estimation formulae were proposed in Chapter 4 for minimum filter length of optimum FIR digital filters. The proposed formula was applied to various types of filters, and the proposed estimation formulae really performed with good accuracy. The proposed formula was evaluated in comparison with the conventional estimation formulae for low-pass filter design, and the proposed formula realizes much smaller estimation distance than that of the conventional formulae. Also for designing other types of filters, it was confirmed that the proposed formula well approximate the required minimum filter length.

5.2 Problems Left for Future Research

As presented by the design example and the simulation in Chapter 2, we need such a high oversampling rate as 512 to make the error around 1%. This situation is almost the same even if we use sampled values of the B-splines because they are almost the same as the RRS functions with respect to approximating the B-splines. A faster digital signal processor should be realized to enable the implementation with the needed higher oversampling ratio.

It would be helpful for design if we have a formula expressing the exact period T in Fig. 2.7 for which the approximation error gets largest. Derivation of the formula seems very difficult but is important for a more advanced design procedure to be developed in further studies.

Frequency transformation matrices studied in Chapter 3 were only for the design of one-dimensional IIR digital filters. It could be applied for two-dimensional bilinear

and frequency transformations. If the automatic procedure for such transformations is established, it would be helpful for the design of two-dimensional digital filters, which are used for image processing.

The estimation formula proposed in Chapter 4 really improves the estimation accuracy of the conventional formulae. The estimation of the minimum filter length for differentiators and Hilbert transformers are studied by Rabiner and Schafer [32, 33] and realizes good accuracy. Thus the further study will be needed in the estimation of the minimum filter length for two-dimentional digital filters. However the optimal design method has not yet established. Similar to the design of one-dimentional IIR digital filters, optimal design methods must be first studied.

Acknowledgments

The author wishes to express his gratitude to the people who have directed his research and who have given him valuable comments, direction, and hospitality. His special debt of gratitude is due to Professor Makoto Natori, vice president at University of Tsukuba, for the sensible supervision and hospitality. Prof. Natori filled the post of chairperson of the two judgments for this dissertation. He owes a great deal to Professor Rokuya Ishii, in the Division of Electrical and Computer Engineering at Yokohama National University, for the valuable comments, direction, encouragement and hospitality. He is quite grateful to Professor Yukio Ishibashi, Professor Kazuhiro Hirasawa and Associate Professor Takashi Kitagawa, in the Institute of Information Sciences and Electronics at University of Tsukuba, for the valuable comments and directions as the examiners of this dissertation. He is most obliged to Associate Professor Masaru Kamada, in the Department of Computer and Information Sciences at Ibaraki University, for the valuable comments, helpful discussions, encouragement and hospitality. He would like to express his sincere gratitude to Assistant Professor Naohisa Otsuka in the Institute of Information Sciences and Electronics for the valuable comments, discussion, encouragement and hospitality. He is deeply indebted to Associate Professor Takahiko Horiuchi, in the Faculty of Software and Information Science at Iwate Prefectural University, for the helpful advice, encouragement and hospitality. His deep gratitude is due to Dr. Mamoru Iwaki, in the Graduate

School of Information Sciences at Japan Advanced Institute of Science and Technology (JAIST), for the valuable comments, advice and encouragement of the research of FIR filter design. He is very grateful to Dr. Steven Trautmann at Texas Instruments Co. Ltd. for the encouragement and for checking English grammar of this dissertation. He sincerely acknowledge the encouragement of Prof. N. K. Bose in the Department of Electrical Engineering at the Pennsylvania State University. The author has sent some of his papers to Professor Bose, and received replies from Professor which encouraged the author very much. Finally, the author would like to express his sincere thanks to all the members of Fundamental Mathematical Sciences Laboratory in the Institute of Information Sciences and Electronics for their helpful discussion and kindness.

References

The Author's Work

Journal Papers

[1] M. Kamada, T. Endoh, K. Ichige and K. Toraichi: "Quadratic Spline Interpolator," *Int'l Journal of Systems Science*, vol. 27, no. 10, pp. 977–983, Oct. 1996.

[2] K. Ichige and M. Kamada, "An Approximation for Discrete B-splines in Time Domain," *IEEE Signal Proc. Letters*, vol. 4, no. 3, pp. 82–84, Mar. 1997.

[3] K. Ichige, N. Otsuka and R. Ishii, "An Automatic Design Procedure of IIR Digital Filters from an Analog Low-Pass Filter," *Signal Processing*, vol. 57, no. 3, pp. 223–231, Mar. 1997.

