TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY
Julkaisu 653 • Publication 653

Peyman Arian

Computationally Efficient Decimators, Interpolators, and Narrow Transition-Band Linear-Phase Finite Impulse Response (FIR) Filters

Tampere 2007

Tampereen teknillinen yliopisto. Julkaisu 653
Tampere University of Technology. Publication 653

Peyman Arian

Computationally Efficient Decimators, Interpolators, and Narrow Transition-Band Linear-Phase Finite Impulse Response (FIR) Filters

Thesis for the degree of Doctor of Technology to be presented with due permission for public examination and criticism in Tietotalo Building, Auditorium TB109, at Tampere University of Technology, on the 23rd of February 2007, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2007

ISBN 978-952-15-1722-8 (printed)
ISBN 978-952-15-1800-3 (PDF)
ISSN 1459-2045

Table of Contents

Table of Contents i

Abstract iii

Acknowledgements v

List of Abbreviations and Symbols vii

1 Introduction 1
- 1.1 Background . . . 1
- 1.2 Author's Contributions . . . 7
- 1.3 Thesis Outline . . . 9

2 Computationally Efficient Decimators 11
- 2.1 Multiple Branch Decimators . . . 11
  - 2.1.1 The Transfer Function . . . 12
  - 2.1.2 The Optimization Problem . . . 17
  - 2.1.3 The Optimization Algorithm . . . 19
  - 2.1.4 Performance Study . . . 20
- 2.2 Single-Stage Two-Filter Decimators . . . 24
  - 2.2.1 The Transfer Function and the Zero-Phase Response . . . 27
  - 2.2.2 The Optimization Problem . . . 28
  - 2.2.3 The Optimization Algorithm . . . 28
  - 2.2.4 The Main Algorithm . . . 29
  - 2.2.5 Order Estimation . . . 31
  - 2.2.6 Performance Study . . . 36
- 2.3 Hybrid Decimators . . . 40
  - 2.3.1 The Transfer Function . . . 41
  - 2.3.2 The Optimization Problem and the Optimization Algorithm . . . 43

2.3.3 Multiplierless Designs 44
2.3.4 Performance Study 45

3 Cascade Structures for Generating Sharp Linear-Phase FIR Filters 51
3.1 Review of Principle of Switching and Resetting 52
3.2 Alternative Structures 55
3.2.1 Cascade Structure I 56
3.2.2 Cascade Structure II 59
3.3 Filter Implementation 62
3.3.1 Cascade Structure I 62
3.3.2 Cascade Structure II 76
3.4 Noise Analysis 84
3.4.1 Cascade Structure I 86
3.4.2 Cascade Structure II 88
3.5 Performance study 91
3.5.1 The Prototype Filter Specifications 92
3.5.2 Simulation Results 93

4 Parallel Structures for Generating Sharp Linear-Phase FIR Filters 105
4.1 Alternative Structures 106
4.1.1 Parallel Structure I 109
4.1.2 Parallel Structure II 109
4.1.3 Parallel Structure III 110
4.2 Filter Implementation 111
4.2.1 Parallel Structure II 112
4.2.2 Parallel Structure III 125
4.3 Noise Analysis 129
4.3.1 Parallel Structure II 131
4.3.2 Parallel Structure III 134
4.4 Performance study 136
4.4.1 Simulation Results 137

5 Concluding Remarks 149

A Derivation of Some Formulae in Chapter 3 151

B Derivation of Some Formulae in Chapter 4 181

Bibliography 217

Abstract

In many digital-filtering applications, it is crucial that the shape of the waveform to be filtered is preserved. This desirable property is owned by a class of digital filters, collectively referred to as linear-phase digital filters.

In very large scale integration (VLSI) applications, the designs requiring the smallest number of multipliers are of particular interest, since depending on the application, less multipliers is tantamount to a smaller or a less power consuming VLSI chip. To this end, there has been a constant effort to come up with designs requiring minimum possible number of multipliers to meet a predefined specification.

Two important classes of digital filters, decimators and interpolators, have been a focus of the above-mentioned effort. Decimators and interpolators are integral parts of a multirate digital signal processing (DSP) system, and because of vast applications of such systems, decimators and interpolators are found virtually in every DSP-utilizing scheme. It should therefore come as no surprise that a lot of designs have been directed to the design of these two important classes of digital filters.

Digital filters with narrow transition band form another important class of digital filters. As the transition band becomes narrower, the required number of multipliers for meeting the specifications increases, and with conventional methods, the required number of multipliers becomes extremely large and the design becomes impractical. That is why efficient implementation of narrow transition-band digital filters have been a focus of intense research.

This work introduces techniques for efficient design of a class of decimators, interpolators and narrow transition-band linear-phase finite impulse response (FIR) filters. Linear-phase one-stage decimators and interpolators are the focus of the first part of the work. The design of this class of digital filters has been addressed as an optimization problem, and an algorithm to solve the problem has been proposed. Next, multiple branch linear-phase decimators have been introduced, and the ideas of a multiple branch design and a one-stage design have been combined to give rise to a hybrid structure. An algorithm leading to an optimum such structure has been proposed, and further constraints have been imposed to yield a structure with the least possible number of multipliers.

The focus of the second part of this work is on the design of narrow transition-band linear-phase FIR filters. The efficiency of the design stems from the fact that infinite impulse response (IIR) filters have been exploited. In essence, the design is a cascade of a stable IIR and an unstable IIR filter. To overcome the adverse effects of roundoff noise, the principle of switching and resetting has been employed. To curb the noise further and to reduce the number of required components, two decomposition schemes has been proposed. The noise generated by the structure has been analyzed in detail, and closed form formulae to measure the noise have been put forward. Finally, the design of this class of filters is addressed as an optimization problem, and a method to find the initial point of the optimizing algorithm is proposed.

The third part of the thesis takes an alternative approach for the design of narrow transition-band linear-phase FIR filters. In this approach, partial fraction expansion is applied to the cascade of a stable IIR filter and its unstable counterpart, but now the transfer function is expressed in terms of $z + z^{-1}$. By factorizing the proposed structure and using the principle of switching and resetting, the filter implementation is discussed. The noise generated by the structure is analyzed in detail, and it is proved that the performance of the proposed structure is not impaired by the generated noise. The efficiency of all the proposed structure have been supported by numerical simulations. When compared with alternative methods, the results of the simulations clearly indicate the attractiveness and potentials of the proposed structures.

Acknowledgements

This work has been carried out at the Institute of Signal Processing, Tampere University of Technology, Tampere, Finland during 2002-2007.

I would like to thank my dear supervisor, Professor Tapio Saramäki for keeping the research work exciting from the beginning to the very last minute. He is a man with a heart of gold, and an exceptional scientist, who never failed to believe in me. Without his brilliant ideas, this work could have never been accomplished. My thanks extend to Professor Paulo Diniz and Professor Vesa Välimäki for their invaluable and constructive comments on the manuscript of the thesis.

I am forever grateful to the Head of the Institute of Signal Processing, Professor Moncef Gabbouj, for lending me a helping hand when I mostly needed. His unconditional assistance and management skills were most crucial to make this dissertation happen.

During the past years, I had the pleasure of sharing my room with Andrey Norkin, Djordje Babić, Dmitriy Paliy, Francisco Lopez, Jussi Vesma, Markku Ekonen, Pilar Martín Martín, Samuli Harju-Jeanty, Tuomo Kuusisto and Professor Vladimir Katkovnik. I would like to thank all of them for providing an enjoyable and relaxed workroom atmosphere.

I would like to thank all my colleagues at Tampere University of Technology for their contribution to the pleasant environment. In particular, I am thankful to Alessandro Foi, Atanas Boev, Daniel Nicorici, Gergely Korodi, Juha Yli-Kaakinen, Professor Karen Egiazarian, Pekka Uotila and Robert Bregović for their company and friendship.

Separate words of gratitude I owe to our system support staff Antti Orava, Jari Pärssinen, Maarit Ahonen-Lehmus, Pentti Jokela, Riikka Löytäinen, Sami Lanu, Tommi Kuisma and

our outstanding secretaries Ms. Elina Orava, Ms. Kirsi Järnström, Ms. Pirkko Ruotsalainen, and Ms. Virve Larmila for their excellent job and unfailing cooperation.

Since August 2006, I was most fortunate to work with my colleagues Asko Valli, Juha Lindfors, Jukka Liukkonen, Pertti Toivonen and Tenho Vanhanen at Helsinki Polytechnic Stadia. I would like to thank all of them for their trust, kindness, and cooperation. I would also wish to thank Pearson Education and in particular Ms. Heidi Strömmer for the priceless assistance they have provided me for my teaching.

I am thankful to all my Persian friends in Finland for the gift of their friendship. In particular I was blessed to have people like Babak Soltanian, Mehdi Rezaei and Sasan Iraji around at TUT. Their friendship is invaluably dear to me and I cherish every moment of their company.

The financial support of Tampere Graduate School of Information Science and Engineering (TISE) and Nokia Foundation is gratefully acknowledged. I am especially thankful to the Director of TISE Professor Markku Renfors for his constant support and Dr. Pertti Koivisto for his invaluable help and guidance during the course of my thesis.

I wish to express my sincere gratitude to all my loved ones, including my mother-in-law Raili Harjunpää for her care and compassion. I also owe a debt of gratitude to my country Finland for providing me a righteous society to grow and flourish.

Last but not least, my profoundest gratitudes belong to my wife Ulrika Harjunpää and my mother Mänije Moghäddäm Kärimi. Mom, I wished you were here to witness my dissertation, I have missed you so much. And Ulrika, I don't find words to tell you what you mean to me and how thankful I am for having you beside me. Every virtue in me is because I had you two angels in my life.

Tampere, February 2007

Peyman Arian

List of Abbreviations and Symbols

1S2F	Single-Stage Two-Filter Decimator
DSP	Digital Signal Processing
FIR	Finite Impulse Response
IIR	Infinite Impulse Response
MBD	Multiple Branch Decimator
NBLP	Narrow Transition Band Linear-Phase
RHS	Right Hand Side
SQP	Sequential Quadratic Programming
VLSI	Very Large Scale Integration
$\lfloor x \rfloor$	The integer part of x
$\lceil x \rceil$	The smallest integer larger than or equal to x
σ_e^2	Noise variance of each quantizer
$\mathbf{x}$	Vector $\mathbf{x}$
$\mathbf{x}^T$	The transpose of vector $\mathbf{x}$
x^*	The complex conjugate of x
$\otimes$	Convolution operation
D	Sampling rate conversion
δ_p	Maximum allowable ripple in the passband
δ_s	Maximum allowable ripple in the stopband
ω_p	Passband edge
ω_s	Stopband edge

Chapter 1

Introduction

1.1 Background

Digital filters, based on the length of their impulse response, are divided into finite-impulse response (FIR) and infinite-impulse response (IIR) classes (For a comparison between FIR and IIR filters, the reader is encouraged to consult [80]). Since the birth of digital filtering (arguably by Kaiser [52, 101]), different schemes have been employed to design FIR and IIR filters.

In almost all digital signal processing (DSP) applications, it is desired that the delay imposed on every single sinusoidal component of the input signal is constant. Filters having such property are called linear-phase filters, and can be designed by many different approaches [5, 6, 21, 22, 64, 73, 75, 83, 95, 100].

The landmark work of Parks and Mcclellan [66, 74] introduced a technique based on Remez multiple exchange algorithm [82] to optimally (in the minimax sense) design four types of linear-phase FIR filters, which were originally introduced by Nowak [70]. The major drawback of linear-phase FIR filters is that they require, especially in applications demanding narrow transition bands, considerably more arithmetic operations and hardware

components than their IIR equivalents. The minimum order of an optimum linear-phase FIR filter $H(z)$ meeting the low-pass filter specifications

$$1-\delta_p \le |H(e^{j\omega})| \le 1+\delta_p \quad \text{for} \quad \omega \in [0,\omega_p] \tag{1.1a}$$

$$|H(e^{j\omega})| \le \delta_{\rm s} \quad \text{for} \quad \omega \in [\omega_s,\pi] \tag{1.1b}$$

is approximately [53]

$$N = \frac{-20\log_{10}\sqrt{\delta_p\delta_{\rm s}}-13}{14.6(\omega_{\rm s}-\omega_p)/(2\pi)}+1. \tag{1.1c}$$

For a more accurate estimate, see [46]. From the above estimate, it is seen that as the transition bandwidth $\omega_{\rm s}-\omega_p$ is made smaller, the required filter length increases inversely proportionally to it. Since the direct-form implementation exploiting the coefficient symmetry requires approximately $N/2$ multipliers, this kind of implementation becomes very costly in narrow transition-band applications.

For a narrow transition-band case, the order N of an IIR digital filters to meet the specifications of (1.1a) and (1.1b) is considerably lower than that given by (1.1c). However, IIR digital filers always introduce some phase distortion [43, 57], and therefore causal linear-phase IIR digital filters cannot be designed.

The most straightforward approach to exploit the efficiency of IIR filters to yield linear phase is to use an allpass IIR equalizer in cascade with an IIR filter, which satisfies the amplitude response requirements but distorts the phase [79]. It turns out that in such cases, the phase response of the amplitude response satisfying IIR is quite nonlinear, and therefore a very high-order IIR equalizer is required [96].[1] In fact when constant group delay is

[1]It has been shown that an approximately linear-phase single IIR filter is more beneficial than using an extra, phase-equalizing IIR filter [40, 50, 51, 59, 60].

required in the passband, a direct optimum linear-phase FIR filter is more efficient than the aforementioned two-IIR cascade scheme [80, 97].

Another approach to use IIR filters to yield efficient linear-phase digital filters is based on data-reversal schemes [19, 42, 57]. In a data-reversal scheme, the input is filtered by an IIR filter and the output is stored. The stored output is subsequently reversed in time and fed to the same IIR filter to yield the final output. In order to implement the aforementioned scheme in real time, different techniques based on sectioning of the input signal are proposed [33–35, 44, 48, 57, 58, 77, 79, 103]. In a sectioning scheme, the input is parsed into shorter blocks, and every second block is fed to a different copy of the same IIR filter. Despite their efficiency, digital filters based on data-reversal schemes suffer from large group delays, resulting from their inherent time reverse circuitry.

In addition to using IIR filters, several authors have observed that by increasing the FIR filter length slightly from the minimum, significant savings in the number of multipliers and, with some methods, also in the number of adders can be achieved [1–3, 20, 23, 25, 26, 36, 37, 49, 54–56, 62, 67–69, 84, 88, 91, 98, 99].

This is due to the fact that optimal direct-form FIR filters are in a way too general structures to implement typical frequency selective filters. In the direct-form implementation, each multiplier determines the value of one impulse response sample independently of the other samples. In the linear-phase implementation, the same is true for approximately half of the impulse response values. However, in practical frequency selective filters there is a relatively strong correlation among neighboring impulse response values. By developing filter structures that exploit this correlation, the number of multipliers required in the implementation can be drastically reduced.

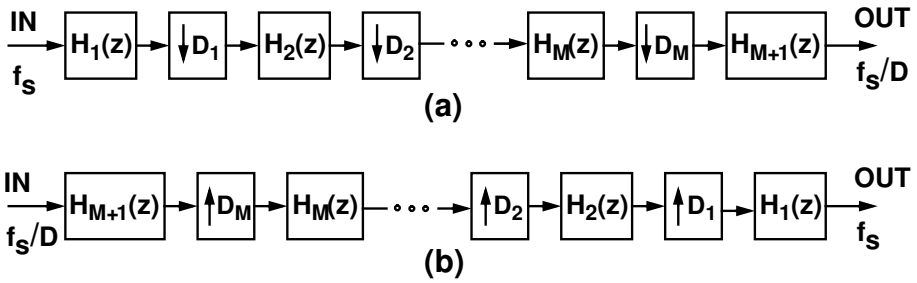

Figure 1.1: *(a) A general realization of a D-to-1 decimator using K decimation stages and one stage for baseband signal shaping. $D = D_1 D_2 \ldots D_M$. (b) An equivalent realization of a 1-to-D interpolator.*

Decimators and interpolators[2] are one of the first solutions proposed to improve the computational efficiency of a digital filtering designing scheme through the above-mentioned principle [18, 32, 41, 78]. Soon after their introduction, a number of different digital filter transfer functions and certain special filter structures for decimation and interpolation purposes emerged [17, 18, 24, 27, 30–32, 65, 81, 85–87, 93, 94]. These designs include both single-stage and multistage finite-impulse response (FIR) filters and infinite-impulse response (IIR) filters.

FIR decimators[3] and interpolators provide several advantages such as guaranteed stability, absence of limit cycles, and linear phase, if desired, compared to their IIR equivalents. A linear-phase response, for instance, is very important in applications where the envelopes of the time waveforms being decimated or interpolated are desired to be preserved. The major advantage of IIR filters over their FIR counterparts is a lower number of multipliers, adders and delay elements required.

[2]For a review of basic concepts of interpolation, see, for instance, [93].

[3]In this contribution, the main focus is laid on designing of decimators. In the sequel, a decimator is defined to be a lowpass filter followed by down-sampling by a given factor. The results can be applied directly to interpolators with very slight modifications, since decimators and interpolators are dual structures.

Decimators and interpolators with the sampling rate alteration factor of D can be implemented using $K+1$ stages, provided that D can be factored into the product

$$D = \prod_{k=1}^{M} D_k, \tag{1.2}$$

where each D_k, $k = 1, 2, \ldots, M$, is an integer. The implementations for such decimators and interpolators are shown in Figs. 1(a) and 1(b), respectively. In the case of conventional multistage FIR decimators [29, 30], the last stage is absent. In the structure proposed in [24, 86, 87], the last stage is present and all filter stages are linear-phase FIR filters. The comparisons given in [87] have shown that the best FIR designs in terms of the minimized number of multiplications per input sample rate are obtained by designing the filter stages $H_k(z)$ for $k = 1, 2, \ldots, M$ such that all their zeros lie on the unit circle and avoid aliasing into the passband and into a part of the transition band. The last stage $H_{M+1}(z)$ then shapes the overall passband response and takes care of the aliasing into the remaining part of the transition band.

A decimator can alternatively be realized as a polyphase structure. The polyphase decomposition of the FIR filter with transfer function

$$H(z) = \sum_{n=0}^{N} h(n) z^{-n}$$

is expressible as [4]

$$H(z) = \sum_{k=1}^{D} A_k(z) B_k(z^D), \tag{1.3a}$$

where

$$A_k(z) = z^{-(k-1)}, \tag{1.3b}$$

and

$$B_k(z) = \sum_{l=0}^{\lfloor (N-k+1)/D \rfloor} h(k-1+lD) z^{-l}, \tag{1.3c}$$

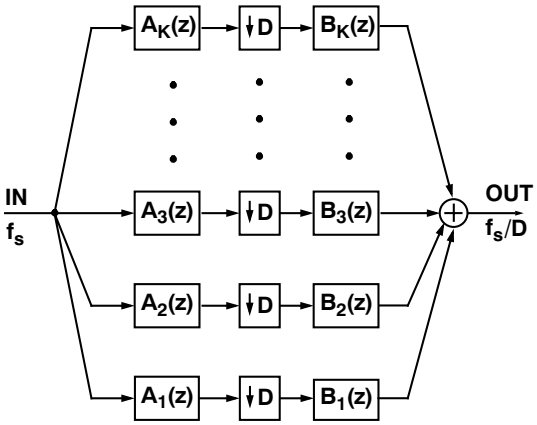

Figure 1.2: *Multiple branch FIR filter structure for a D-to-1 decimator.*

where $\lfloor x \rfloor$ stands for the integer part of x. The polyphase decomposition of $H(z)$ according to (1.3) consists of D parallel branches. Moreover, as seen from (1.3c), the FIR filters $B_k(z)$ are not guaranteed to be linear-phase.

Based on polyphase decomposition of FIR filters, a class of linear-phase FIR decimators has been introduced [90]. To reduce the arithmetic complexity compared to the polyphase structure, both $A_k(z)$'s and $B_k(z^D)$'s have been designed to be linear-phase FIR filters. $B_k(z^D)$'s are implemented at the output sampling rate as $B_k(z)$'s. Moreover, the number of branches have been reduced from D to K, with $K = 2$ or 3. Throughout this work, the proposed structure, presented in Fig. 1.2, is referred to as a *multiple branch decimator* (MBD). In an MBD, the number of delay elements and the number of multipliers can be minimized using the direct-form structure exploiting the coefficient symmetry for the $A_k(z)$'s and the $B_k(z)$'s, and the fact that $A_k(z)$'s, equally well as $B_k(z)$'s, can share the same delay elements. Efficient implementation of single-stage ($M = 1$ in (1.2)) structures presented in Figs. 1.1 and 1.2 is the subject of the first part of this work. In particular, the case $M = 1$ and $K = 1$ is referred to as *single-stage two-filter decimator* (1S2F) and has been considered in Section 2.2.

Most techniques developed so far for exploiting the correlation between the impulse response values mentioned earlier are based on the use of nonrecursive FIR subfilters, but techniques for building filters using recursive FIR structures also exist [38, 89, 102]. The subject of the second part of this work is introducing a new technique for exploiting even further the aforementioned correlation, using recursive FIR structures. These FIR filters mimic the performance of the cascade of a causal $G(z)$ and the corresponding anti-causal $G(z^{-1})$ IIR filters. Their impulse response is a shifted and truncated version of that of $G(z)G(z^{-1})$. Efficient structures are developed for implementing the resulting FIR filters. These structures are parallel connections of several branches. Each branch generates a response corresponding to a complex conjugate pole pair and its mirror-image pair. The truncated version is obtained by using a feedforward term which provides pole-zero cancellations. The key to the implementation is the use of the principle of switching and resetting between two identical copies of the same IIR filter [38]. This stabilizes the pole-zero cancellation and avoids the quantization noise from growing excessively.

1.2 Author's Contributions

The thesis is based on 10 publications, for which the contribution of the author has been essential; he was the first author in all the publications and all the simulations have been carried out by the author.

The contribution of the author can be summarized as follows:

1. The author formulated the design of an MBD as an optimization problem, and developed a systematic optimization technique for their design [15].

2. The author formulated the design of an 1S2F as an optimization problem, and developed a systematic method for the design of this class of filters [7].

3. The author proposed a method for the design of narrow transition-band linear-phase (NBLP) FIR filters [8]. The method is based on an optimization scheme for approximating a linear-phase IIR filter.

4. The author considered the implementational aspects of the design proposed in [8]. Additionally, methods to minimize the required number of elements to implement the filter structure was put forward [10].

5. The author has demonstrated that through the decomposition of the transfer function, the class of filters introduced in [89] turns out to be a special case of a broader class of linear-phase FIR filters. This decomposibility was shown to imply reduction both in terms of the number of components required to implement the filter structure, and in term of roundoff noise [9].

6. The author developed a new transfer function, corresponding to a new class of NBLP FIR filters [11]. In this work, this class of filters will be referred to as *Cascade Structure I.*

7. The author proved that any linear-phase FIR filter of length nK, with n a positive integer, can be decomposed into a multiple branch structure. He also proposed new constraints to the original optimization problem introduced in [15], through which more efficient filters emerged [16].

8. The author proposed an alternative class of NBLP FIR filters, as a result of a new truncation scheme applied to a zero-phase IIR filter. The approach was proven to

excel that earlier presented in [8], at least for certain specifications [12].

9. The author introduced a technique for realization of the structure proposed in [8] as a cascade of subfilters. The cascaded structure proved to be more flexible, more efficient, and less noisy than the original one [13].

10. The noise of the structures introduced in Chapter 3 has been analyzed in detail, and the noise tolerance of the structures has been established [14].

1.3 Thesis Outline

This thesis is organized as follows. Chapter 2 considers efficient design of 1S2F's. First, alternative decimator designs in the literature are briefly introduced, and their pros and cons are reviewed. Next, a novel optimization algorithm for designing an 1S2F is put forward, and a theorem, providing the theoretical background for the algorithm is established. Next, MBD's are introduced, and the ideas of 1S2F and MBD are combined to yield an efficient decimator design. The design is improved by imposing further optimization constraints.

Chapter 3 concentrates on an efficient design of NBLP FIR filters. The principle of switching and resetting, which is of central importance to implementability of the design, is briefly reviewed. Next, the rationale behind the design has been established, the roundoff noise generated by the structure is analyzed, and various decomposition schemes applicable to the design have been derived.

The focus of Chapter 4 is on an alternative design for NBLP FIR filters. The transfer functions of the implementable FIR filters exploiting this approach have been derived, the roundoff noise effects have been investigated, and the decomposibility of the aforementioned transfer functions is established. The efficiency of the proposed designs in Chapters

2–4 is supported by simulation results.

The proofs for most of equations appearing in Chapters 3 and 4 are provided in Appendices A and B.

Chapter 2

Computationally Efficient Decimators

This chapter addresses the problem of 1S2F's from a multiple branch realization perspective. First, the optimum MBD is treated in detail. Next the case of 1S2F (Fig. 1.1(a), with $M = 1$) is addressed, and an optimization technique to solve the design problem is proposed. Finally, the two results converge through a hybrid decimator, i.e., a single-stage decimators, where the first constituent filter is realized as a multiple branch structure. This amounts to a multiple branch structure as given by Fig. 1.2, where an additional filter stage $C(z)$ has been used at the decimator output. Comparisons included in [90] have revealed that the multiplication rate for this design is approximately equal to that of a design with only one branch with a significantly reduced number of delay elements.

2.1 Multiple Branch Decimators

This section introduces a systematic design technique applicable to MBD's. The justification of the design procedure is presented through two fundamental results. Next the multiple branch filter design problem is addressed as an optimization problem, and an algorithm to solve this task is worked out. By adding new constraints to the original optimization

problem, the proposed approach is later refined to lead to more efficient structures.

2.1.1 The Transfer Function

In this section, the generic transfer function of an MBD is introduced, and some properties of this class of transfer functions are derived.

Consider the transfer function of an MBD (Fig. 1.2), which is given by

$$B(z) = \sum_{k=1}^{K} A_k(z) B_k\big(z^D\big). \tag{2.1}$$

In (2.1), $A_k(z)$'s for $k = 1, 2, \ldots, K$ are of the form

$$A_k(z) = \sum_{n=0}^{N_A} a_k[n] z^{-n}, \tag{2.2}$$

where $a_k[N_A - n] = a_k[n]$ for $n = 0, 1, \ldots, N_A$ and $a_k[N_A - n] = -a_k[n]$ for $n = 0, 1, \ldots, N_A$ for k odd and k even, respectively. Correspondingly, $B_k(z)$'s for $k = 1, 2, \ldots, K$ are of the form

$$B_k(z) = \sum_{n=0}^{N_B} b_k[n] z^{-n}, \tag{2.3}$$

where $b_k[N_B - n] = b_k[n]$ for $n = 0, 1, \ldots, N_B$ and $b_k[N_B - n] = -b_k[n]$ for $n = 0, 1, \ldots, N_B$ for k odd and k even, respectively. Moreover, K is the number of branches, that is, $K = 2$ and $K = 3$ refer to two-branch and three-branch decimators, respectively.

It has been shown in [24, 87] that if the length of $B(z)$ is a multiple of D, it is always possible to design $B(z)$ in the form of (2.1) such that all the $A_k(z)$'s are of length D, $K = D$, and the $A_k(z)$'s with k odd are linearly independent mirror-image symmetric polynomials and the $A_k(z)$'s for k even are anti-mirror-image symmetric polynomials in z.

To clarify this idea, consider a linear-phase FIR filter of odd order N_B, in other words,

N_B is assume to be either a type II or a type IV linear-phase FIR filter[1]. The transfer function of this filter, denoted by $B(z)$, can be expressed as

$$B(z) = \sum_{n=0}^{N_B} b[n] z^{-n} \tag{2.4}$$

where for $n = 0, 1, \ldots, N_B$

$$b[n] = \begin{cases} b[N_B - n], & \text{for type II } B(z)\,, \quad (2.5a) \\ -b[N_B - n], & \text{for type IV } B(z). \quad (2.5b) \end{cases}$$

$B(z)$ can now be expressed as [90]

$$B(z) = B_e\big(z^2\big) + z^{-1} B_o\big(z^2\big) \tag{2.6a}$$

where

$$B_e\big(z^2\big) = b[0] + b[2]z^{-2} + b[4]z^{-4} + \cdots + b[N_B - 1]z^{-(N_B-1)} \tag{2.6b}$$

and

$$B_o\big(z^2\big) = b[1] + b[3]z^{-2} + b[5]z^{-4} + \cdots + b[N_B]z^{-(N_B-1)}. \tag{2.6c}$$

Using the above $B_o(z^2)$ and $B_e(z^2)$, the following transfer functions can be defined:

$$B_1\big(z^2\big) = \frac{1}{2}\big[B_e\big(z^2\big) + B_o\big(z^2\big)\big] \tag{2.6d}$$

and

$$B_2\big(z^2\big) = \frac{1}{2}\big[B_e\big(z^2\big) - B_o\big(z^2\big)\big]. \tag{2.6e}$$

From (2.6d) and (2.6e), it is seen that the orders of both $B_1(z)$ and $B_2(z)$ is $(N_B - 1)/2$. The impulse responses of $B_1(z)$ and $B_2(z)$ can be expressed as

[1] In brief, the transfer function of a type I (II) linear-phase FIR filter is a mirror-image polynomial in z^{-1} of even (odd) order, while the transfer function of a type III (IV) linear-phase FIR filter is an antimirror-image polynomial in z^{-1} of even (odd) order. For a more detailed discussion on different types of linear-phase FIR filters, the reader may consult [47, 70, 71].

$$b_1[n] = \frac{b[2n] + b[2n+1]}{2}, \quad n = 0, 1, \ldots, (N_B - 1)/2, \tag{2.7a}$$

and

$$b_2[n] = \frac{b[2n] - b[2n+1]}{2}, \quad n = 0, 1, \ldots, (N_B - 1)/2, \tag{2.7b}$$

respectively.

If $B(z)$ is a type II linear-phase FIR filter, then according to (2.5a) we have:

$$\frac{b[2n] + b[2n+1]}{2} = \frac{b[N_B - 2n] + b[N_B - 2n - 1]}{2} \tag{2.8a}$$

and

$$\frac{b[2n] - b[2n+1]}{2} = \frac{b[N_B - 2n] - b[N_B - 2n - 1]}{2} \tag{2.8b}$$

But according to (2.7a) $\left(\text{with } n \to \frac{N_B - 1}{2} - n\right)$,

$$\frac{b[N_B - 2n] + b[N_B - 2n - 1]}{2} = b_1\left[\frac{N_B - 1}{2} - n\right] \tag{2.9a}$$

and

$$\frac{b[N_B - 2n] - b[N_B - 2n - 1]}{2} = b_2\left[\frac{N_B - 1}{2} - n\right]. \tag{2.9b}$$

Therefore according to (2.7a), (2.8a) and (2.9a) we have:

$$b_1[n] = b_1\left[\frac{N_B - 1}{2} - n\right] \quad n = 0, 1, \ldots, (N_B - 1)/2, \tag{2.10a}$$

and according to (2.7b), (2.8b) and (2.9b) we have:

$$b_2[n] = -b_2\left[\frac{N_B - 1}{2} - n\right] \quad n = 0, 1, \ldots, (N_B - 1)/2. \tag{2.10b}$$

Equation (2.10a) implies that $B_1(z)$ is a linear-phase type I or type II FIR filter, while (2.10b) implies that $B_2(z)$ is a linear-phase type III or type IV FIR filter. Using a similar approach, it can be shown that if $B(z)$ is a type IV linear-phase FIR filter, then $B_1(z)$ will

be a linear-phase type III or type IV and $B_2(z)$ will be a linear-phase type I or type II FIR filter.

Using (2.6a)–(2.6e) enables one to define $B(z)$ in terms of $B_1(z)$ and $B_2(z)$ as follows:

$$B(z) = (1 + z^{-1})B_1\big(z^2\big) + (1 - z^{-1})B_2\big(z^2\big) = B_e\big(z^2\big) + z^{-1}B_o\big(z^2\big). \tag{2.11}$$

Comparing (2.1) and (2.11) gives

$$B(z) = \sum_{k=1}^{2} A_k(z)B_k\big(z^2\big), \tag{2.12}$$

where $A_1(z) = 1 + z^{-1}$ is a mirror-image symmetric polynomial and $A_2(z) = 1 - z^{-1}$ is an anti-mirror-image symmetric polynomial. It can be shown in a similar manner that if the length of $B(z)$ is a multiple of 3, then $B(z)$ can be expressed as

$$B(z) = \sum_{k=1}^{3} A_k(z)B_k\big(z^3\big) \tag{2.13}$$

where $A_1(z) = 1+z^{-1}+z^{-2}$ and $A_3(z) = 1+z^{-2}$ are mirror-image symmetric polynomials and $A_2(z) = 1 - z^{-2}$ is an anti-mirror-image symmetric polynomial.

The credibility of (2.12) and (2.13) has been proven formally in [16]. Proposition 1, to be presented shortly, establishes a more general decomposition scheme for a linear-phase FIR filter. In what follows, the term "symmetric" refers to a type I or type II, and the term "antisymmetric" refers to a type III or type IV linear-phase FIR filter. Moreover, $\lceil x \rceil$ is the smallest integer greater than or equal to x and the length of a filter signifies the length of its impulse response.

Proposition 1. *If $H(z)$ is a symmetric (antisymmetric) linear-phase FIR filter of a length n divisible by D, then it can always be decomposed according to* (1.3a)*. For $k = 1, 2, \ldots, D$, $i = 0, 1, \ldots, \lfloor D/2 \rfloor$, and $j = 0, 1 \ldots \lfloor D/2 - 1 \rfloor$, $A_k(z) = a_0^{(k)} + a_1^{(k)}z^{-1} + \ldots +$*

$a_{D-1}^{(k)} z^{-(D-1)}$, *with*

$$a_i^{(k)} = \begin{cases} 1, & 0 \le i \le \lfloor (D-k)/2 \rfloor, \\ 0, & \textit{otherwise,} \end{cases} \tag{2.14}$$

and

$$a_{D-1-j}^{(k)} = \begin{cases} a_j^{(k)}, & k \textit{ is odd,} \\ -a_j^{(k)}, & k \textit{ is even.} \end{cases} \tag{2.15}$$

For k odd (even), $B_k(z) = b_0^{(k)} + b_1^{(k)} z^{-1} + \ldots + b_{n/D-1}^{(k)} z^{-(D-1)}$ is a symmetric (anti-symmetric) linear-phase FIR, .

Proof. Defining $m = \left\lceil \lceil n/2 \rceil / D \right\rceil$, $\mathbf{x} = [b_0^{(1)}, b_0^{(2)}, \ldots, b_0^{(D)}, b_1^{(1)}, b_1^{(2)}, \ldots, b_1^{(D)}, \ldots$ $b_{m-1}^{(1)}, b_{m-1}^{(2)}, \ldots, b_{m-1}^{(D)}]^T$, and $\mathbf{h} = [h_0, h_1, \ldots, h_{Dm-1}]^T$, we require

$$\mathbf{A}\mathbf{x} = \mathbf{h}, \tag{2.16a}$$

where

$$\mathbf{A} = \begin{pmatrix} \mathbf{H} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{H} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{H} & \ldots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{H} \end{pmatrix} \tag{2.16b}$$

with $\mathbf{A} \in \{-1, 0, 1\}^{(m+1)\times(m+1)}$,

$$\mathbf{H} = \begin{pmatrix} a_0^{(1)} & a_0^{(2)} & \ldots & a_0^{(D)} \\ a_1^{(1)} & a_1^{(2)} & \ldots & a_1^{(D)} \\ \vdots & \vdots & \ddots & \vdots \\ a_{D-1}^{(1)} & a_{D-1}^{(2)} & \ldots & a_{D-1}^{(D)} \end{pmatrix}, \tag{2.16c}$$

and $\mathbf{0}$ is a D by D matrix, with its all entries being 0.

Proving the existence of an $\mathbf{x}$ satisfying (2.16a) is tantamount to proving the decomposibility of $H(z)$ according to (1.3). But as seen from (2.16c), matrix $\mathbf{A}$ is nonsingular and hence the solution $\mathbf{x} = \mathbf{A}^{-1}\mathbf{h}$ satisfies (2.16a).

□

2.1.2 The Optimization Problem

This section states the optimization problem for the proposed class of decimators.

By exploiting the coefficient symmetries of $A_k(z)$'s and $B_k(z)$'s, the overall number of multipliers for the two-branch $(K = 2)$ case becomes

$$R_M^{(2)} = (N_A + 1) + (N_B + 1). \tag{2.17}$$

Similarly, the number of multipliers for the three-branch case $(K = 3)$ is

$$R_M^{(3)} = (N_A + 1) + (N_B + 1) + \left\lfloor \frac{N_A + 1}{2} \right\rfloor + \left\lfloor \frac{N_B + 1}{2} \right\rfloor. \tag{2.18}$$

From (2.1), the zero-phase response (the phase term $e^{-j\omega(N_A+DN_B)/2}$ is omitted from the frequency response) of MBD's can be expressed as

$$H(\omega) = \sum_{k=1}^{K} (-1)^{k-1} A_k(\omega) B_k(D\omega), \tag{2.19}$$

where for k odd

$$B_k(\omega) = \begin{cases} b_k\left[\dfrac{N_B}{2}\right] + 2\displaystyle\sum_{n=1}^{N_B/2} b_k\big[N_B - n\big]\cos(n\omega), & \text{for } N_B \text{ even,} \\ 2\displaystyle\sum_{n=0}^{\frac{N_B-1}{2}} b_k\left[\dfrac{N_B - 1}{2} - n\right]\cos\left(\dfrac{2n\omega + 1}{2}\right), & \text{for } N_B \text{ odd.} \end{cases} \tag{2.20}$$

$A_k(\omega)$ can be expressed in the same form by replacing N_B by N_A and the $b_k[n]$'s by the

$a_k[n]$'s. For k even

$$B_k(\omega) = \begin{cases} 2\sum_{n=0}^{N_B/2} b_k[N_B - n]\sin[(n+1)\omega], & \text{for } N_B \text{ even,} \\ 2\sum_{n=0}^{\frac{N_B-1}{2}} b_k\left[\frac{N_B-1}{2} - n\right]\cos\left(\frac{2n\omega+1}{2}\right), & \text{for } N_B \text{ odd,} \end{cases} \tag{2.21}$$

and $A_k(\omega)$ can be expressed in the same form by replacing N_B by N_A and the $b_k[n]$'s by the $a_k[n]$'s. It is worth mentioning that the term $(-1)^{k-1}$ in (2.19) is due to the fact the phase term for k odd (k even) are $e^{jN_A/2}$ and $e^{jDN_B/2}$ ($je^{jN_A/2}$ and $je^{jDN_B/2}$). Therefore, when expressing the overall response in terms of the zero-phase response and the phase terms $e^{j(DN_B+N_A)}$, the multiplier $(-1)^{k-1} = j^2$ should be included for k even.

The following criteria is stated for $H(\omega)$, as given by (2.19):

$$1 - \delta_p \le H(\omega) \le 1 + \delta_p \text{ for } \omega \in [0,\ \alpha\pi/D], \tag{2.22a}$$

$$-\delta_s \le H(\omega) \le \delta_s \text{ for } \omega \in [\pi/D,\ \pi], \tag{2.22b}$$

where D is the decimation ratio, δ_p and δ_s are the maximum allowable ripples in the pass- and stopband respectively, and $\alpha < 1$ specifies the passband edge to be $\omega_p = \alpha\pi/D$. Alternatively, these criteria can be expressed as

$$|E(\omega)| \le \delta_p \text{ for } \omega \in [0,\ \alpha\pi/D] \cup \omega \in [\pi/D,\ \pi], \tag{2.23a}$$

where

$$E(\omega) = W(\omega)\big[D(\omega) - H(\omega)\big], \tag{2.23b}$$

$$D(\omega) = \begin{cases} 1, & \text{for } \omega \in [0,\ \alpha\pi/D], \\ 0, & \text{for } \omega \in [\pi/D, \pi], \end{cases} \tag{2.23c}$$

and

$$W(\omega) = \begin{cases} 1, & \text{for } \omega \in [0,\ \alpha\pi/D], \\ \delta_p/\delta_s, & \text{for } \omega \in [\pi/D, \pi]. \end{cases} \tag{2.23d}$$

The optimization problem under consideration is the following:

Optimization Problem: Given D, α, δ_p, δ_s, and the number of branches K (either two or three), find the orders and coefficients of $A_k(z)$'s and $B_k(z)$'s, as given by (2.2) and (2.3), to meet the criteria given by (2.22) such that first $R_M^{(2)}$ or $R_M^{(3)}$, as given by (2.17) or (2.18), is minimized, and second,

$$\epsilon = \max_{\omega \in [0,\ \alpha\pi/D] \cup [\pi/D,\ \pi]} |E(\omega)| \tag{2.24}$$

is minimized. This thesis concentrates on the $K = 2$ and $K = 3$ cases, since these selections have turned out to give the best solutions in terms of the required number of multiplications per input sample.

2.1.3 The Optimization Algorithm

This section describes efficient algorithms for solving the optimization problem stated in Section 2.1.2. Denoting the two-branch and three-branch cases by $K = 2$ and $K = 3$, respectively, the overall optimization algorithm can be efficiently carried out as follows:

- Step 1: Design a minimum-order direct-form linear-phase FIR filter $F(z)$ to meet the criteria given by (2.22a) and (2.22b), with $D = K$. Let this order be $N_{\min}$.

- Step 2: Determine the minimum value of the integer L satisfying $LK \geq N_{\min} + 1$.

- Step 3: Redesign a direct-form linear-phase FIR filter transfer function $F(z)$ of order $LK - 1$ to minimize the peak absolute value of $E(\omega)$ as given by (2.23) with $D = K$.

- Step 4: For $K = 2$ [$K = 3$], find $B_1(z^2)$ and $B_2(z^2)$ [$B_1(z^2)$ and $B_2(z^2)$, and $B_3(z^3)$] to satisfy $F(z) = (1 + z^{-1})B_1(z^2) + (1 - z^{-1})B_2(z^2)$ [$F(z) = (1 + z^{-1} + z^{-2})B_1(z^3) + (1 - z^{-2})B_2(z^3) + (1 + z^{-2})B_3(z^3)$].

- Step 5: Find the minimum order N_A for the transfer functions $A_1(z)$ and $A_2(z)$ [$A_1(z)$, $A_2(z)$ and $A_3(z)$] for $K = 2$ ($K = 3$) together with the corresponding $B_k(z)$'s using the following two-step procedure in such a manner that the overall transfer function resulting after Step 5(b) meets the given criteria:

  - Step 5(a): Use linear programming to determine the coefficients of the $A_k(z)$'s order of N_A by keeping the $B_k(z)$ fixed such that

    $$\tilde{\delta}_s = \max_{\omega \in [\pi/D,\ \pi]} |H(\omega)| \tag{2.25}$$

    is minimized subject to the condition that $H(\omega) = 1$ at $\omega = 0$. Here, $H(\omega)$ is given by (2.19).

  - Step 5(b): Optimize the $A_k(z)$ and $B_k(z)$'s simultaneously to minimize ϵ as given by (2.24) using Sequential Quadratic Programming (SQP).

The above algorithm has been implemented as a MATLAB program. There are two reasons for performing Steps 1, 2, 3, and 4 in the above manner. First, it is simple to find initial values for the $B_k(z)$'s with the aid of a set of linear equations. Second, these initial values are very close to the optimum ones also for a high value of D.

2.1.4 Performance Study

Example 1: Consider the filter specifications: $D = 10$, $\alpha = 0.5$, $\delta_p = 0.01$ and $\delta_s = 0.001$. This means that the passband and the stopband edges are located at $\omega_p = 0.5\pi/D$ and $\omega_s = \pi/D$ respectively. The minimum length of an optimum single stage direct-form filter to meet the criteria is 109. There exist two alternative decimator structures for this transfer function, namely, the direct-form structure exploiting the coefficient symmetry and the fact that only every tenth output sample has to be evaluated [31], and the commutative

polyphase structure where the ten branch filters share the same delays [39]. Note that for the polyphase implementation, only one branch filter has a symmetrical impulse response.

As mentioned above, the minimum length of a direct-form linear-phase FIR filter to meet the specifications is 109. To make the length divisible by 10, it is increased to 110. If the decomposition is performed into 10 branches according to the discussion of Section 2.1.1, then the lengths of $A_k(z)$'s become 10 and those of the $B_k(z)$'s become 11.