[4] K. Ichige and M. Kamada: "A Simple Discrete Version of B-splines," *Int'l Journal of Systems Science*, vol. 29, no. 3, pp. 335–340, Mar. 1998.

[5] K. Ichige, M. Kamada and R. Ishii: "A Simple Scheme of Decomposing and Reconstructing Continuous-time Signals by B-splines," *IEICE Trans. Fundamentals*, vol. E80-A, no. 11, pp. 2391–2399, Nov. 1998.

[6] K. Ichige, N. Otsuka and R. Ishii: "Properties of The Frequency Transformation

Matrices for IIR Digital Filters," *Signal Processing*, vol. 71, no.3, pp. 227–233, Dec. 1998.

[7] N. Otsuka, K. Ichige, M. Yurube, K. Shiomi and R. Ishii: "The Schur Stability of Real Weighted Diamond Polynomials," *Trans. IEICE Japan*, vol. J82–A, no. 3, Mar. 1999, (in Japanese), (in press).

[8] K. Ichige, M. Iwaki and R. Ishii: "Accurate Estimation of Minimum Filter Length for Optimum FIR Low-pass Digital Filters," (submitted for publication).

[9] K. Ichige, M. Iwaki and R. Ishii: "Accurate Estimation of Minimum Filter Length for Optimum FIR Digital Filters," (submitted for publication).

Presentations at International Conferences

[10] K. Shiomi, M. Kamada, K. Toraichi, T. Endoh and K. Ichige: "A Smooth Signal Generator by Quadratic Spline Interpolation," *Proc. IASTED Int'l Conf. AEN'94*, pp. 56–59, Zurich, Switzerland, July 1994.

[11] K. Ichige, M. Kamada, K. Toraichi and R. Ishii: "A Function Generator Using the Oversampling Filter Based on Quadratic B-spline Functions," *Proc. IASTED Int'l Conf. AEN'94*, pp. 60–63, Zurich, Switzerland, July 1994.

[12] K. Ichige, R. Ishii and N. Otsuka: "Matrices on the Coefficient Space Isomorphic to the Frequency Transformation of IIR Digital Filters," *Proc. 6th Int'l Conf. on Signal Processing Applications and Technology*, pp. 634–638, Boston, MA, U.S.A., Oct. 1995.

[13] K. Ichige, M. Kamada and R. Ishii: "A Scheme of Decomposition and Reconstruction of Continuous-time Signals by B-spline Functions," Presented at *Math. Theory of Networks and Systems '96*, St. Louis, MO, U.S.A, June 1996.

[14] K. Ichige and M. Kamada: "Discrete B-spline Functions," *Proc. 8th European Signal Processing Conf.*, pp. 871–874, Trieste, Italy, Sep. 1996.

[15] K. Ichige, K. Ueda, M. Iwaki and R. Ishii: "An Accurate Estimation Formula for Minimum Filter Length of Optimum FIR Low-pass Digital Filters," *Proc. 1st Int'l Conf. on Info., Comm. and Signal Processing*, pp. 1303 1307, Singapore, Sep. 1997.

[16] K. Ichige, M. Kamada and R. Ishii: "A Simple Scheme of Decomposition and Reconstruction of Continuous-time Signals by B-splines," *Proc. 1998 IEEE Int'l Sympo. on Circuits and Systems*, WPA1-2, Monterey, CA, U.S.A, May–June, 1998.

[17] K. Ichige, M. Kamada, N. Otsuka and R. Ishii: "A Method of Obtaining a Digital Simulator of Continuous-time Systems for Computer Simulation," *Proc. IFAC/SICE Workshop on Control in Natural Disasters*, pp. 29 33, Tokyo, Japan, Sep. 1998.

[18] K. Ichige, M. Iwaki and R. Ishii: "A New Estimation Formula for Minimum Filter Length of Optimum FIR Digital Filters," *Proc. 4th Int'l Conf. on Signal Processing*, pp. 89–92, Beijing, China, Oct. 1998.

Other References

[19] J. W. Adams and A. N. Wilson, Jr., "A New Approach to FIR Digital Filters with Fewer Multipliers and Reduced Sensitivity," *IEEE Trans. Circuits and Systems*, vol. 30, no. 5, pp. 277–283, May 1983.