This decomposition would require 105 multipliers, 10.5 multiplications per input sample, and 19 delay elements when the coefficient symmetries are exploited, and both the $A_k(z)$'s and the $B_k(z)$'s share the same delay elements. By exploiting the coefficient symmetry, the direct-form FIR filters of length 109 can be implemented using 55 multipliers, 5.5 multiplications per input sample, and 108 delay elements. If the direct-form FIR filter is implemented using the commutative polyphase structure, then it requires 104 multipliers, 10.4 multiplications per input sample, and 10 delay elements.

Based on the above data, it is not beneficial to use $K = 10$ branches. However, the $K = 2$ branch and the $K = 3$ branch cases are computationally very efficient. For $K = 2$, the proposed algorithm results in the design with the lengths of $B_1(z)$ and $B_2(z)$ being equal to 11 and the lengths of $A_1(z)$ and $A_2(z)$ being equal to 18. Hence, when reducing the number of branches from 10 to 2, the lengths of the $B_k(z)$'s remain the same, whereas those of $A_k(z)$'s increase from 10 to 18. For the $K = 3$ case, the lengths of the $B_k(z)$'s remain 11, whereas the lengths of the $A_k(z)$'s become 13. The two-branch (three-branch) design requires 29 (37) multipliers, 2.9 (3.7) multiplications per input sample, and 27 (22) delay elements. Figures 2.1 and 2.2 illustrate the amplitude responses for the two-branch and the three-branch cases.

Further savings in the number of multipliers of the optimal (minimum-order) filter can

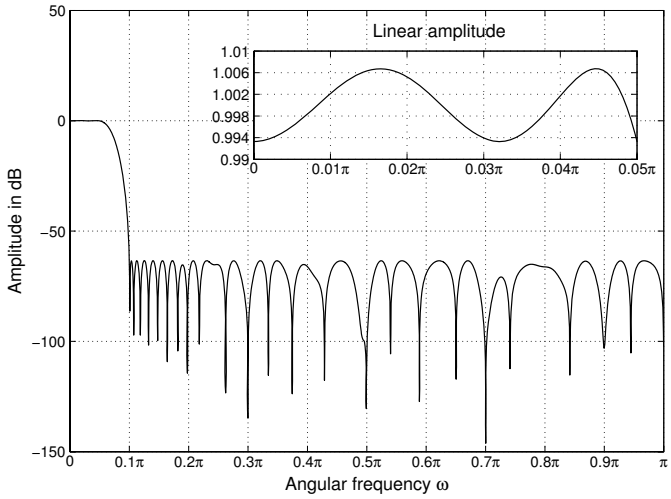

Figure 2.1: *The amplitude response of the overall two branch structure.*

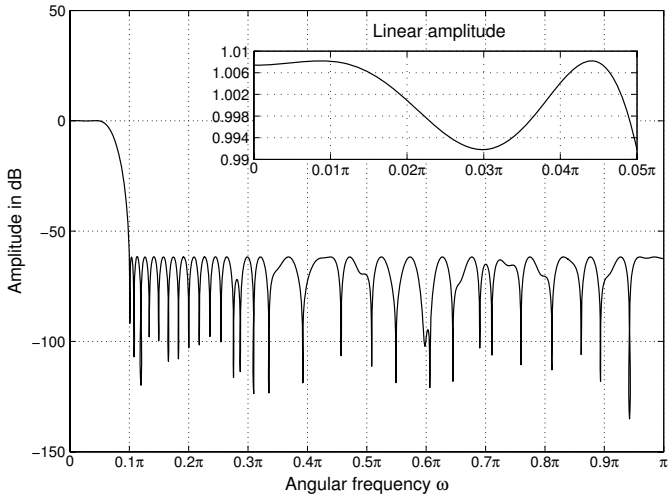

Figure 2.2: *The amplitude response of the overall three branch structure.*

be made if the impulse responses of its subfilters are slightly modified. To illustrate the idea, consider the impulse responses of the non-periodic filter $A_2(z)$ meeting the specifications of the two-branch structure in Section 2.1.4, presented in Figure 2.3.

It is observed that the values of some of the coefficients are close to zero. It is therefore justified to check the existence of other minimum-order linear-phase FIR filters fulfilling

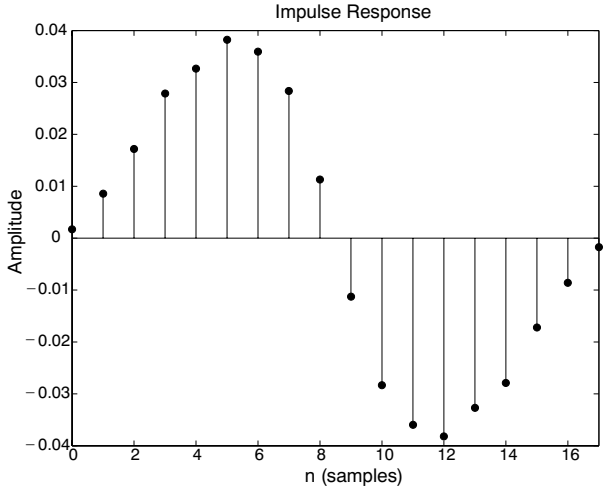

Figure 2.3: *The impulse response of the $A_2(z)$ meeting the specifications of section 2.1.4.*

the specification, with some of their coefficients fixed to zero. If such filter or filters exist, the number of multipliers needed for the realization will be less than the original least-order filter by the number of the coefficients set to zero.

It proves that for the two-branch example considered in Section 2.1.4, such a filter does exist, and at most two coefficients can be set to zero. Figure 2.4 presents the impulse response of the subfilter $A_2(z)$ with its first and last coefficients set to zero, while Fig. 2.5 presents the amplitude response of the filter structure. The new structure requires one multiplier and two adders less than the original design, that is, the number of multipliers and the number of multipliers per input sample are now 28 and 2.8, respectively.

The same idea can be applied to the case in which the decimator filter is realized using three branches. Now a total number of nine coefficients $\big(a_1[0], a_1[6], a_1[12], a_3[1], a_3[2], a_3[3], a_3[9], a_3[10], a_3[11]$, as defined in (2.2)$\big)$ can be set to zero simultaneously, yet the least-order FIR filter still meets the specifications. Compared with the original design, the number of multipliers decreases from 37 to 31, and that of multiplications per input sample decreases from 3.7 to 3.1. The amplitude response of this filter is presented in Fig. 2.6.

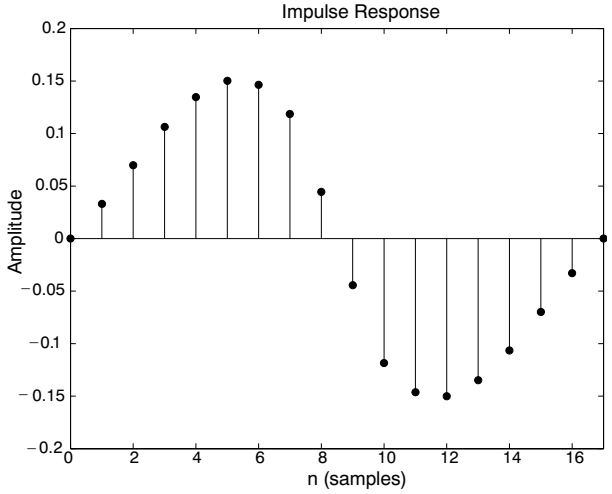

Figure 2.4: *The impulse response of the $A_2(z)$ meeting the specifications of section 2.1.4, with two coefficients forced to zero.*

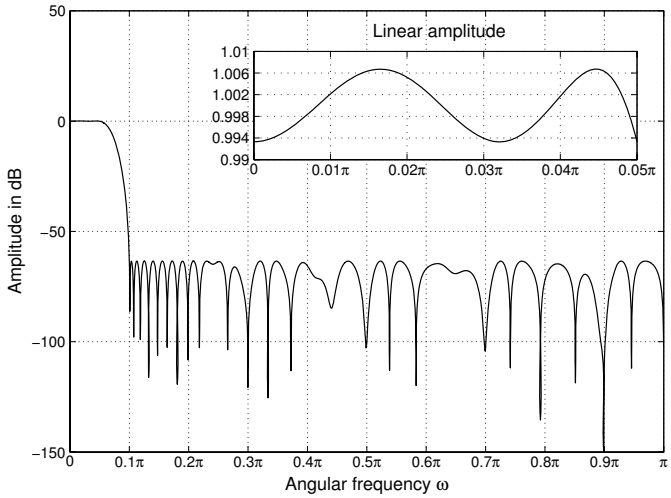

Figure 2.5: *The amplitude response of the two-branch filter structure meeting the specifications of section 2.1.4, with two coefficients forced to zero.*

2.2 Single-Stage Two-Filter Decimators

It has been observed by several authors [24,27,61,86,87] that the computational complexity of a decimator (interpolator) can be drastically reduced by using an additional filter stage at the output (input) sampling rate. Figures 2.7 and 2.8 show the resulting structures and their

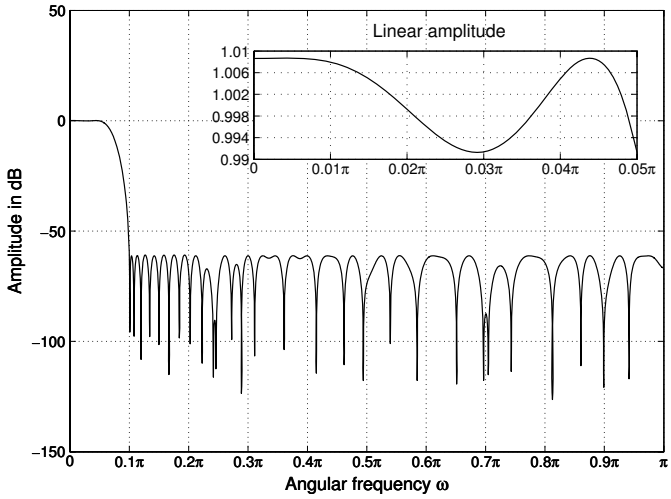

Figure 2.6: *The amplitude response of the three-branch filter structure meeting the specifications of section 2.1.4, with ten coefficients forced to zero.*

single-stage equivalents used for the analysis and synthesis purposes. This observation has been first made by Martinez and Parks in [65]. In their design scheme for decimators, $A(z)$ is a transfer function of a linear-phase FIR filter and $B(z)$ is an all-pole filter. The role of $A(z)$ is to shape the stopband in the desired manner, whereas $B(z)$ gives the desired response for the overall passband. This results in a significant reduction in the overall number of multiplications per input sample at the expense of a nonlinear-phase performance in the passband. In order to achieve a linear-phase performance, Saramäki modified this approach by using a linear-phase FIR transfer function for $B(z)$ [86]. The resulting filters require a slightly higher number of multipliers per input sample. In [86], $B(z)$ has been designed to provide one zero at $z = -1$ in order to reduce the multiplication rate even further in the case where the stopband edge of the decimator is located at $\omega = \pi/D$.

In the above-mentioned two approaches, the role of $A(z)$ is mainly to take care of the stopband shaping, whereas the role of $B(z)$ is to generate the desired passband response.

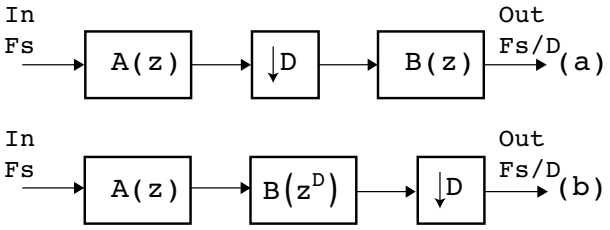

Figure 2.7: single-stage two-filter structure for a D-to-1 decimator. (a) Actual implementation. (b) Single-stage equivalent.

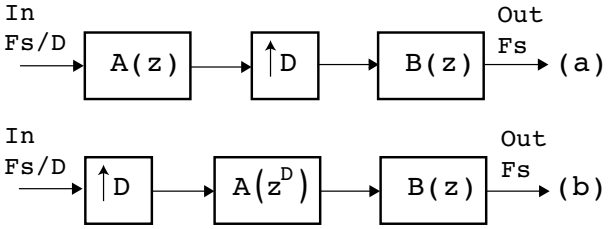

Figure 2.8: single-stage two-filter structure for a 1-to-D interpolator. (a) Actual implementation. (b) Single-stage equivalent.

In [24] and [27], Chu and Burrus have proposed a different strategy in designing linear-phase FIR decimators. In their design scheme, the goal is to meet the given overall criteria such that $A(z)$ has the minimum complexity. As has been observed by Saramäki in [87], the best results in terms of the multiplication rate are obtained between the above extreme cases.

This section presents a systematic approach for designing the decimator structures of Figure 2.7 in such a manner that the overall number of multipliers is minimized. We concentrate on the case where both $A(z)$ and $B(z)$ are linear-phase FIR filters. In [87], the optimum solution has been found by trying various frequency-response-shaping responsibilities between $A(z)$ and $B(z)$ and then minimizing their orders for each selection. Since there are a huge number of alternatives, this approach is very time consuming. Furthermore, it is very difficult to generate a systematic design scheme for automatically finding

the optimum solution based on this approach.

2.2.1 The Transfer Function and the Zero-Phase Response

The transfer function of the proposed linear-phase FIR decimators is of the form

$$H(z) = A(z)B(z^D), \tag{2.26}$$

where

$$A(z) = \sum_{n=0}^{N_A} a[n]z^{-n} \tag{2.27}$$

with $a[N_A - n] = a[n]$ for $n = 0, 1, \ldots, N_A$ and

$$B(z) = \sum_{n=0}^{N_B} b[n]z^{-n} \tag{2.28}$$

with $b[N_B - n] = b[n]$ for $n = 0, 1, \ldots, N_B$. Here, D is the sampling rate conversion ratio. When this filter is used for decimation, $B(z^D)$ is realized as $B(z)$ at the lower output sampling rate as shown in Figure 2.7(a). This reduces the number of delay elements required in the implementation significantly. The zero-phase frequency response for the above transfer function is expressible as

$$H(\omega) = A(\omega)B(D\omega), \tag{2.29}$$

where

$$A(\omega) = \begin{cases} b\left[\dfrac{N_A}{2}\right] + 2\displaystyle\sum_{n=1}^{N_A/2} a[N_A - n]\cos(n\omega), & \text{for } N_A \text{ even,} \\ 2\displaystyle\sum_{n=0}^{\frac{N_A-1}{2}} a\left[\dfrac{N_A-1}{2} - n\right]\cos(\dfrac{2n\omega+1}{2}), & \text{for } N_A \text{ odd,} \end{cases} \tag{2.30}$$

and

$$B(\omega) = \begin{cases} b\left[\frac{N_B}{2}\right] + 2\sum_{n=1}^{N_B/2} b[N_B - n]\cos(n\omega), & \text{for } N_B \text{ even,} \\ 2\sum_{n=0}^{\frac{N_B-1}{2}} b\left[\frac{N_B-1}{2} - n\right]\cos(\frac{2n\omega+1}{2}), & \text{for } N_B \text{ odd.} \end{cases} \quad (2.31)$$

2.2.2 The Optimization Problem

This section states the optimization problem for the proposed decimators.

When exploiting the coefficient symmetries of $A(z)$ and $B(z)$, the overall number of multipliers becomes

$$R_M = \lfloor (N_A + 2)/2 \rfloor + \lfloor (N_B + 2)/2 \rfloor. \quad (2.32)$$

By defining the criteria for $H(\omega)$ as those stated by (2.22) and (2.23), the optimization problem under consideration will be the following:

Optimization Problem: Given D, α, δ_p, and δ_s, find the orders and coefficients of $A(z)$ and $B(z)$, as given by (2.27) and (2.28), to meet the criteria given by (2.22) and (2.23a) such that first R_M, as given by (2.32), is minimized, and second, ϵ, as given by (2.24) is minimized.

2.2.3 The Optimization Algorithm

This section describes the proposed algorithm for finding the optimum solution to the problem stated in Section 2.2.2.

Sub-algorithm Used in the Main Algorithm

Before describing the overall algorithm, a sub-algorithm is introduced. Given the decimator criteria as well as N_A and N_B, the orders of $A(z)$ and $B(z)$, this algorithm is carried out

using the following three steps:

- Step 1: Use the Remez algorithm or linear programming to determine the coefficients of $B(z)$ of order N_B to minimize

$$\tilde{\delta}_p = \max_{\omega \in [0, \alpha\pi]} |B(\omega) - 1| \tag{2.33}$$

  subject to the condition that $B(\omega) = 0$ at $\omega = \pi$.

- Step 2: Use the Remez algorithm or linear programming to determine the coefficients of $A(z)$ of order N_A to minimize

$$\tilde{\delta}_s = \max_{\omega \in [\pi/D, \pi]} |A(\omega)B(D\omega)| \tag{2.34}$$

  subject to the condition that $A(\omega) = 1$ at $\omega = 0$.

- Step 3: Use SQP to simultaneously determine the coefficients of $A(z)$ and $B(z)$ to minimize ϵ as given by (2.24).

2.2.4 The Main Algorithm

Based on the use of the above sub-algorithm, the overall procedure is performed using the following steps:

- Step 1: Use the Remez algorithm to find the minimum-order linear-phase FIR filter to meet the criteria of (2.22). Let this order be N_{min}. Determine an initial guess for N_B as $N_B^{(1)} = \lceil N_{min}/D \rceil$.

- Step 2: Use the above sub-algorithm with a small number of grid points for designing various overall transfer functions $H(z)$, as given by the 1, for the fixed $N_B = N_B^{(1)}$. First use $N_A = D$ and increment it by D until the overall stopband ripple becomes

less than or equal to $2\delta_s$. Save this order as $N_A^{(2)}$. Another alternative would be estimating N_A from the specifications, which is discussed in Section 2.2.5 below.

- Step 3: Use the sub-algorithm with a small number of grid points in such a way that $N_A = N_A^{(2)}$ is fixed and $N_B = N_B^{(1)}$ or is incremented by 1 until the passband ripple becomes less than or equal to $1.2\delta_p$. Save this order as $N_B^{(3)}$.

- Step 4: Use the sub-algorithm with a small number of grid points in such a way that $N_B = N_B^{(3)}$ is fixed and $N_A = N_A^{(2)}$ or is incremented by 1 until the stopband ripple becomes less than or equal to $1.15\delta_s$. Save this order as $N_A^{(4)}$.

- Step 5: Use the sub-algorithm with a small number of grid points in such a way that $N_A = N_A^{(4)}$ is fixed and $N_B = N_B^{(3)}$ or is incremented by 1 until the passband ripple becomes less than or equal to $1.1\delta_p$. Save this order as $N_B^{(5)}$.

- Step 6: Form candidate order pairs (N_A, N_B) for the values of N_B in the range $N_B^{(5)} \leq N_B \leq N_B^{(5)} + J$ with J being an integer. To do this, the sub-algorithm with a small number of grid points is used for each N_B in the above range in order to find the minimum value of N_A to meet the given criteria. A good initial guess for N_A is $N_A = N_A^{(4)}$. Save the resulting pairs (N_A, N_B).

- Step 7: Apply the sub-algorithm with a high number of grid points in order to check whether the given criteria are met by each candidate pair (N_A, N_B). If this is not true for some order pairs, then increment N_A until the criteria are met. Save the resulting pairs (N_A, N_B).

- Step 8: Select among the order pairs the one minimizing R_M as given by (2.32). If there are more than one pair giving the same minimum value, then select the pair

having the smallest value of N_B.

In the above algorithm, fewer grid points are used at Steps 2, 3, 4, 5, and 6 in order to make them faster. At these steps there is no need to use a large number of grid points since we are looking for the potential candidates and not for the ultimate solution. The choice of the number of grid points depends strongly on the specifications, particularly on the width of the transition band. Our simulations suggest that in the few-grid-point stages of the algorithm, the proper number of grid points in the passband and stopband are

$$M_p = \left\lfloor \frac{\pi}{\omega_s - \omega_p} \right\rfloor \tag{2.35}$$

and

$$M_s = \left\lfloor \frac{\pi(\pi - \omega_s)}{\omega_p(\omega_s - \omega_p)} \right\rfloor \tag{2.36}$$

respectively. For the stages with a high number of grid points, M_p and M_s are multiplied by four. The optimization algorithm has been illustrated by the flowchart of Fig. 2.9.

2.2.5 Order Estimation

This section addresses the problem of order estimation for the filter $A(z)$.

In the algorithm introduced above, Step 1 determines the order of the filter $B(z)$, denoted by N_B, fairly precisely. The order of $A(z)$, denoted by N_A, is however found based on a progressive search at Step 2. A good estimation for N_A, specially in designing FIR filters having narrow transition band, can speed up the execution time of the algorithm drastically.

In what follows, two different methods for order estimation of the filter $A(z)$ are introduced. Later on in Section 2.2.6, the results with and without order estimation for filter $A(z)$ have been compared.

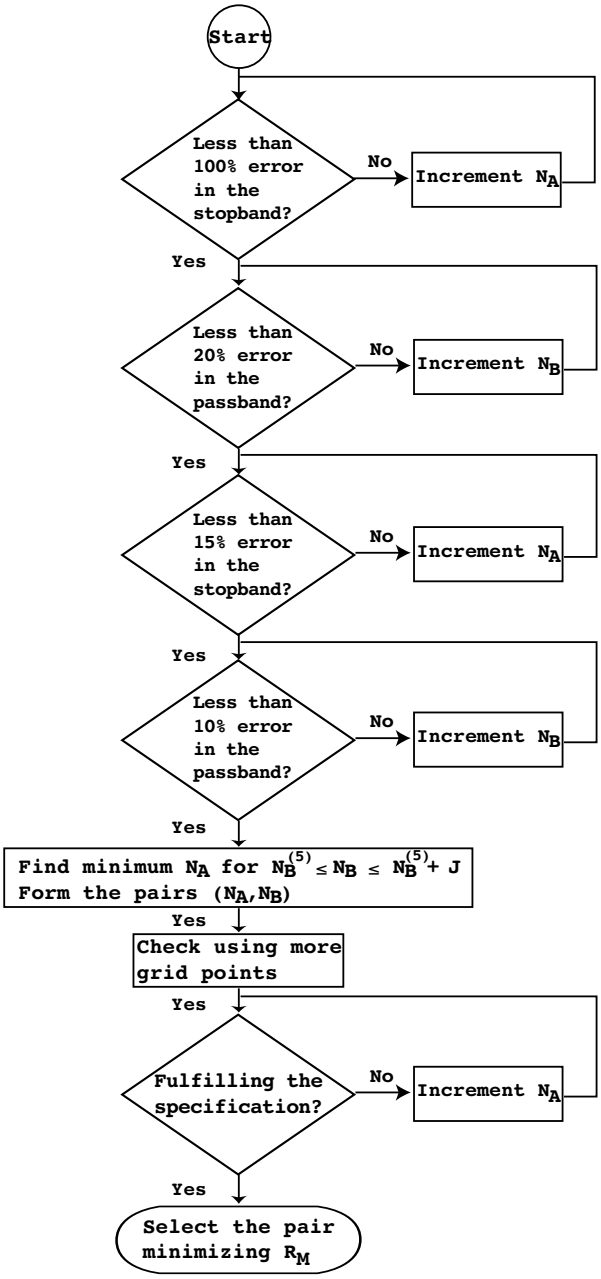

Figure 2.9: The flowchart for the proposed algorithm of Section 2.2.4.

Order estimation using Remez

As mentioned in Section 2.2, $A(z)$ is responsible for the stopband attenuation. This means that to estimate N_A, attention should be paid both to the transition band specification of the

filter, and to its stopband attenuation.

Since $A(z)$ is responsible for the stopband attenuation in the cascade structure $A(z)B\left(z^D\right)$, it is safe to assume that its stopband edge is located approximately at ω_s, where ω_s is the stopband edge of the filter. On the other hand, since the order of $A(z)$ is inversely proportional to its transition band, it follows that the narrower the passband of $A(z)$, the smaller its order. Therefore the minimum order $A(z)$ has a passband of length 0 and a transition band approximately equal to the stopband edge of $A(\omega)B(D\omega)$.

Remez algorithm can be used to estimate the order of $A(z)$. It is impossible to define a zero-width passband in Remez, but since for the fixed transition band, the passband width doesn't have any influence on the estimated order, an arbitrary passband can be assigned. Since in the ideal case the passband is infinitely narrow, the ripple assigned should be as small as possible. The assigned ripple should not however exceed the smallest ripple of the specifications.

If the transition band of the filter to be designed is not too narrow, the stopband edge of $A(z)$ can stretch into the transition band of $B\left(z^D\right)$, which means that the overlap of the transition bands of $A(z)$ and $B\left(z^D\right)$ can still meet the stopband specifications. In these cases, the transition band of $A(z)$ can be thought to be wider then the stopband of the filter, and hence its estimated order becomes smaller. The overlap of transition bands of $A(\omega)$ and $B(D\omega)$ proves to be insignificant in narrow-transition-band filter designs, therefore this approach is not accurate for these extreme cases.

Order estimation using fixed zeros at $z = -1$

It has been experimentally observed that for the optimum design of 1S2F's, $B(z)$ has either two or three zeros on the unit circle. Based on this fact, a lower estimate for N_B and an

upper estimate for N_A can be found rather accurately as follows:

First, $B(z)$, as given by (2.28), is factorized as

$$B(z) = E(z)F(z),$$

where

$$E(z) = (1 - z^{-1})^L$$

and

$$F(z) = \sum_{n=0}^{N_B - L} f[n] z^{-n}.$$

Here, L is either two or three, $N_B - L$ is even and $f[n] = f[N_B - L - n]$ for $n = 1, 2, \ldots, N_B - L$. In this case, the overall zero-phase response becomes

$$H(\omega) = F(D\omega)E(D\omega)A(\omega),$$

where

$$F(\omega) = b\left[\frac{(N_B - L)}{2}\right] + 2 \sum_{n=1}^{(N_B - L)/2} b[N_B - n] \cos(n\omega),$$

$$E(\omega) = [(2\cos(\omega/2)]^L,$$

and $A(\omega)$ is given by (2.30).

Second, the overall criteria are met by designing simultaneously $F(z)$ to meet

$$1 - \delta_p \leq F(\omega)E(\omega)A(\omega/D) \leq 1 + \delta_p \quad \text{for} \quad \omega \in [0, D\omega_p]$$

and $A(z)$ to meet

$$A(0) = 1$$

and

$$-\delta_s \leq F(D\omega)E(D\omega)A(\omega) \leq \delta_s \quad \text{for} \quad \omega \in [\pi/D, \pi].$$

Third, the minimum values of N_A and $N_B - L$ for meeting the given criteria can be found conveniently using the following iterative algorithm:

1. Set $F(\omega) = 1$.

2. Determine $A(z)$ using the Remez algorithm such that the maximum absolute value of the following error function

$$E_A(\omega) = W_A(\omega)[A(\omega) - D_A(\omega)],$$

where

$$D_A(\omega) = \begin{cases} 1 & \text{for } \omega \in [0, \epsilon] \\ 0 & \text{for } \omega \in [\pi/D, \pi] \end{cases}$$

and

$$W_A(\omega) = \begin{cases} \alpha & \text{for } \omega \in [0, \epsilon] \\ E(\omega)F(D\omega) & \text{for } \omega \in [\pi/D, \pi] \end{cases}$$

becomes less than or equal to δ_s with the minimum value of N_A. By selecting ϵ to be a very small positive number and α to be a very large positive number, the Remez algorithm uses only one grid point ($\omega = 0$) on $[0, \epsilon]$ and forces $A(\omega)$ to take the value of unity at $\omega = 0$.

3. Determine $F(z)$ using the Remez algorithm such the the maximum absolute value of the following error function

$$E_F(\omega) = W_F(\omega)[F(\omega) - D_F(\omega)],$$

where

$$D_F(\omega) = 1/[E(\omega)A(\omega/D)], \omega \in [0, D\omega_p]$$

and

$$W_F(\omega) = E(\omega)A(\omega/L), \omega \in [0, L\omega_p]$$

becomes less than or equal to δ_p with the minimum even value of $N_B - L$.

4. Repeat Steps 2 and 3 until the difference between successive solutions is within the given tolerance limits.

Typically, four to six iterations of the above algorithm are required to arrive at the desired solution. The above algorithm is practically the same as that described in [86], with the exception that now the number of fixed zeros at $z = -1$ for $B(z)$ is two or three. As the algorithm described in [86], the above algorithm determines for the given value of L, the number of zeros at $z = -1$, the minimum order of $B(z)$ ($N_B - L + L = N_B$) for achieving the desired passband response for the overall filter as well as the minimum order for $A(z)$ for providing the desired stopband response.

The key idea of using the above algorithm is the fact that when forcing the L zeros (L is two or three) of the optimum design lying on the unit circle to be located at $z = -1$, the resulting order N_B (N_A) is less than or equal to (greater than or equal to) that of the optimized overall design. The desired lower (upper) estimate for N_B (N_A) can be found by first carrying out the above algorithm for both $L = 2$ and $L = 3$ and then selecting those values of N_B and N_A, for which the sum $N_A + N_B$ is lower.

2.2.6 Performance Study

This section illustrates, by means of two examples, the efficiency and flexibility of the proposed algorithm.

Example 2: In this example, a step-by-step execution of the above-mentioned algorithm for the specifications of *Example 1* on page 20 is considered. The PC performing the simulation was a Pentium 1.6 GHz with 512 Mb of memory. For simplicity, for all the steps of the sub-algorithm of Subsection 2.2.3 the function `fminimax` from the optimization toolbox provided by MathWorks Inc. has been used [28].

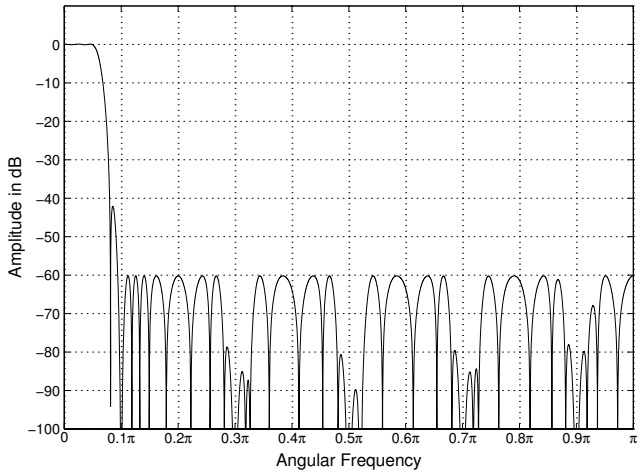

Figure 2.10: Amplitude response of the optimal 1S2F of *Example 2*.

First the algorithm described in the previous section starts with the order of $B(z)$ being equal to 11 (end of Step 1) and that of $A(z)$ being equal to 10. After 9 seconds, the orders of $A(z)$ and $B(z)$ become 40 and 11, respectively (end of Step 2). At this stage, the stopband ripple is less than or equal to $2\delta_s$. At the end of Step 3, which takes only 3 seconds, the orders of $A(z)$ and $B(z)$ become 40 and 12, respectively. Now the passband ripple is less than or equal to $1.2\delta_p$.

It turns out that the same orders of $A(z)$ and $B(z)$ make the passband ripple less than $1.15\delta_s$. Therefore, Step 4 is skipped. After another 2 seconds, Step 5 is finished, resulting in $A(z)$ and $B(z)$ of orders 40 and 12, respectively. Step 6 takes 38 seconds, and at the end of this step, we have the following pairs of candidates: (40,12), (38,13), and (39,14). The first member of each pair represents the order of $A(z)$ and the second one represents the order of $B(z)$. In our simulation, we chose $J = 2$.

At the end of Step 7, which takes 31 seconds, we have the candidate pairs (40,12), (38,13), and (39,14). The last stage is the selection of the result, which corresponds to Steps 8. The pair (38,13) is the solution resulting in the smallest R_M, and is, therefore, the

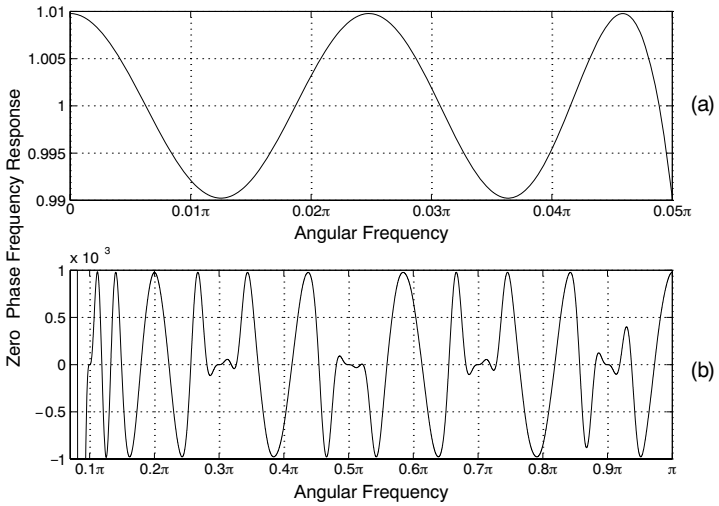

Figure 2.11: Zero-phase frequency response of the optimal 1S2F of *Example 2*. (a) Passband details. (b) Stopband details.

optimal solution.

If the estimation method introduced in Subsection 2.2.5 is utilized, the order of $A(z)$ will be found to be equal to 43, and Step 2 will take 7 seconds. The filter pair $A(z)$ and $B(z)$ of order 43 and 11 respectively fulfill the criteria of Steps 3 to 5. Step 6 takes 38 seconds, and at the end of this step, we have the following pairs of candidates: (42,11), (40,12), and (38,13), where the first member of each pair represents the order of $A(z)$ and the second one represents the order of $B(z)$. J is chosen to be 2, as in the previous case. The same pairs pass through Step 7, which takes 31 seconds. The pair (38,13) is obviously the optimal solution. Various responses of the optimum overall filter with $N_A = 38$ and $N_B = 13$ are depicted in Figs. 2.10 and 2.11.

The results for this example suggests that although order estimation introduced in Subsection 2.2.5 speeds up the optimization run time, the saving in terms of the elapsed time is not substantial.

The optimized filter requires 27 multipliers, 51 delay elements, and 2.7 multiplications

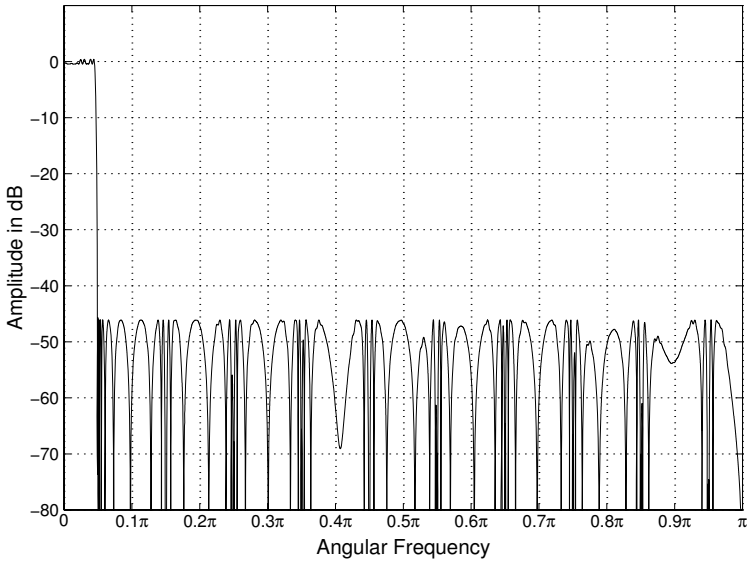

Figure 2.12: Amplitude response of the optimal 1S2F of *Example 3*.

per input sample, implying that the proposed 1S2F provides an excellent performance when compared to its direct-form counterpart, for which the corresponding figures, as already indicated in *Example 1*, are 55, 108, and 5.5 respectively.

Example 3: If we apply the same procedure for the following specifications [24, 61, 65, 86, 87]: $D = 20$, $\delta_p = 0.05$, $\delta_s = 0.005$, $\omega_p = 0.045\pi$, and $\omega_s = 0.05\pi$, then the algorithm of Section 2.2.3 shows that the optimum solution is achieved by $N_A = 105$ and $N_B = 40$. The corresponding decimator requires 74 multipliers, 145 delay elements, and 3.7 multiplications per input sample. The corresponding figures for the direct-form linear-phase FIR filter of order 652 exploiting the coefficient symmetry and the fact that only every twentieth output sample is evaluated are 327, 652, and 16.35. Once again the superiority of the proposed design is clearly evident. Figures 2.12 and 2.13 show some responses for this optimized design.

Order estimation proves to be very significant for this case. The first order estimation

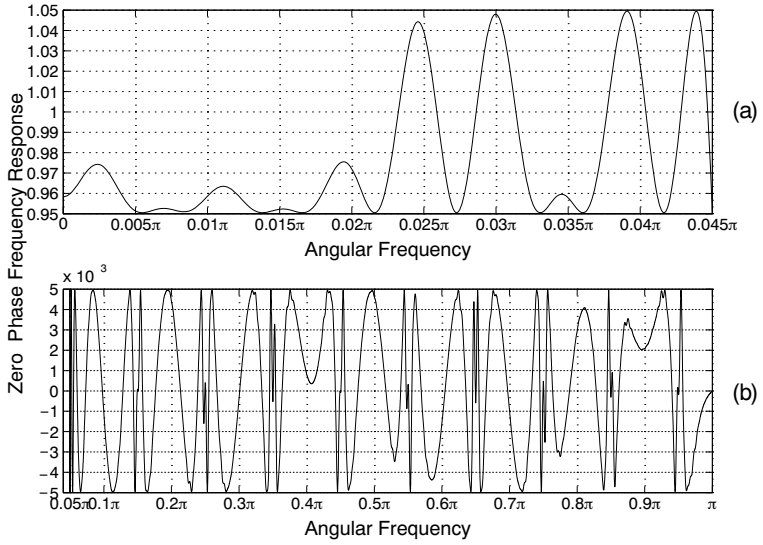

Figure 2.13: Zero-phase frequency response of the optimal 1S2F of *Example 3*. (a) Passband details. (b) Stopband details.

scheme suggested in 2.2.5 shortens the execution time of Step 2 of the algorithm by a factor of 8, and the few-grid-point phase of the algorithm (Steps 1–6) is cut down by 26%. The effect of the second order estimation scheme is even more pronounced; the execution time of Step 2 of the algorithm is reduced by a factor of 27 and that of the few-grid-point phase of the algorithm (Steps 1–6) is cut down by 73%.

2.3 Hybrid Decimators

An interesting class of decimators can be developed by merging the ideas presented in Sections 2.1 and 2.2. These structures enjoy the optimality of both the aforementioned decimators, and can be readily designed by some modifications to the algorithms presented in Sections 2.1.2 and 2.2.2.

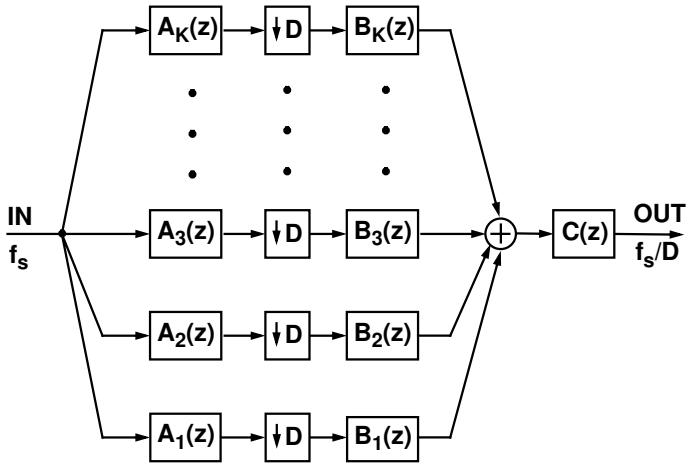

Figure 2.14: *Hybrid FIR filter structure for a D-to-1 decimator.*

2.3.1 The Transfer Function

This section introduces the proposed class of hybrid decimators. To arrive at the linear-phase overall decimator with a symmetrical impulse response, the transfer function of this class of decimators is initially considered. Then, the zero-phase frequency response for this transfer function is expressed for the optimization purposes. Finally the optimization problem is addressed.

Transfer Function of the Hybrid Decimators

The transfer function of the proposed linear-phase FIR hybrid decimators is of the form

$$H(z) = B(z)C\big(z^D\big), \tag{2.37}$$

where $B(z)$ is given by (2.1). When this filter is used for decimation, $C(z^D)$ as well as the $B_k(z^D)$'s can be implemented as $C(z)$ and $B_k(z)$, respectively, at the lower output sampling rate as shown in Fig. 2.14. This reduces the number of delay elements required in the implementation significantly.

Restrictions on the Transfer Function

As has been pointed out in Section 2.1.1, any symmetric transfer function $B(z)$ is expressible in the form of (2.1) with $K \rightarrow D$ provided that

1. The length of $B(z)$ is a multiple of D, that is, the order is $N_B = nD-1$ with n being an integer.

2. The orders of the $A_k(z)$'s are $D-1$ (the lengths are D).

3. The $A_k(z)$'s for k odd (even) are linearly independent mirror-image (anti-mirror-image) symmetrical polynomials and the $A_k(z)$'s for k even are linearly independent anti-mirror-image symmetrical polynomials.

In this decomposition, the orders of the $B_k(z)$'s are $D-1$ (lengths are D). The $B_k(z)$'s for k odd (even) are mirror-image (anti-mirror-image) symmetrical polynomials.

As demonstrated in Section 2.1.1, the above decomposition is not very beneficial, since in most cases, $K = 2$ or $K = 3$ minimizes the arithmetic complexity of the overall decimator at the expense of a slightly increased overall filter order. The increase is mainly due to the fact that after decreasing the number of branches from D to K, the transfer functions $A_k(z)$'s require a higher order to meet the given criteria. The restrictions imposed on $B_k(z)$'s and $A_k(z)$'s are presented in (2.2) and (2.3) respectively, while $C(z)$ is restricted to be of the form

$$C(z) = \sum_{n=0}^{N_C} c[n] z^{-n}, \tag{2.38}$$

where $c[N_C - n] = c[n]$ for $n = 0, 1, \ldots, N_C$.

The zero-phase frequency response for the above transfer function is expressible as (ignoring the delay term $e^{-j\omega(D(N_C+N_B)+N_A)/2}$)

$$H(\omega) = C(D\omega)\sum_{k=1}^{K}(-1)^{k-1}B_k(D\omega)A_k(\omega). \tag{2.39}$$

Here,

$$C(\omega) = \begin{cases} c\left[\frac{N_C}{2}\right] + 2\sum_{n=1}^{N_C/2} c\big[N_C - n\big]\cos(n\omega), & \text{for } N_C \text{ even,} \\ 2\sum_{n=0}^{\frac{N_C-1}{2}} c\left[\frac{N_C-1}{2} - n\right]\cos\left(\frac{2n\omega+1}{2}\right), & \text{for } N_C \text{ odd,} \end{cases} \tag{2.40}$$

and $A(\omega)$ and $B_k(\omega)$ are given by (2.20) and (2.21) respectively.

2.3.2 The Optimization Problem and the Optimization Algorithm

The optimization problem is exactly that of Section 2.1.2. The only modification is that now $R_M^{(2)}$ and $R_M^{(3)}$ are defined to be

$$R_M^{(2)} = (N_A+1) + (N_B+1) + \left\lfloor \frac{N_C+1}{2} \right\rfloor. \tag{2.41}$$

and

$$R_M^{(3)} = (N_A+1) + (N_B+1) + \left\lfloor \frac{N_A+1}{2} \right\rfloor + \left\lfloor \frac{N_B+1}{2} \right\rfloor + \left\lfloor \frac{N_C+1}{2} \right\rfloor \tag{2.42}$$

respectively, and moreover, the order and coefficients of $C(z)$ should also be obtained.

To solve the aforementioned optimization problem, the optimization algorithm already presented in Section 2.1.3 can be utilized. The main difference is that now at Steps 1, 2 and 3, instead of $B(z)$, the transfer function $F(z)C(z^K)$ is used such that

1. The impulse responses of both $C(z)$ and $F(z)$ are symmetric.

2. The order of $F(z)$ is selected such that the desired factorization at Step 4 can be performed.

3. $\lfloor (N_C + 2)/2 \rfloor + \lfloor (N_F + 2)/2 \rfloor$, where N_C and N_F are the orders of $C(z)$ and $F(z)$, respectively, is minimized.