[20] N. K. Bose: "Properties of the Q_n–matrix in Bilinear Transformation", *Proc. IEEE*, vol. 71, no. 9, pp. 1110–1111, Sep. 1983.

[21] V. Cappellini, A. G. Constantinides and P. Emiliani, *Digital Filters and Their Applications*, Academic Press, Orlando, FL, 1978.

[22] C. K. Chui, *An Introduction to Wavelets*, Academic Press, Orlando, FL, 1992.

[23] A. G. Constantinides: "Spectral Transformations for Digital Filters", *Proc. IEE*, vol. 117, no. 8, pp. 1585–1590, Aug. 1970.

[24] O. Herrmann, L. R. Rabiner and D. S. K. Chan: "Practical Design Rules for Optimum Finite Impulse Response Low-pass Digital Filters," *Bell Syst. Tech. J.*, vol. 52, no. 6, pp. 769–799, July–Aug. 1973.

[25] R. Ishii: "S–Z Transformations for Digital Filters", *Electronics Letters*, vol. 19, no. 9, pp. 350–352, Apr. 1983.

[26] L. B. Jackson, J. F. Kaiser and H. S. McDonald, "An Approach to the Implementation of Digital Filters," *IEEE Trans. Audio and Electroacoustics*, vol. 16, no. 3, pp.413–421, Sep. 1968.

[27] J. F. Kaiser: "Nonrecursive Digital Filter Design Using I_0-sinh Window Function," *Proc. IEEE Int'l Sympo. Circuits and Systems*, pp. 20–23, Apr. 1974.

[28] T. W. Parks and J. H. McClellan: "Chebyshev Approximation for Nonrecursive Digital Filters with Linear Phase," *IEEE Trans. Circuit Theory*, vol. CT–19, no. 5, pp. 189–194, Mar. 1972.

[29] L. R. Rabiner: "Approximate Design Relationships for Low-Pass FIR Digital Filters," *IEEE Trans. Audio and Electroacoustics*, vol. AU–21, no. 5, pp. 456–460, Oct. 1973.

[30] L. R. Rabiner and B. Gold: *Theory and Application of Digital Signal Processing*, Prentice-hall, Englewood Cliffs, NJ, 1975.

[31] L. R. Rabiner and O. Herrmann: "On the Design of Optimum FIR Low-Pass Filters with Even Impulse Response Duration," *IEEE Trans. Audio and Electroacoustics*, vol. AU–21, no. 4, pp. 329–336, Aug. 1973.

[32] L. R. Rabiner and R. W. Schafer: "On the Behavior of Minimax Relative Error FIR Digital Differentiators," *Bell Syst. Tech. J.*, vol. 53, no. 2, pp. 333–361, Feb. 1974.

[33] L. R. Rabiner and R. W. Schafer: "On the Behavior of Minimax FIR Hilbert Transformers," *Bell Syst. Tech. J.*, vol. 53, no. 2, pp. 363–390, Feb. 1974.

[34] E. Ya. Remez: "General Communication Methods of Chebyshev Approximation," Kiev, USSR: *Atomic Energy Translation 4491*, pp. 1–85, 1957.

[35] T. Saramäki, T. Karema, T. Ritoniemi and H. Tenhunen, "Multiplier-Free Decimator Algorithms for Superresolution Oversampled Converters," *Proc. 1990 IEEE Int'l Symposium on Circuits and Systems*, vol. 4, pp. 3275–3278, May 1990.

[36] T. Saramäki, Y. Neuvo and S. K. Mitra, "Design of Computationally Efficient Interpolated FIR Filters," *IEEE Trans. Circuits and Systems*, vol. 35, no. 1, pp. 70–88, Jan. 1988.

[37] I. J. Schoenberg, *Cardinal Spline Interpolation*, SIAM, Philadelphia, 1973.

[38] M. Unser, A. Aldroubi and M. Eden, "Fast B-spline Transforms for Continuous Image Representation and Interpolation," *IEEE Trans. Pattern Anal. and Machine Intell.*, vol. 13, no. 3, pp. 277–285, Mar. 1991.

Vita

Koichi Ichige was born in Hitachinaka (former Katsuta), Ibaraki, Japan, on February 22, 1972, the son of Katsuhiko Ichige and Sadako Ichige. He entered University of Tsukuba (College of Information Sciences), Tsukuba, Ibaraki, Japan, in April 1990. He received the degrees of Bachelor and Master of engineering from University of Tsukuba in 1994 and 1996, respectively.