Filters minimizing the quantity given by Condition 3 and fulfilling Conditions 1 and 2 can readily be obtained using the algorithm described in Section 2.2.3. At step 5(a), a fixed $C(z)$ is used, whereas at Step 5(b) it is optimized together with the $A_k(z)$'s and the $B_k(z)$'s.

2.3.3 Multiplierless Designs

Multiplierless realization of the composite filters leads to further savings in the number of the required components. An instance is a computationally efficient implementation of the proposed multibranch design, exploiting multiplierless $A_k(z)$'s of the form

$$A_k(z) = \Big(\frac{1 - z^{-D}}{1 - z^{-1}}\Big)^L \sum_{n=0}^{D-1} \alpha_k(n) z^{-n} \tag{2.43a}$$

with

$$\alpha_k(n) = \big(n - (D-1)/2\big)^{(k-1)}. \tag{2.43b}$$

The design of the desired two-branch structure starts by determining the optimized $B(z)C(z^2)\big(1 + z^{-1}\big)^L$ for $D = 2$, according to the algorithm described in Section 2.2.3. The resulting $B(z)$ is subsequently decomposed according to Proposition 1 (with $D = 2$) to yield $B_1(z)$ and $B_2(z)$, and the coefficients of the $B_k(z)$'s and $C(z)$ thus obtained are used as the starting point in the final optimization routine for designing the desired filter with the given value of D.

The design of the three-branch structure is carried out correspondingly. The only difference is that the initial transfer function to be optimally designed through the algorithm of Section 2.2.3 is $B(z)C(z^3)\left(1+z^{-1}+z^{-2}\right)^L$, and the resulting $B(z)$ will be decomposed according to Proposition 1 with $D = 3$.

2.3.4 Performance Study

Example 4: Consider the specifications of *Example 1* of Section 2.1.4. The proposed algorithm for the two-branch hybrid case results in an optimum filter structure, where the lengths of $A_1(z)$ and $A_2(z)$ are 18, the lengths of $B_1(z)$ and $B_2(z)$ are 5 and the length of $C(z)$ is 8. This decimator requires 27 multipliers, 2.7 multiplications per input sample, and 29 delay elements. Hence, compared to the corresponding MBD case (where $C(z)$ is absent), the number of multipliers reduces from 29 to 27 at the expense of a higher number of delay elements(29 compared to 27).

Figures 2.15, and 2.16 present the amplitude responses for $K = 2$ and $K = 3$ hybrid structures respectively. It proves that for the hybrid two-branch structure, two coefficients can be set to zero. For the hybrid three-branch case, there is a maximum of six coefficients to be set to zero. Figure 2.17 (2.18) presents the amplitude response for such two (three) branch structure.

The specifications of this example can also be met through the multiplierless designs, introduced in Section 2.3.3. With $L = 4$ in (2.43a), a two-branch structure with $B_1(z)$ and $B_2(z)$ of length 4 and $C(z)$ of length 8 meets the criteria. The structure requires 8 multipliers and 26 delay elements, and operates at 0.8 multiplications per input sample. The amplitude response of this design is presented in Fig. 2.19. Table 2.1 compares different decimator structures fulfilling the specifications of this example.

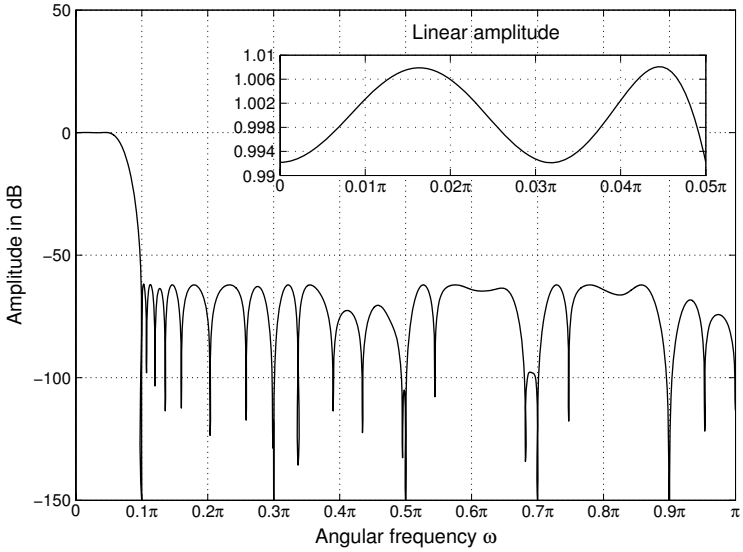

Figure 2.15: The amplitude response of the whole hybrid two branch structure in *Example 4*.

Table 2.1: Comparison Between Various Decimator Structures for the Specifications of *Example 1*. For More Details, See the Text.

Decimator Type	Filter Orders	Number of Multipliers	Multiplication Rate	Number of Delays
Conventional FIR, Direct-Form	$N = 108$	55	5.5	108
MBD, $K = 2$	$N_A = 17, N_B = 10$	29	2.9	27
Hybrid, $K = 2$	$N_A = 17, N_B = 4, N_C = 7$	27	2.7	28
MBD, $K = 3$	$N_A = 12, N_B = 10$	39	3.9	22
Hybrid, $K = 3$	$N_A = 12, N_B = 4, N_C = 7$	30	3.0	23
MBD, coefficients zeroed, $K = 2$	$N_A = 17, N_B = 10$	28	2.8	27
Hybrid, coefficients zeroed, $K = 2$	$N_A = 17, N_B = 4, N_C = 7$	26	2.6	28
MBD, coefficients zeroed, $K = 3$	$N_A = 12, N_B = 10$	34	3.4	22
Hybrid, coefficients zeroed, $K = 3$	$N_A = 12, N_B = 4, N_C = 7$	27	2.7	23
Hybrid, Multiplierless, $K = 2$	$N_B = 3, N_C = 7$	8	0.8	26
Hybrid, Multiplierless, $K = 3$	$N_B = 3, N_C = 7$	10	1.0	26
1S2F	$N_A = 38, N_B = 13$	27	2.7	51

Example 5: Consider the specifications of *Example 3* of Section 2.2.6. The design criteria are met using a multiplierless two-branch structure with $L = 4$, $B_1(z)$ and $B_2(z)$

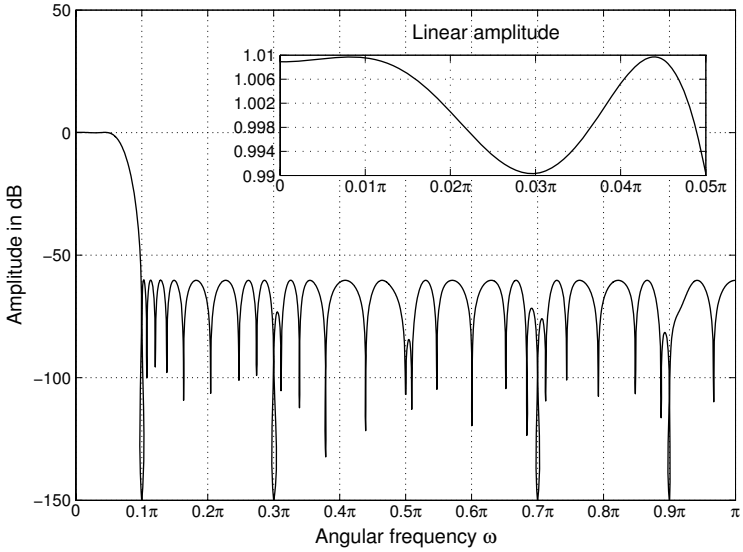

Figure 2.16: The amplitude response of the whole hybrid three branch structure in *Example 4*.

of length 6 and $C(z)$ of length 34, requiring 23 multipliers, 1.15 multiplications per input sample and 54 delay elements, implying a huge advantage of multiplierless hybrid design compared to the implementation according to 1S2F design of Section 2.2.6. The amplitude response of the optimum design is presented in Fig. 2.20.

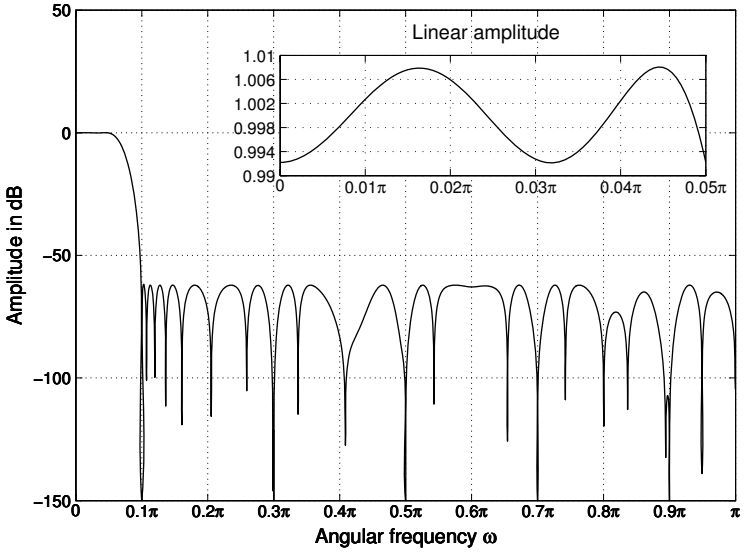

Figure 2.17: *The frequency response of the hybrid two-branch filter structure in Example 4, with two coefficients forced to zero.*

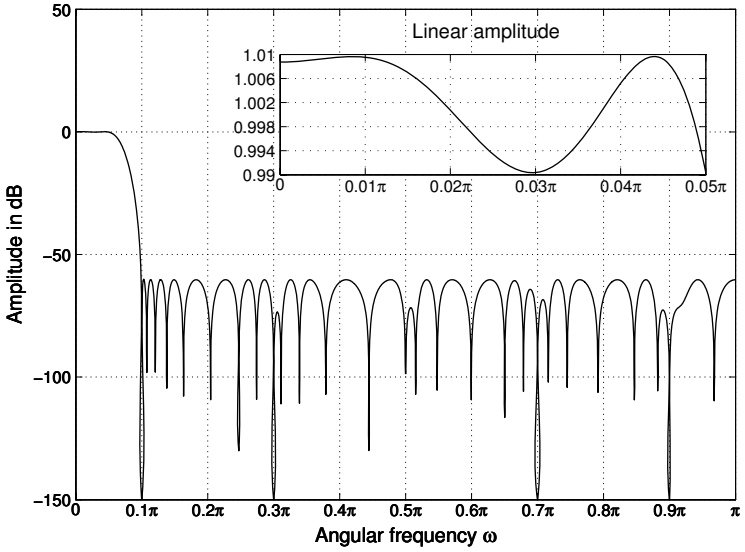

Figure 2.18: *The frequency response of the hybrid three-branch filter structure in Example 4, with six coefficients forced to zero.*

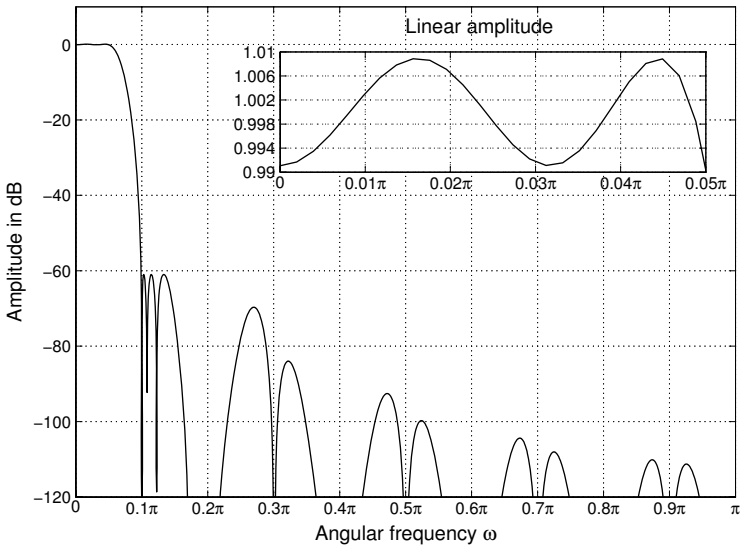

Figure 2.19: Amplitude response of the two-branch filter with fixed $A_k(z)$ in *Example 4*.

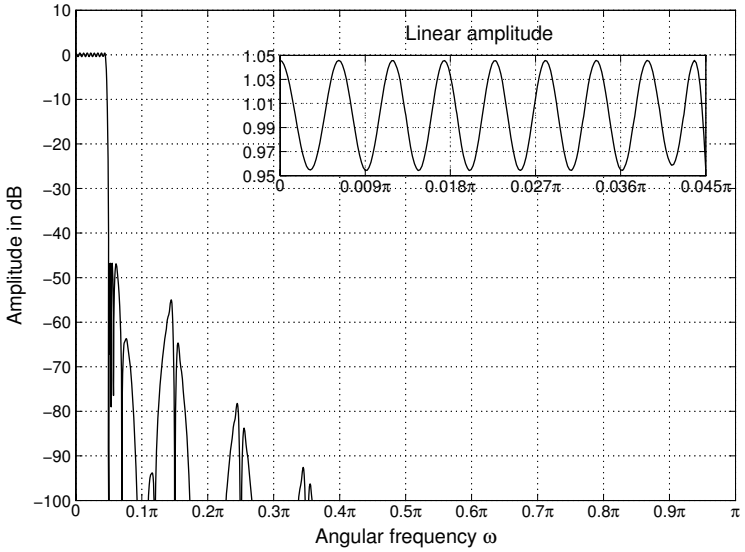

Figure 2.20: Amplitude response of the two-branch filter with fixed $A_k(z)$ in *Example 5*.

Chapter 3

Cascade Structures for Generating Sharp Linear-Phase FIR Filters

In this chapter, efficient structures composed of two cascaded FIR blocks are introduced. Each block in turn is a parallel connection of several FIR filters, which are realized by using IIR filters. Thanks to the incorporation of IIR filters, the structures can be implemented efficiently. When generating these structures, special attention should be paid to combat the roundoff noise.

This chapter starts by a short introduction to the principle of switching and resetting, which is of key importance to the realizability of the design. Next, two transfer functions for the proposed cascade structures are derived, and efficient schemes for their implementations are put forward. Finally, closed-form formulae for the noise generated in the proposed structures are derived, and the efficiency of the resulting designs is illustrated through computer simulations.

3.1 Review of Principle of Switching and Resetting

In [38], an approach for synthesizing linear-phase FIR filters based on switching and resetting of the same IIR filters has been introduced. To explain this principle, consider the following first-order stable IIR transfer function:

$$G_1(z) = \frac{1}{1 - bz^{-1}}. \tag{3.1}$$

Here, b is either real or complex and $|b| < 1$ to make the filter stable. The FIR filter transfer function

$$H_1(z) = \frac{1 - b^N z^{-N}}{1 - bz^{-1}} = \sum_{n=0}^{N-1} b^n z^{-n} \tag{3.2}$$

could be made to have a frequency response that is as close to that of $G_1(z)$ as desired by selecting an appropriately large N.

Similarly, the approximating causal FIR filter transfer function to the corresponding unstable IIR transfer function

$$G_2(z) = \frac{1}{1 - b^{-1}z^{-1}} \tag{3.3}$$

is

$$H_2(z) = \frac{1 - b^{-N} z^{-N}}{1 - b^{-1}z^{-1}} = \sum_{n=0}^{N-1} b^{-n} z^{-n}. \tag{3.4}$$

Cascading the filters with transfer functions as given by (3.2) and (3.4) results in a linear-phase FIR filter that has an even symmetry.

$H_1(z)$ as given by (3.2) represents the transfer function of an FIR filter because the pole at $z = b$ is cancelled by one of the equispaced zeros on the circle of radius $|b|$. If the cancellation due to finite coefficient wordlength is inexact, then the transfer function becomes

$$\widehat{H}_1(z) = \frac{1 - \widehat{b^N} z^{-N}}{1 - \widehat{b} z^{-1}}, \tag{3.5}$$

where $\widehat{b}$ and $\widehat{b^N}$ denote the implemented finite-precision values of b and b^N, respectively. The resulting impulse response of $\widehat{H}_1(z)$ is given by

$$\widehat{h}_1(n) = \begin{cases} \widehat{b}^n, & 0 \le n \le N-1, \\ \big(\widehat{b}^N - \widehat{b^N}\big)(\widehat{b})^{n-N}, & n \ge N. \end{cases} \tag{3.6}$$

By a similar argument, $H_2(z)$ with the transfer function as given by (3.4) represents an FIR filter because the pole at $z = b^{-1}$ is cancelled by one of the equispaced zeros on the circle of radius $|b^{-1}|$. If the cancellation due to finite coefficient wordlength is inexact, then the transfer function becomes

$$\widehat{H}_2(z) = \frac{1 - \widehat{b^{-N}} z^{-N}}{1 - \widehat{b^{-1}} z^{-1}}, \tag{3.7}$$

where $\widehat{b^{-1}}$ and $\widehat{b^{-N}}$ denote the implemented finite-precision values of b^{-1} and b^{-N}, respectively. The impulse response of $\widehat{H}_2(z)$ is given by

$$\widehat{h}_2(n) = \begin{cases} \big(\widehat{b^{-1}}\big)^n, & 0 \le n \le N-1, \\ \big((\widehat{b^{-1}})^N - \widehat{b^{-N}}\big)\big(\widehat{b^{-1}}\big)^{n-N}, & n \ge N. \end{cases} \tag{3.8}$$

Based on (3.6) and (3.8), it is obvious that in these two cases, the pole is not exactly removed and its effect will appear in the output for $n \ge N$. In particular, for the filter $\widehat{H}_2(z)$ given by (3.7), the effect of an inexact pole-zero cancellation $\big(\widehat{b^{-N}} \ne (\widehat{b^{-1}})^N\big)$ is growing as $|\widehat{b^{-1}}|^n$ for $n > N$.

To avoid the above-mentioned problem, a technique based on switching and resetting of IIR filters has been proposed in [38]. To elaborate on the idea, assume that the input to $\widehat{H}_1(z)$ (or equally well $\widehat{H}_2(z)$) consists of a set of $2N$ samples $x(0), x(1), \ldots, x(2N-1)$, of which the last N samples $x(N), x(N+1), \ldots, x(2N-1)$ are zero-valued. If $\widehat{H}_1(z)$ works in the ideal way, i.e., if the finite wordlength effects are ignored, then the output at time $n = 2N-1$ should be precisely equal to zero. This motivates to reset the state variable

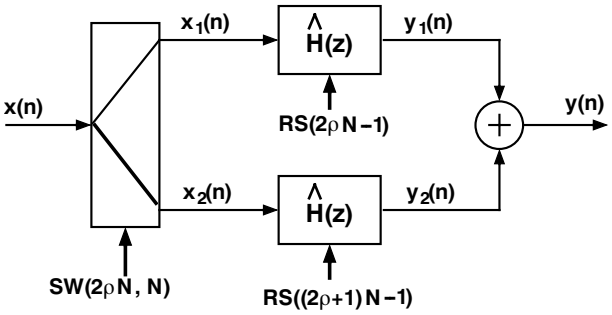

Figure 3.1: Switching and resetting for stabilizing pole-zero cancellation. A demultiplexer is used to decompose $x(n)$ into two signals $x_1(n)$ and $x_2(n)$ with periodic sequences of N zeros. If the demultiplexer is switched every N points, then each copy of the filter is fed N actual data points followed by N zeros. Each filter is reset following the Nth zero and just before receiving the next sequence of N actual data points.

at time $n = 2N - 1$ and hence avoids the effect of an inexact pole-zero cancellation from growing too much. To extend the application of switching and resetting to arbitrary inputs, the input sequence of the filter $x(n)$ is decomposed into two sequences as follows:

$$x(n) = x_1(n) + x_2(n), \tag{3.9a}$$

where

$$x_1(n) = \begin{cases} x(n), & rN \le n \le (r+1)N - 1,\ r \text{ is even}, \\ 0, & \text{otherwise}, \end{cases} \tag{3.9b}$$

$$x_2(n) = \begin{cases} x(n), & rN \le n \le (r+1)N - 1,\ r \text{ is odd}, \\ 0, & \text{otherwise}. \end{cases} \tag{3.9c}$$

The subsequences $x_1(n)$ and $x_2(n)$ are applied to two identical copies of $\widehat{H}_1(z)$ and the outputs are added (see Fig. 3.1). This is equivalent to filtering $x(n)$ by $\widehat{H}_1(z)$ as implied by (3.9b)–(3.9c). Only the finite wordlength effects are different. Because after each set of N actual data samples there is a set of N zero-valued samples, the filters can be reset at the time when the last zero-valued sample enters the filter.

3.2 Alternative Structures

The cascade structures introduced in this chapter are based on a finite approximation of an IIR filter. To elaborate on the idea, consider the following transfer function of a stable and causal IIR filter:

$$G(z) = \frac{\sum_{k=0}^{L} n_k z^{-k}}{1 - \sum_{k=1}^{K} d_k z^{-k}} = g \frac{\prod_{k=1}^{L} (1 - q_k z^{-1})}{\prod_{k=1}^{K} (1 - p_k z^{-1})}. \tag{3.10}$$

Here, the zeros q_k and the poles p_k are assumed to be either real or to occur in complex conjugate pairs to make the coefficients n_k and d_k real. In addition, it is assumed that there are K_C complex conjugate pole pairs β_k and β_k^* for $k = 1, 2, \ldots, K_C$ and K_R real poles α_k for $k = 1, 2, \ldots, K_R$. Using a partial fraction expansion and assuming that there are no repeated poles, the above transfer function can be written as

$$G(z) = A(z) + \sum_{k=1}^{K_R} G_k^{(r)}(z) + \sum_{k=1}^{K_C} G_k^{(c)}(z), \tag{3.11a}$$

where

$$A(z) = \sum_{k=0}^{L-K} a_k z^{-k}, \tag{3.11b}$$

$$G_k^{(r)}(z) = \frac{\kappa_k}{1 - \alpha_k z^{-1}}, \tag{3.11c}$$

and

$$\begin{aligned} G_k^{(c)}(z) &= \frac{\gamma_k}{1 - \beta_k z^{-1}} + \frac{\gamma_k^*}{1 - \beta_k^* z^{-1}} \\ &= 2\Re\Big\{ \frac{\gamma_k}{1 - \beta_k z^{-1}} \Big\}, \end{aligned} \tag{3.11d}$$

with

$$\kappa_k = (1 - \alpha_k z^{-1}) G(z) \Big|_{z=\alpha_k} \tag{3.11e}$$

and

$$\gamma_k = (1 - \beta_k z^{-1}) G(z)\Big|_{z=\beta_k}. \tag{3.11f}$$

Here, $\Re\{H(z)\}$ denotes the transfer function that is the average of $H(z)$ and the transfer function obtained by replacing the complex coefficients of $H(z)$ by their complex conjugates. The first term on the right-hand side of (3.11a) is absent if $L < K$.

Based on different approximations applied to (3.11c) and (3.11d), two alternative structures emerge, which will be referred to as Cascade Structure I and Cascade Structure II.

3.2.1 Cascade Structure I

Under the assumption that

$$\frac{1}{1 - \alpha_k z^{-1}} \approx \frac{1 - \alpha_k^{M_k} z^{-M_k}}{1 - \alpha_k z^{-1}} \tag{3.12a}$$

and

$$\frac{1}{1 - \beta_k z^{-1}} \approx \frac{1 - \beta_k^{N_k} z^{-N_k}}{1 - \beta_k z^{-1}}, \tag{3.12b}$$

$G(z)$ as given by (3.11a)–(3.11f) can be approximated by the following FIR filter transfer function:

$$E(z) = A(z) + \sum_{k=1}^{K_R} E_k^{(r)}(z) + \sum_{k=1}^{K_C} E_k^{(c)}(z), \tag{3.13a}$$

where

$$E_k^{(r)}(z) = \frac{\kappa_k (1 - \alpha_k^{M_k} z^{-M_k})}{1 - \alpha_k z^{-1}} \tag{3.13b}$$

and

$$E_k^{(c)}(z) = 2\Re\Big\{ \frac{\gamma_k (1 - \beta_k^{N_k} z^{-N_k})}{1 - \beta_k z^{-1}} \Big\}. \tag{3.13c}$$

Using the notations

$$\gamma_k = R_k e^{j\Phi_k} \tag{3.14a}$$

and

$$\beta_k = r_k e^{j\phi_k}, \tag{3.14b}$$

$E_k^{(c)}(z)$ can be alternatively expressed as

$$E_k^{(c)}(z) = \frac{2\Re\{\gamma_k(1-\beta_k^{N_k}z^{-N_k})(1-\beta_k^* z^{-1})\}}{(1-\beta_k z^{-1})(1-\beta_k^* z^{-1})} \tag{3.15a}$$

$$= \frac{b_0 + b_1 z^{-1} + b_2 z^{-N_k} + b_3 z^{-N_k-1}}{1 - 2r_k\cos(\phi_k)z^{-1} + r_k^2 z^{-2}}, \tag{3.15b}$$

where

$$b_0 = 2R_k\cos(\Phi_k), \tag{3.16a}$$

$$b_1 = -2R_k r_k\cos(\Phi_k - \phi_k), \tag{3.16b}$$

$$b_2 = -2R_k r_k^{N_k}\cos(\Phi_k + N_k\phi_k), \tag{3.16c}$$

and

$$b_3 = 2R_k r_k^{N_k+1}\cos(\Phi_k + (N_k-1)\phi_k). \tag{3.16d}$$

The impulse response of the resulting $E(z)$ is given by

$$e(n) = e^{(a)}(n) + \sum_{k=1}^{K_R} e_k^{(r)}(n) + \sum_{k=1}^{K_C} e_k^{(c)}(n), \tag{3.17a}$$

where

$$e^{(a)}(n) = \begin{cases} a_n, & 0 \le n \le L-K, \\ 0, & \text{otherwise,} \end{cases} \tag{3.17b}$$

$$e_k^{(r)}(n) = \begin{cases} \kappa_k(\alpha_k)^n, & 0 \le n \le M_k - 1, \\ 0, & \text{otherwise,} \end{cases} \tag{3.17c}$$

and

$$e_k^{(c)}(n) = \begin{cases} 2\text{Re}\{\gamma_k(\beta_k)^n\} = 2R_k(r_k)^n\cos(\Phi_k + n\phi_k), & 0 \le n \le N_k - 1, \\ 0, & \text{otherwise.} \end{cases} \tag{3.17d}$$

The above $E(z)$ is the transfer function of an FIR filter, whose impulse response becomes zero at $n = D + 1$, where

$$D = \max\{M_1 - 1, M_2 - 1, \cdots, M_{K_R} - 1, N_1 - 1, N_2 - 1, \cdots, N_{K_C} - 1\}. \quad (3.18)$$

The FIR filter transfer function $\widehat{E}(z)$, whose impulse response coefficients $\widehat{e}(n)$ are related to the coefficients $e(n)$ of $E(z)$ via

$$\widehat{e}(n) = e(D - n), \quad n = 0, 1, 2, \cdots, N - 1 \quad (3.19)$$

is then

$$\widehat{E}(z) = z^{-D} E(z^{-1}). \quad (3.20)$$

This transfer function is expressible as follows:

$$\widehat{E}(z) = z^{-(D-(L-K))}\widehat{A}(z) + \sum_{k=1}^{K_R} z^{-(D+1-M_k)}\widehat{E}_k^{(r)}(z) + \sum_{k=1}^{K_C} z^{-(D+1-N_k)}\widehat{E}_k^{(c)}(z), \quad (3.21a)$$

where

$$\widehat{A}(z) = \sum_{k=0}^{L-K} a_k z^{-(L-K-k)}, \quad (3.21b)$$

$$\widehat{E}_k^{(r)}(z) = \kappa_k \alpha_k^{M_k - 1} \frac{(1 - (1/\alpha_k)^{M_k} z^{-M_k})}{1 - (1/\alpha_k) z^{-1}} = \frac{\kappa_k \alpha_k^{-1} (\alpha_k^{M_k} - z^{-M_k})}{1 - (1/\alpha_k) z^{-1}}, \quad (3.21c)$$

and

$$\widehat{E}_k^{(c)}(z) = 2\Re\Big\{ \frac{\gamma_k \beta_k^{-1} [(\beta_k)^{N_k} - z^{-N_k}]}{1 - (1/\beta_k) z^{-1}} \Big\}. \quad (3.21d)$$

A linear-phase FIR filter transfer function can then be constructed as follows:

$$F(z) = c_0 z^{-D} + c_1 E(z) \widehat{E}(z). \quad (3.22)$$

Here, the role of the constants c_0 and c_1 is to adjust the zero-phase response of $F(z)$ to stay within the limits of the specifications. Alternatively, using the substitutions $a_k \sqrt{c_1} \rightarrow a_k$

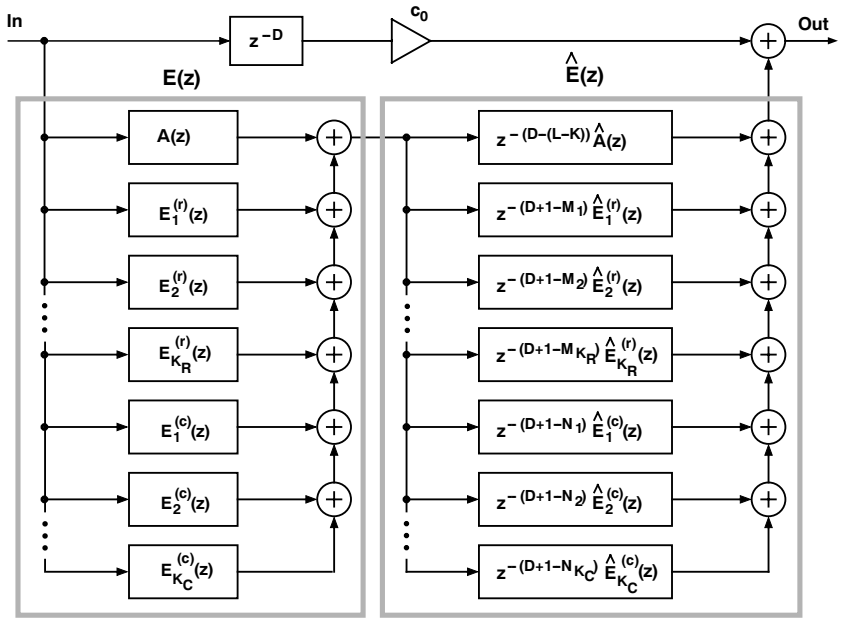

Figure 3.2: Graphic representation of Cascade Structure I.

for $k = 0, 1, \ldots, L-K$, $\kappa_k\sqrt{c_1} \rightarrow \kappa_k$ for $k = 1, 2, \ldots, K_R$, and $\gamma_k\sqrt{c_1} \rightarrow \gamma_k$ for $k = 1, 2, \ldots, K_C$, the effect of c_1 can be included in $E(z)$ and $\widehat{E}(z)$ and $F(z)$ can be expressed as

$$F(z) = c_0 z^{-D} + E(z)\widehat{E}(z). \tag{3.23}$$

The basic structure for implementing the overall filter is shown in Fig. 3.2. More detail on how to practically construct the constituent subfilters will be discussed in Section 3.3.1.

3.2.2 Cascade Structure II

For generating Cascade Structure II, $G_k^{(c)}(z)$ as given by (3.11d) is first expressed as

$$G_k^{(c)}(z) = 2\Re\left\{\frac{\gamma_k(1-\beta_k^*)z^{-1}}{(1-\beta_k z^{-1})(1-\beta_k^* z^{-1})}\right\}$$

Alternatively, this equation can be written in the following forms:

$$G_k^{(c)}(z) = \frac{2\Re\{\gamma_k(1-\beta_k^* z^{-1})\}}{(1-\beta_k z^{-1})(1-\beta_k^* z^{-1})} = \frac{b_0 + b_1 z^{-1}}{1 - 2r_k\cos(\phi_k)z^{-1} + (r_k)^2 z^{-2}}, \tag{3.24}$$

where b_0 and b_1 are given by (3.16a) and (3.16b). Then, applying the approximation of (3.12b) and

$$\frac{1}{1-\beta_k^* z^{-1}} \approx \frac{1-(\beta_k^*)^{N_k} z^{-N_k}}{1-\beta_k^* z^{-1}} \tag{3.25}$$

yields

$$\begin{aligned} E_k^{(c)}(z) =& \frac{2\Re\{\gamma_k(1-\beta_k^* z^{-1})\}\left(1-\beta_k^* z^{-N_k}\right)\left(1-\beta_k z^{-N_k}\right)}{(1-\beta_k z^{-1})(1-\beta_k^* z^{-1})} \\ =& \frac{(b_0 + b_1 z^{-1})\Omega_k(z)}{1 - 2r_k\cos(\phi_k)z^{-1} + (r_k)^2 z^{-2}} = 2\Omega_k(z)\Re\Big\{\frac{\gamma_k}{1-\beta_k z^{-1}}\Big\}, \end{aligned} \tag{3.26}$$

where

$$\Omega_k(z) = 1 - 2(r_k)^{N_k}\cos(N_k\phi_k)z^{-N_k} + (r_k)^{2N_k} z^{-2N_k}. \tag{3.27}$$

The corresponding impulse response is given by

$$e_k^{(c)}(n) = \begin{cases} 2\text{Re}\{\gamma_k(\beta_k)^n\} = 2R_k(r_k)^n\cos(\Phi_k + n\phi_k), & 0 \le n \le N_k - 1, \\ -2\text{Re}\{\gamma_k(\beta_k^*)^{N_k}(\beta_k)^{(n-N_k)}\} = -2R_k(r_k)^n\cos(\Phi_k + (n-2N_k)\phi_k), & \\ & N_k \le n \le 2N_k - 1, \\ 0, & \text{otherwise.} \end{cases} \tag{3.28}$$

In this case, $E(z)$ is given by (3.13a), (3.13b) and (3.26). The corresponding impulse response of the overall transfer function becomes zero at $n = D + 1$, where

$$D = \max\{M_1 - 1, M_2 - 1, \cdots, M_{K_R} - 1, 2N_1 - 1, 2N_2 - 1, \cdots, 2N_{K_C} - 1\}. \tag{3.29}$$

The corresponding

$$\widehat{E}(z) = z^{-D}E(z^{-1})$$

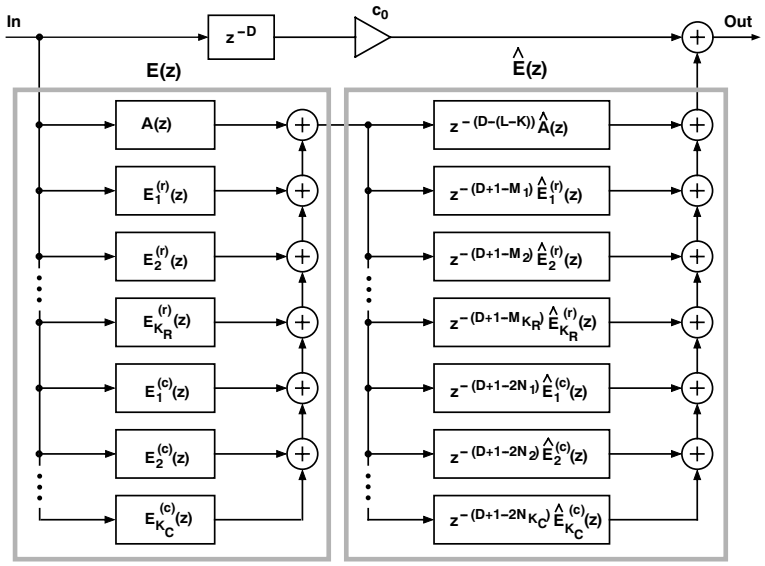

Figure 3.3: Graphic representation of Cascade Structure II.

is then given by

$$\widehat{E}(z) = z^{-(D-(L-K))}\widehat{A}(z) + \sum_{k=1}^{K_R} z^{-(D+1-M_k)}\widehat{E}_k^{(r)}(z) + \sum_{k=1}^{K_C} z^{-(D+1-2N_k)}\widehat{E}_k^{(c)}(z), \quad (3.30a)$$

where $\widehat{A}(z)$ and the $\widehat{E}_k^{(r)}(z)$'s are given by (3.21b) and (3.21c), and

$$\widehat{E}_k^{(c)}(z) = 2\widehat{\Omega}_k(z)\Re\Big\{\frac{\widehat{\gamma}_k}{1-(1/\beta_k)z^{-1}}\Big\} = \frac{(\widehat{b}_0+\widehat{b}_1 z^{-1})\widehat{\Omega}_k(z)}{1-2(1/r_k)\cos(\phi_k)z^{-1}+(1/r_k)^2 z^{-2}}, \quad (3.30b)$$

with

$$\widehat{\Omega}_k(z) = r_k^{(2N_k)}\big(1-2(r_k)^{-N_k}\cos(N_k\phi_k)z^{-N_k}+(r_k)^{-2N_k}z^{-2N_k}\big), \quad (3.30c)$$

$$\widehat{\gamma}_k = -\gamma_k/\beta_k, \quad (3.30d)$$

$$\widehat{b}_0 = -2R_k(r_k)^{-1}\cos(\Phi_k-\phi_k), \quad (3.30e)$$

and

$$\widehat{b}_1 = 2R_k(r_k)^{-2}\cos(\Phi_k). \quad (3.30f)$$

The graphical representation for implementing the overall filter is presented in Fig. 3.3.

3.3 Filter Implementation

This section shows how to practically implement the subfilters for both Cascade Structure I and Cascade Structure II respectively. In the sequel, special emphasis will be laid on finite wordlength effects.

3.3.1 Cascade Structure I

As mentioned earlier in Section 3.2.1, Cascade Structure I is composed of two blocks in cascade, where the first (second) block consists of parallel connection of stable (unstable) IIR filters. In what follows, the implementations of the filter parts corresponding to the stable and unstable IIR filter blocks are considered separately.

Stable Blocks

As mentioned in Section 3.1, applying the principle of switching and resetting is not necessary for implementation of $E_k^{(r)}(z)$ and $E_k^{(c)}(z)$, since for these subfilters the output noise variance does not increase boundlessly. Furthermore, instead of using two copies of the same filter, it is advisable to use more bits for the coefficients in the feedforward part to provide a sufficient pole-zero cancellation. For $E_k^{(c)}(z)$ implemented as shown in Fig. 3.4(b) (see Fig. 3.5 for the definitions of the basic arithmetic operations), the input-output relation can be described by the following difference equations:

$$y(l) = \frac{2}{\nu_k} \operatorname{Re}\Big\{W(l)\Big\}, \tag{3.31a}$$

where

$$W(l) = \beta_k W(l-1) + V(l). \tag{3.31b}$$

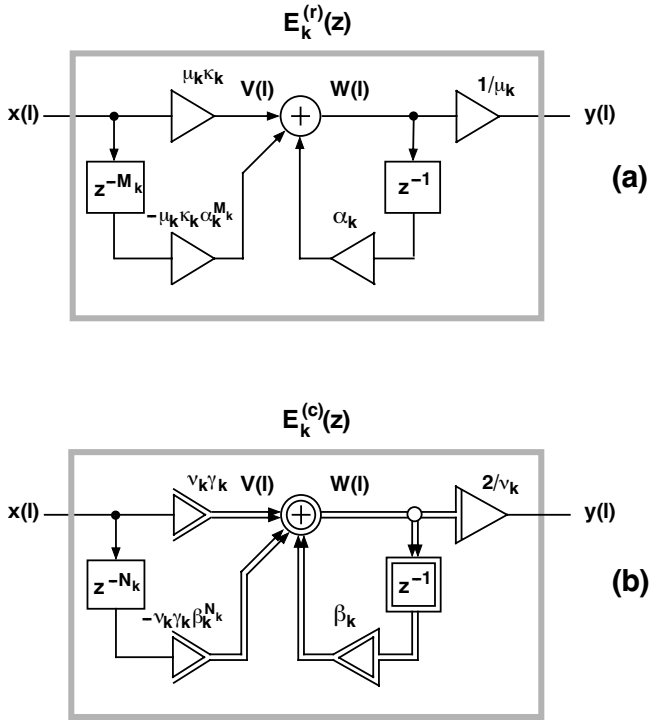

Figure 3.4: Implementations for the subfilters $E_k^{(r)}(z)$ and $E_k^{(c)}(z)$ for Cascade Structure I.

with

$$V(l) = \nu_k \gamma_k x(l) - \nu_k \gamma_k (\beta_k)^{N_k} x(l - N_k), \tag{3.31c}$$

Here, $V(l)$ $(W(l))$ is a complex-valued sequence before (after) the adder of Fig. 3.4(b), whereas ν_k is the scaling constant. Alternatively, the above equations can be written using real arithmetic as shown in Table 3.1, by expressing $V(l)$ and $W(l)$ in terms of their real and imaginary parts, denoted by $V_{\text{re}}(l)$ $V_{\text{im}}(l)$, $W_{\text{re}}(l)$ $W_{\text{im}}(l)$, as $V(l) = V_{\text{re}}(l) + jV_{\text{im}}(l)$ and $W(l) = W_{\text{re}}(l) + jW_{\text{im}}(l)$.

From Table 3.1, it follows that the number of real-valued multiplications, additions, and delays involved in the implementation are 9, 6, and $N_k + 2$, respectively [1] . The role of

[1]Generating $W_{\text{re}}(l)$ and $W_{\text{im}}(l)$ takes two multiplications and two additions, while generating $V_{\text{re}}(l)$ and $V_{\text{im}}(l)$ requires two multiplications and one addition. Finally to yield $y(l)$, one more multiplication is required. Other results presented in the coming pages of this work follow a similar approach.

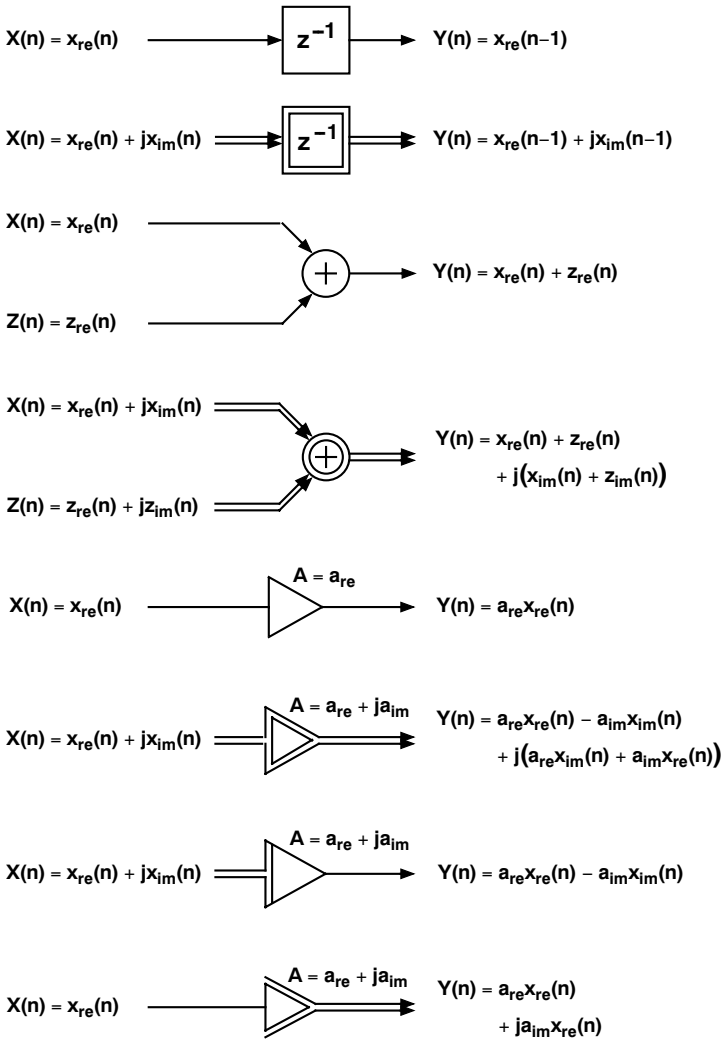

Figure 3.5: Basic arithmetic operations.

the scaling constant ν_k is to avoid overflows in the real and imaginary parts of $W(l)$. In the case of perfect pole-zero cancellation, the unit sample responses from the input to these variables are given by

$$h_k^{(\text{real})}(l) = \begin{cases} \text{Re}\{\nu_k \gamma_k \beta_k^l\}, & 0 \le l \le N_k - 1, \\ 0, & \text{otherwise}, \end{cases} \tag{3.32a}$$

Table 3.1: Real Arithmetic Input-Output Relation for $E_k^{(c)}(z)$

$y(l) = (2/\nu_k)W_{\text{re}}(l)$,
where
$W_{\text{re}}(l) = \text{Re}\{\beta_k\}W_{\text{re}}(l-1) - \text{Im}\{\beta_k\}W_{\text{im}}(l-1) + V_{\text{re}}(l)$
and
$W_{\text{im}}(l) = \text{Im}\{\beta_k\}W_{\text{re}}(l-1) + \text{Re}\{\beta_k\}W_{\text{im}}(l-1) + V_{\text{im}}(l)$
with
$V_{\text{re}}(l) = \text{Re}\{\nu_k\gamma_k\}x(l) - \text{Re}\{\nu_k\gamma_k(\beta_k)^{N_k}\}x(l-N_k)$
and
$V_{\text{im}}(l) = \text{Im}\{\nu_k\gamma_k\}x(l) - \text{Im}\{\nu_k\gamma_k(\beta_k)^{N_k}\}x(l-N_k)$

and

$$h_k^{(\text{imag})}(l) = \begin{cases} \text{Im}\{\nu_k\gamma_k\beta_k^l\}, & 0 \le l \le N_k - 1, \\ 0, & \text{otherwise,} \end{cases} \tag{3.32b}$$

respectively.

$E_k^{(r)}(z)$, in turn, can be implemented using conventional real arithmetic as shown in Fig. 3.4(a). The scaling transfer functions for this filter is given by

$$H_k(z) = \mu_k E_k^{(r)}(z). \tag{3.33}$$

The implementation of $E_k^{(r)}(z)$ requires 4 real-valued multiplications and 2 real-valued additions. $A(z)$, as given by (3.11b), is a conventional FIR filter requiring $L - K + 1$ real-valued multiplications and $L - K$ real-valued additions. These results are summarized in Table 3.2.

Unstable Blocks

Due to the unstable poles of $\widehat{E}_k^{(r)}(z)$'s and $\widehat{E}_k^{(c)}(z)$'s, special care should be taken to control the output noise variance due to the multiplication roundoff errors. By exploiting the

Table 3.2: Arithmetic Operations for Cascade Structure I

	Number of Real-Valued Multiplications	Number of Real-Valued Additions
$E_k(z)$		
$E_k^{(r)}(z)$	4	2
$E_k^{(c)}(z)$	9	4
$A_k(z)$	$L-K+1$	$L-K$
$\widehat{E}_k(z)$		
$\widehat{E}_k^{(r)}(z)$	$m_k^{(b)}+3$	$m_k^{(b)}+1$
$\widehat{E}_k^{(c)}(z)$	$2n_k^{(b)}+7$	$2n_k^{(b)}+2$
$\widehat{A}_k(z)$	$L-K+1$	$L-K$
Overall Filter	$2(L-K+1)+1$ $+7K_R+\sum_{k=1}^{K_R} m_k^{(b)}$ $+16K_C+2\sum_{k=1}^{K_C} n_k^{(b)}$	$2(L-K)+1$ $+4K_R+\sum_{k=1}^{K_R} m_k^{(b)}$ $7K_C+2\sum_{k=1}^{K_C} n_k^{(b)}$
	Number of Delay Elements	
$E_k(z)$ and $c_0 z^{-D}$	$D+1+K_R+2K_C$	
$\widehat{E}_k(z)$	$D+1+2K_R+4K_C$	
Overall Filter	$2D+2+3K_R+6K_C$	

identity

$$1-x^{m_k^{(a)}m_k^{(b)}}=(1-x^{m_k^{(a)}})\sum_{l=0}^{m_k^{(b)}-1}[x^{m_k^{(a)}}]^l, \tag{3.34}$$

$\widehat{E}_k^{(r)}(z)$, as given by (3.21c) can be expressed for

$$M_k=m_k^{(a)}m_k^{(b)} \tag{3.35a}$$

as

$$\widehat{E}_k^{(r)}(z)=\Upsilon_k^{(a)}(z)\Upsilon_k^{(b)}(z), \tag{3.35b}$$

where

$$\Upsilon_k^{(a)}(z) = \frac{\kappa_k \alpha_k^{-1}(\alpha_k^{m_k^{(a)}} - z^{-m_k^{(a)}})}{1-(1/\alpha_k)z^{-1}} \tag{3.35c}$$

and

$$\Upsilon_k^{(b)}(z) = \sum_{l=0}^{m_k^{(b)}-1} [(\alpha_k)^{m_k^{(a)}}]^{(m_k^{(b)}-1-l)} z^{-l m_k^{(a)}}. \tag{3.35d}$$

Similarly, $\widehat{E}_k^{(c)}(z)$ as given by (3.21d) is expressible for

$$N_k = n_k^{(a)} n_k^{(b)} \tag{3.36a}$$

as

$$\widehat{E}_k^{(c)}(z) = 2\Re\Big\{\Gamma_k^{(a)}(z)\Gamma_k^{(b)}(z)\Big\}, \tag{3.36b}$$

where

$$\Gamma_k^{(a)}(z) = \frac{\gamma_k \beta_k^{-1}(\beta_k^{n_k^{(a)}} - z^{-n_k^{(a)}})}{1-(1/\beta_k)z^{-1}} \tag{3.36c}$$

and

$$\Gamma_k^{(b)}(z) = \sum_{l=0}^{n_k^{(b)}-1} [(\beta_k)^{n_k^{(a)}}]^{(n_k^{(b)}-1-l)} z^{-l n_k^{(a)}}. \tag{3.36d}$$

Figure 3.6 shows the resulting implementation for $z^{-(D+1-N_k)}\widehat{E}^{(c)}(z)$, where the scaling constant $\widehat{\nu}_k$ is included and the principle of switching and resetting is employed. Since the order of $\Gamma_k^{(a)}(z)$ is $n_k^{(a)} - 1$, the demultiplexers are switched every $n_k^{(a)}$ samples. The feedforward part is realized only once. This is possible by synchronizing the two demultiplexers such that the first and second demultiplexers start feeding actual data samples to the upper and the lower feedback loops at the time instants $n = 2\rho n_k^{(a)}$ and $n = (2\rho-1)n_k^{(a)}$, respectively, where ρ is an integer. Ignoring the effects of switching and resetting as well as those of the additional delay terms, the input-output relation for $E_k^{(c)}(z)$ can be described by the following difference equations:

$$y(l) = \sum_{i=0}^{n_k^{(b)}-1} \mathrm{Re}\Big\{\frac{2[(\beta_k)^{n_k^{(a)}}]^{(n_k^{(b)}-1-i)}}{\widehat{\nu}_k} W(l - i n_k^{(a)})\Big\}, \tag{3.37a}$$

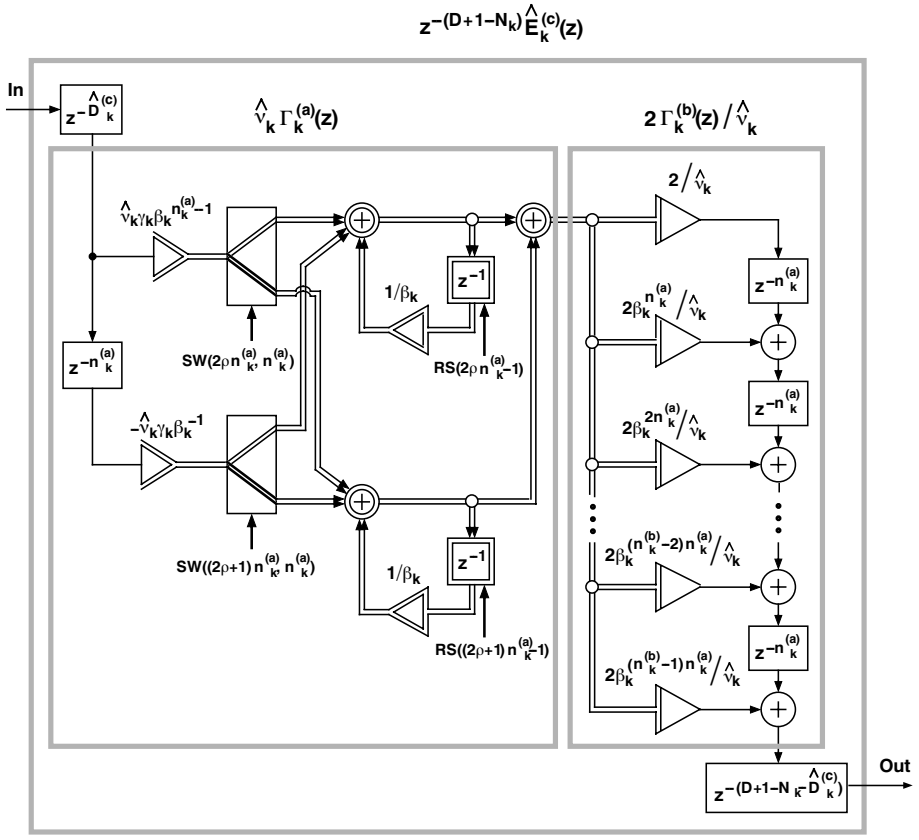

Figure 3.6: Implementation for $z^{-(D+1-N_k)}\widehat{E}_k^{(c)}(z)$ in Cascade Structure I of Fig. 3.2.

where

$$W(l) = \frac{1}{\beta_k} W(l-1) + V(l) \tag{3.37b}$$

with

$$V(l) = \widehat{\nu}_k \gamma_k (\beta_k)^{n_k^{(a)}-1} x(l) - \widehat{\nu}_k \gamma_k (\beta_k)^{-1} x(l - n_k^{(a)}). \tag{3.37c}$$

The corresponding real-arithmetic input-output relation is given in Table 3.3.

In the implementation of Fig. 3.6, complex data is present in the delay elements in the feedback loops as well as at the input and output of this loop. In the feedforward parts before and after the feedback loops, in turn, the data is real. When implementing the two copies of the feedback loop by demultiplexing the same filter, generating the complex

Table 3.3: Real Arithmetic Input-Output Relation for $\widehat{E}_k^{(c)}(z)$

$$y(l) = \sum_{i=0}^{n_k^{(b)}-1} \text{Re}\{\frac{2[(\beta_k)^{n_k^{(a)}}]^{(n_k^{(b)}-1-i)}}{\widehat{\nu}_k}\} W_{\text{re}}(l - in_k^{(a)})$$
$$- \sum_{i=0}^{n_k^{(b)}-1} \text{Im}\{\frac{2[(\beta_k)^{n_k^{(a)}}]^{(n_k^{(b)}-1-i)}}{\widehat{\nu}_k}\} W_{\text{im}}(l - in_k^{(a)}),$$

where

$$W_{\text{re}}(l) = \text{Re}\{1/\beta_k\} W_{\text{re}}(l-1) - \text{Im}\{1/\beta_k\} W_{\text{im}}(l-1) + V_{\text{re}}(l),$$

and

$$W_{\text{im}}(l) = \text{Im}\{1/\beta_k\} W_{\text{re}}(l-1) + \text{Re}\{1/\beta_k\} W_{\text{im}}(l-1) + V_{\text{im}}(l),$$

with

$$V_{\text{re}}(l) = \text{Re}\{\widehat{\nu}_k \gamma_k (\beta_k)^{n_k^{(a)}-1}\} x(l) - \text{Re}\{\widehat{\nu}_k \gamma_k (\beta_k)^{-1}\} x(l - n_k^{(a)}),$$

and

$$V_{\text{im}}(l) = \text{Im}\{\widehat{\nu}_k \gamma_k (\beta_k)^{n_k^{(a)}-1}\} x(l) - \text{Im}\{\widehat{\nu}_k \gamma_k (\beta_k)^{-1}\} x(l - n_k^{(a)}).$$

outputs of the feedback loops requires 8 real-valued multiplications. From this output, the overall output is obtained by using $2n_k^{(b)} - 1$ real valued multiplications[2]. The overall number of real-valued multiplications is thus $2n_k^{(b)}+7$. The number of real-valued additions is $2n_k^{(b)} + 2$.

The role of the scaling constant $\widehat{\nu}_k$ in the implementation of $z^{-(D+1-N_k)}\widehat{E}_k^{(c)}(z)$ is to avoid overflows in the real and imaginary parts of $W(l)$, denoted by $W_{\text{re}}(l)$ and $W_{\text{im}}(l)$. Ignoring the additional delay terms, the unit sample responses of the scaling transfer functions for these parts are given by

$$h_k^{(\text{real})}(l) = \begin{cases} \text{Re}\{\widehat{\nu}_k \gamma_k \beta_k^{(n_k^{(a)}-1-l)}\}, & 0 \le l \le n_k^{(a)} - 1, \\ 0, & \text{otherwise}, \end{cases} \quad (3.38a)$$

[2]The coefficient $2/\widehat{\nu}_k$ is real-valued.

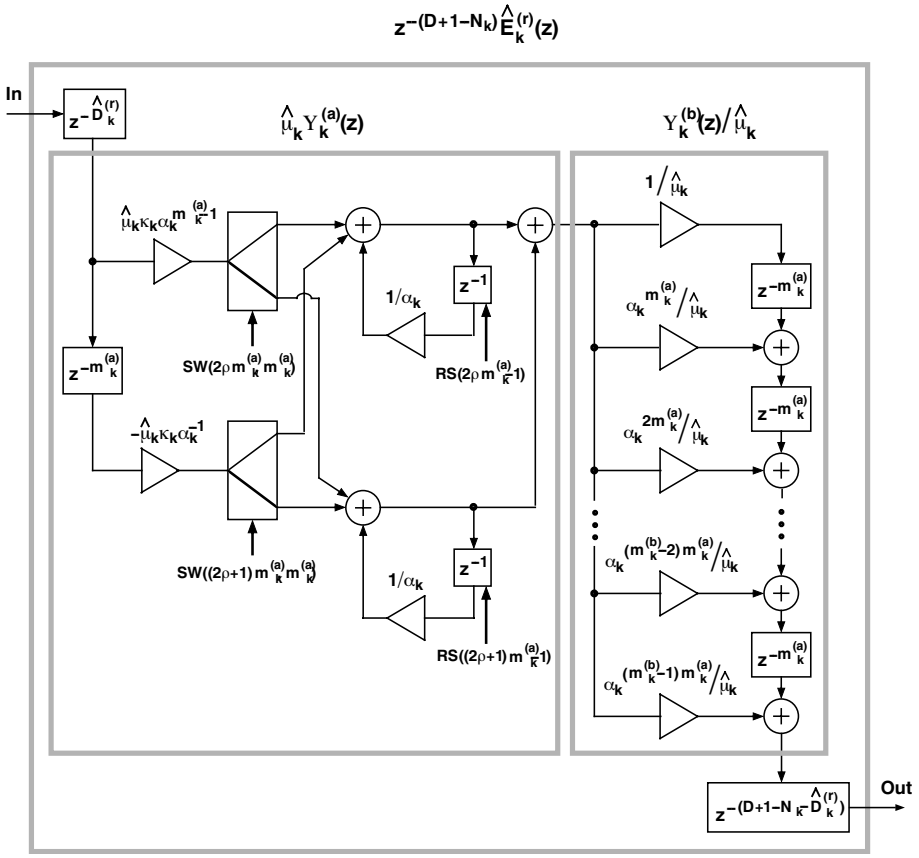

Figure 3.7: Implementation for $z^{-(D+1-M_k)}\widehat{E}_k^{(r)}(z)$ in Cascade Structure I of Fig. 3.2.

and

$$h_k^{(\text{imag})}(l) = \begin{cases} \text{Im}\{\widehat{\nu}_k \gamma_k \beta_k^{(n_k^{(a)}-1-l)}\}, & 0 \le l \le n_k^{(a)} - 1, \\ 0, & \text{otherwise}, \end{cases} \tag{3.38b}$$

respectively.

For $\widehat{E}_k^{(r)}(z)$ in turn, the scaling transfer function is given by

$$H_k(z) = \widehat{\mu}_k \Upsilon_k^{(a)}(z). \tag{3.39}$$

The corresponding implementation of $z^{-(D+1-M_k)}\widehat{E}_k^{(r)}(z)$ is depicted in Fig. 3.7, where $\widehat{\mu}_k$ is a scaling multiplier. This implementation requires $m_k^{(b)} + 3$ real-valued multiplications

and $m_k^{(b)} + 2$ real-valued additions by using one copy of the feedback loop. The number of real-valued multiplications and real-valued additions for the overall implementation are given in Table 3.2 on page 66. By factorizing the nonrecursive part of the unstable block, the required number of multipliers and adders can be further reduced. To elaborate, consider the identity

$$\sum_{l=0}^{m^{(1)}m^{(2)}\cdots m^{(I)}-1} x^{lm^{(a)}} = \prod_{i=1}^{I}\left[\sum_{l=0}^{m^{(i)}-1} x^{lM^{(i)}}\right], \tag{3.40}$$

where $M^{(1)} = m^{(a)}$ and $M^{(i)} = m^{(a)}m^{(1)}m^{(2)}\cdots m^{(i-1)}$ for $i = 2, 3, \ldots, I$. Through (3.40), a factorizing of $m_k^{(b)}$ as

$$m_k^{(b)} = \prod_{i=1}^{I_k} m_k^{(i)}, \tag{3.41a}$$

leads to a factorization of $\Upsilon_k^{(b)}(z)$ (given in (3.35d)) as

$$\Upsilon_k^{(b)}(z) = \prod_{i=1}^{I_k} \Upsilon_k^{(i)}(z), \tag{3.41b}$$

where

$$\Upsilon_k^{(i)}(z) = \sum_{l=0}^{m_k^{(i)}-1} (\alpha_k^{M_k^{(i)}})^{m_k^{(i)}-1-l} z^{-lM_k^{(i)}} \tag{3.41c}$$

with

$$M_k^{(i)} = m_k^{(a)} \prod_{l=1}^{i-1} m_k^{(l)}. \tag{3.41d}$$

Similarly, $\Gamma_k^{(b)}(z)$ as given by (3.36d) for

$$n_k^{(b)} = \prod_{j=1}^{J_k} n_k^{(j)} \tag{3.42a}$$

can be rewritten as

$$\Gamma_k^{(b)}(z) = \prod_{j=1}^{J_k} \Gamma_k^{(j)}(z), \tag{3.42b}$$

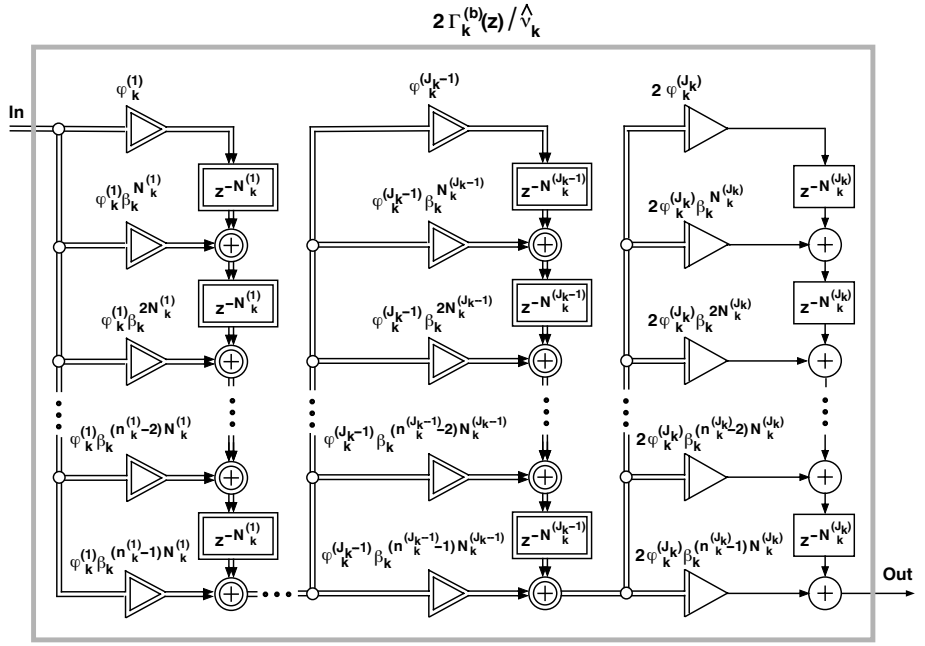

Figure 3.8: Decomposition of $\Upsilon_k^{(b)}(z)$.

where

$$\Gamma_k^{(j)}(z) = \sum_{l=0}^{n_k^{(j)}-1} (\beta_k^{N_k^{(j)}})^{n_k^{(j)}-1-l} z^{-lN_k^{(j)}} \tag{3.42c}$$

with

$$N_k^{(j)} = n_k^{(a)} \prod_{l=1}^{j-1} n_k^{(l)}. \tag{3.42d}$$

Implementations for the above transfer functions including the scaling constants are shown in Figs. 3.8 and 3.9. For $\Gamma_k^{(b)}(z)$, the constants $\varphi_k^{(j)}$ for $j = 1, 2 \ldots, J_k - 1$ are determined such that overflows are avoided at the outputs of the corresponding blocks $\Gamma_k^{(j)}(z)$, whereas

$$\varphi_k^{(J_k)} = 1 / \Big[\widehat{\nu}_k \prod_{j=1}^{J_k-1} \varphi_k^{(j)} \Big]. \tag{3.43}$$

The impulse responses of the scaling transfer functions to the real and imaginary parts in the output of $\Gamma_k^{(j)}(z)$ are given by

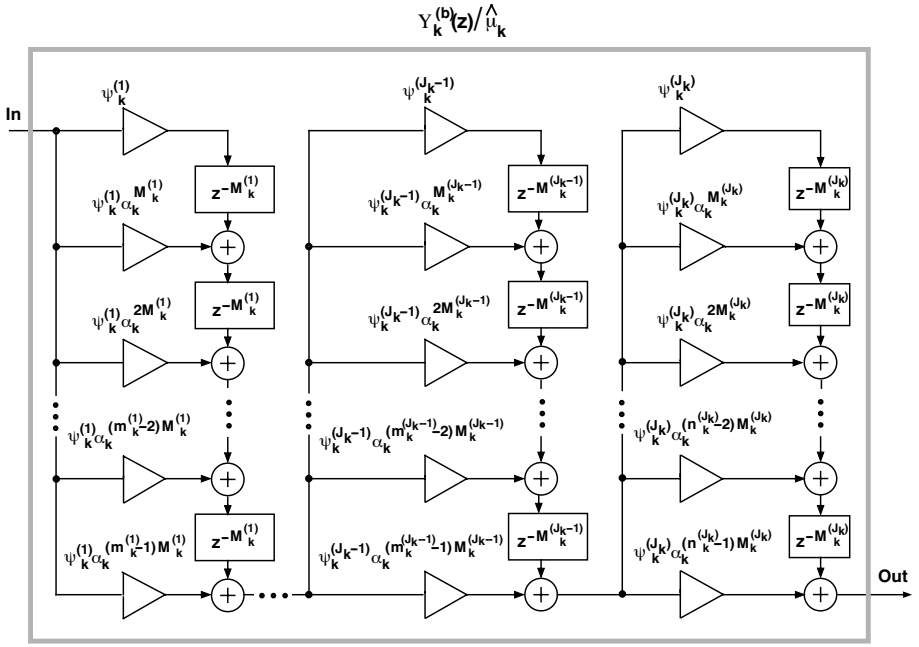

Figure 3.9: Decomposition of $\Gamma_k^{(b)}(z)$.

$$h_k^{(\text{real})}(j,l) = \begin{cases} \text{Re}\{\widehat{\nu}_k \Big[\prod_{i=1}^{j} \varphi_k^{(i)}\Big] \gamma_k \beta_k^{(N_k^{(j+1)}-1-l)}\}, & 0 \le l \le N_k^{(j+1)} - 1, \\ 0, & \text{otherwise,} \end{cases} \tag{3.44a}$$

and

$$h_k^{(\text{imag})}(j,l) = \begin{cases} \text{Im}\{\widehat{\nu}_k \Big[\prod_{i=1}^{j} \varphi_k^{(i)}\Big] \gamma_k \beta_k^{(N_k^{(j+1)}-1-l)}\}, & 0 \le l \le N_k^{(j+1)} - 1, \\ 0, & \text{otherwise.} \end{cases} \tag{3.44b}$$

For $\Upsilon_k^{(b)}(z)$, the scaling constants $\psi_k^{(i)}$ are determined in a similar manner.

Table 3.4 compares the multistage (factorized) implementations to the single-stage implementation in two cases. In the first case, the subfilters are scaled as shown in Figs. 3.8 and 3.9, whereas in the second case $\varphi_k^{(j)} = 1$ for $j = 1, 2, \ldots J_k - 1$ and $\psi_k^{(i)} = 1$ for $i = 1, 2, \ldots I_k - 1$ and the scalings of the overall filters are performed with the aid of the

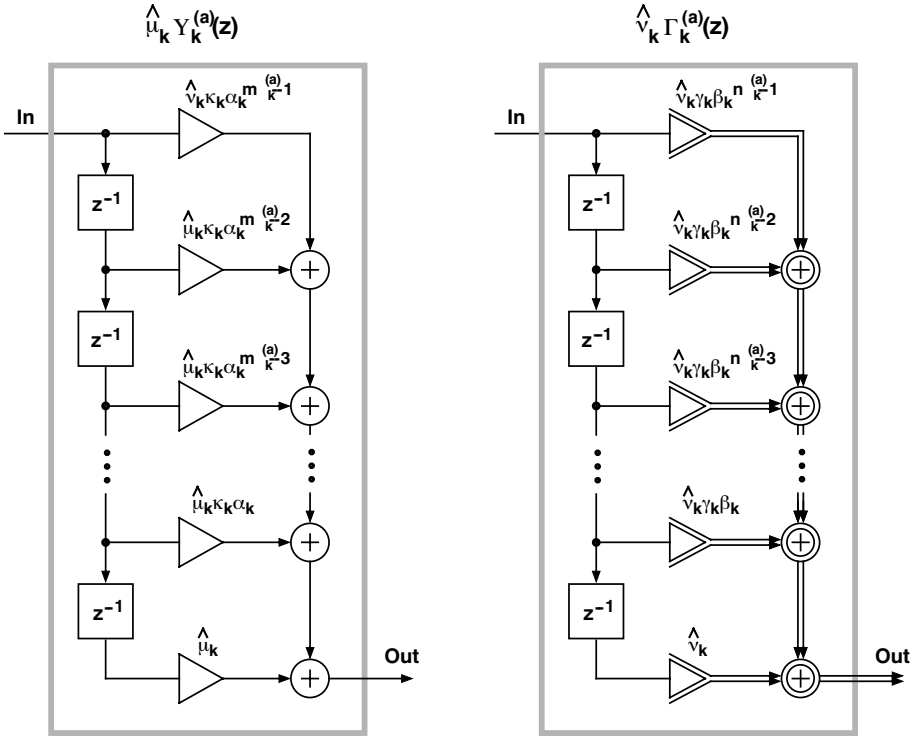

Figure 3.10: Non-recursive implementations for $\Upsilon_k^{(a)}(z)$ and $\Gamma_k^{(a)}(z)$.

scaling constant $\widehat{\nu}_k$ and $\widehat{\mu}_k$. It is seen that the factorized structures provide considerable savings when the $m_k^{(i)}$'s and the $n_k^{(j)}$'s are small integers (two or three).

Alternatively, the scaled $\Upsilon_k^{(a)}(z)$ and $\Gamma_k^{(a)}(z)$ can be rewritten in the following non-recursive forms:

$$\Upsilon_k^{(a)}(z) = \sum_{l=0}^{m_k^{(a)}-1} \widehat{\mu}_k \kappa_k (\alpha_k)^{m_k^{(a)}-1-l} z^{-l} \tag{3.45a}$$

and

$$\Gamma_k^{(a)}(z) = \sum_{l=0}^{n_k^{(a)}-1} \widehat{\nu}_k \gamma_k (\beta_k)^{n_k^{(a)}-1-l} z^{-l}, \tag{3.45b}$$

respectively. The comparison of Table 3.4 shows that the non-recursive implementations shown in Fig. 3.10 become more attractive only for relatively small values of $m_k^{(a)}$ and $n_k^{(a)}$.

Table 3.4: Comparison between Different Implementations of $\widehat{E}_k^{(r)}(z)$ and $\widehat{E}_k^{(c)}(z)$ for Cascade Structure I

	Number of Real-Valued Multiplications	Number of Real-Valued Additions
$\Upsilon_k^{(b)}(z)$		
Single-Stage	$\prod_{i=1}^{I_k} m_k^{(i)}$	$\left[\prod_{i=1}^{I_k} m_k^{(i)}\right] - 1$
Multistage General Scaling	$\sum_{i=1}^{I_k} m_k^{(i)}$	$\sum_{i=1}^{I_k}(m_k^{(i)} - 1)$
Multistage One Scaler	$1 + \sum_{i=1}^{I_k}(m_k^{(i)} - 1)$	$\sum_{i=1}^{I_k}(m_k^{(i)} - 1)$
$\Upsilon_k^{(a)}(z)$		
Recursive	3	3
Non-recursive	$m_k^{(a)}$	$m_k^{(a)} - 1$
$\Gamma_k^{(b)}(z)$		
Single-Stage	$2\left[\prod_{j=1}^{J_k} n_k^{(j)}\right] - 1$	$\left[\prod_{j=1}^{J_k} n_k^{(j)}\right] - 1$
Multistage General Scaling	$\sum_{j=1}^{J_k-1}(4n_k^{(j)} - 2)$ $+2n_k^{(J_k)} - 1$	$+\sum_{j=1}^{J_k-1} 2(n_k^{(j)} - 1)$ $+n_k^{(J_k)} - 1$
Multistage One Scaler	$\sum_{j=1}^{J_k-1} 4(n_k^{(j)} - 1)$ $+2n_k^{(J_k)} - 1$	$\sum_{j=1}^{J_k-1} 2(n_k^{(j)} - 1)$ $+n_k^{(J_k)} - 1$
$\Gamma_k^{(a)}(z)$		
Recursive	8	8
Non-recursive	$2n_k^{(a)} - 1$	$2n_k^{(a)} - 2$

Implementation of Delays

In the overall implementation of Fig. 3.2, the block delays z^{-D}, z^{-M_k}, and z^{-N_k} as well as the delays in $A(z)$ can be obtained from a common delay term $z^{-(D+1)}$. This is because D is equal to the maximum of the M_k's and N_k's. Hence, the implementation of $c_0 z^{-D}$, $A(z)$, $E_k^{(r)}(z)$'s and $E_k^{(c)}(z)$'s requires $D + 1 + K_R + 2K_C$ delay elements.

The block delays $z^{-\widehat{D}_k^{(c)}}$, $z^{-\widehat{D}_k^{(r)}}$, $z^{-(D+1-N_k-\widehat{D}_k^{(c)})}$, and $z^{-(D+1-M_k-\widehat{D}_k^{(r)})}$ in the implementations of Figs. 3.6 and 3.7 are used to minimize the number of delays required in implementing the overall $\widehat{E}_k(z)$ in the structure of Fig. 3.2. This minimization can be accomplished by using a delay term $z^{-\widetilde{D}_{\text{in}}}$ at the input of $\widehat{E}_k(z)$, where

$$\widetilde{D}_{\text{in}} = \max\{\max_{1\le k\le K_R}\{m_k^{(a)}\}, \max_{1\le k\le K_C}\{n_k^{(a)}\}\}, \tag{3.46a}$$

and a delay term $z^{-\widetilde{D}_{\text{out}}}$ at the output, where

$$\widetilde{D}_{\text{out}} = \max\Big\{\max_{1\le k\le K_R}\{m_k^{(a)}(m_k^{(b)}-1)\}, \max_{1\le k\le K_C}\{n_k^{(a)}(n_k^{(b)}-1)\}\Big\}. \tag{3.46b}$$

By selecting

$$\widehat{D}_k^{(r)} = \min\{\widetilde{D}_{\text{in}} - m_k^{(a)}, D+1-M_k\}, \quad k=1,2,\ldots,K_R, \tag{3.47a}$$

and

$$\widehat{D}_k^{(c)} = \min\{\widetilde{D}_{\text{in}} - n_k^{(a)}, D+1-N_k\}, \quad k=1,2,\ldots,K_C, \tag{3.47b}$$

the $z^{-\widehat{D}_k^{(c)}}$'s, $z^{-\widehat{D}_k^{(r)}}$'s, $z^{-n_k^{(a)}}$'s and $z^{-m_k^{(a)}}$'s that are located before the feedback loops can be obtained from $z^{-\widetilde{D}_{\text{in}}}$. Correspondingly, the $z^{-n_k^{(b)}}$'s, $z^{-m_k^{(b)}}$'s that are located after the feedback loops as well as the $z^{-(D+1-N_k-\widehat{D}_k^{(c)})}$'s and $z^{-(D+1-M_k-\widehat{D}_k^{(r)})}$'s can be obtained from a common delay term $z^{-\widetilde{D}_{\text{out}}}$. In most cases, $\widetilde{D}_{\text{in}}$ and $\widetilde{D}_{\text{out}}$ are determined by one or two outermost pole pairs. In this case, $\widetilde{D}_{\text{out}} = D - \widetilde{D}_{\text{in}}$. The number of delay elements required by $E_k(z)$, $\widehat{E}_k(z)$, and the overall filter in this case are shown in Table 3.2.

3.3.2 Cascade Structure II

Similar to Cascade Structure I, Cascade Structure II, shown in Fig. 3.3 is composed of two blocks in cascade, where the first (second) block consists of parallel connection of stable (unstable) IIR filters. As in Section 3.3.1 for Cascade Structure I, the implementations of the stable and the unstable parts of this structure are considered separately in this section.

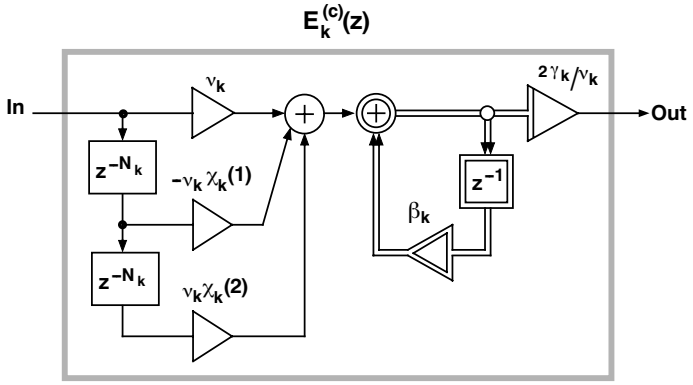

Figure 3.11: Implementation for $E_k^{(c)}(z)$ in the Cascade Structure II.

Stable Blocks

For $E_k^{(c)}(z)$, as given by (3.26), the input-output relation can be described by the following difference equations:

$$y(l) = \frac{2}{\nu_k}\text{Re}\Big\{\gamma_k W(l)\Big\}, \tag{3.48a}$$

where

$$W(l) = \beta_k W(l-1) + v(l), \tag{3.48b}$$

and

$$v(l) = \nu_k x(l) - \nu_k \chi_k(1) x(l-N_k) + \nu_k \chi_k(2) x(l-2N_k) \tag{3.48c}$$

with

$$\chi_k(1) = 2(r_k)^{N_k}\cos(N_k\phi_k), \tag{3.48d}$$

and

$$\chi_k(2) = (r_k)^{2N_k}. \tag{3.48e}$$

Here, ν_k is a scaling constant and only $W(l)$ is a complex variable. An implementation for $E_k^{(c)}(z)$ is shown in Fig. 3.11, requiring 8 real-valued multiplications and 4 real-valued additions.

For $A(z)$ and the $E_k^{(r)}(z)$'s, the implementation costs are the same as that for Cascade Structure I as shown in Table 3.5. Furthermore, similarly to Cascade Structure I, z^{-D} as well as the delays in $A(z)$ and in the feedforward parts of the $E_k^{(r)}(z)$'s and the $E_k^{(c)}(z)$'s can be obtained from a common block delay $z^{-(D+1)}$, where D is given by (3.29). The overall number of real delays is shown in Table 3.5.

The unit sample responses of the scaling transfer functions to the real and imaginary parts of the feedback loop are given by

$$h_k^{(\text{real})}(l) = \begin{cases} 2\nu_k \,\text{Re}\big\{(\beta_k)^l\big\}, & 0 \le l \le N_k - 1, \\ -2\nu_k \,\text{Re}\big\{(\beta_k^*)^{N_k}(\beta_k)^{(l-N_k)}\big\}, & N_k \le l \le 2N_k - 1, \\ 0, & \text{otherwise,} \end{cases} \tag{3.49a}$$

and

$$h_k^{(\text{imag})}(l) = \begin{cases} 2\nu_k \,\text{Im}\big\{(\beta_k)^l\big\}, & 0 \le l \le N_k - 1, \\ -2\nu_k \,\text{Im}\big\{(\beta_k^*)^{N_k}(\beta_k)^{(l-N_k)}\big\}, & N_k \le l \le 2N_k - 1, \\ 0, & \text{otherwise,} \end{cases} \tag{3.49b}$$

respectively.

Unstable Blocks

The output noise variance of $\widehat{E}_k^{(c)}(z)$ as given by (3.30b)–(3.30e), can be reduced by factorizing $\widehat{\Omega}_k(z)$, as given by (3.30c), for

$$N_k = n_k^{(a)} n_k^{(b)} \tag{3.50}$$

as

$$\widehat{\Omega}_k(z) = \widehat{\Omega}_k^{(a)}(z)\widehat{\Omega}_k^{(b)}(z), \tag{3.51a}$$

where

$$\widehat{\Omega}_k^{(a)}(z) = (r_k)^{2n_k^{(a)}} - 2(r_k)^{n_k^{(a)}} \cos(n_k^{(a)}\phi_k) z^{-n_k^{(a)}} + z^{-2n_k^{(a)}} \tag{3.51b}$$

Table 3.5: Arithmetic Operations for Cascade Structure II

	Number of Real-Valued Multiplications	Number of Real-Valued Additions
$E_k(z)$		
$E_k^{(r)}(z)$	4	2
$E_k^{(c)}(z)$	9	4
$A_k(z)$	$L-K+1$	$L-K$
$\widehat{E}_k(z)$		
$\widehat{E}_k^{(r)}(z)$	$m_k^{(b)}+3$	$m_k^{(b)}+1$
$\widehat{E}_k^{(c)}(z)$	$2n_k^{(b)}+8$	$2n_k^{(b)}+4$
$\widehat{A}_k(z)$	$L-K+1$	$L-K$
Overall	$2(L-K+1)+1$	$2(L-K)+1$
Filter	$+7K_R+\sum_{k=1}^{K_R} m_k^{(b)}$	$+4K_R+\sum_{k=1}^{K_R} m_k^{(b)}$
	$+17K_C+2\sum_{k=1}^{K_C} n_k^{(b)}$	$9K_C+2\sum_{k=1}^{K_C} n_k^{(b)}$
	Number of Delay Elements	
$E_k(z)$ and $c_0 z^{-D}$	$D+1+K_R+2K_C$	
$\widehat{E}_k(z)$	$D+1+2K_R+4K_C$	
Overall Filter	$2D+2+3K_R+6K_C$	

and

$$\widehat{\Omega}_k^{(b)}(z)=\sum_{l=0}^{2n_k^{(b)}-2}\eta_k(l)z^{-ln_k^{(a)}} \tag{3.51c}$$

with

$$\eta_k(l)=\begin{cases}(r_k)^{n_k^{(a)}(2n_k^{(b)}-2-l)}\dfrac{\sin[(l+1)n_k^{(a)}\phi_k]}{\sin[n_k^{(a)}\phi_k]}, & 0\le l\le n_k^{(b)}-1,\\ (r_k)^{n_k^{(a)}(2n_k^{(b)}-2-l)}\dfrac{\sin[(2n_k^{(b)}-1-l)n_k^{(a)}\phi_k]}{\sin[n_k^{(a)}\phi_k]}, & n_k^{(b)}\le l\le 2n_k^{(b)}-2.\end{cases} \tag{3.51d}$$

Based on these equations, $E_k^{(c)}(z)$ can be rewritten as

$$\widehat{E}_k^{(c)}(z) = \left[\widehat{\Omega}_k^{(a)}(z)\Re\Big\{\Gamma_k^{(a)}(z)\Big\}\right][2\widehat{\Omega}_k^{(b)}(z)], \tag{3.52a}$$

where

$$\Gamma_k^{(a)}(z) = \frac{\widehat{\gamma}_k}{1-(1/\beta_k)z^{-1}}. \tag{3.52b}$$

By including the scaling constant $\widehat{\nu}_k^{(1)}$ and $\widehat{\nu}_k^{(2)}$, the input-output relation for this transfer function can be described by the following difference equations:

$$y(l) = \sum_{j=0}^{2n_k^{(b)}-2} \frac{2\eta_k(l)}{\widehat{\nu}_k^{(1)}\widehat{\nu}_k^{(2)}} w(l-jn_k^{(a)}), \tag{3.53a}$$

where

$$w(l) = \text{Re}\Big\{\widehat{\nu}_k^{(2)}\widehat{\gamma}_k W(l)\Big\}, \tag{3.53b}$$

$$W(l) = (1/\beta_k)W(l-1) + v(l), \tag{3.53c}$$

and

$$v(l) = \widehat{\nu}_k^{(1)}\chi_k(0)x(l) - \widehat{\nu}_k^{(1)}\chi_k(1)x(l-n_k^{(a)}) + \widehat{\nu}_k^{(1)}x(l-2n_k^{(a)}) \tag{3.53d}$$

with

$$\chi_k(0) = (r_k)^{2n_k^{(a)}} \tag{3.53e}$$

and

$$\chi_k(1) = 2(r_k)^{n_k^{(a)}}\cos(n_k^{(a)}\phi_k). \tag{3.53f}$$

An implementation applying the principle of switching and resetting is depicted in Fig. 3.12. The scaling constant $\widehat{\nu}_k^{(1)}$ is determined to avoid overflows in the real and imaginary parts of $W(l)$ based on the fact that the unit sample responses of corresponding scaling transfer functions are given by

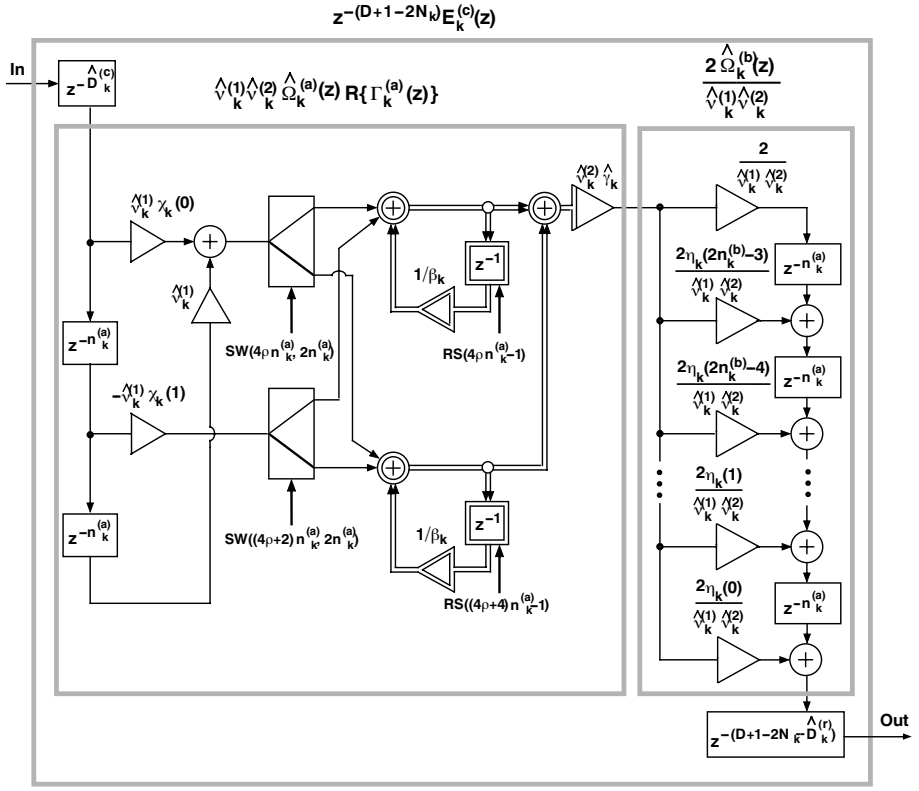

Figure 3.12: Implementation for $z^{-(D+1-2N_k)}\widehat{E}_k^{(c)}(z)$ in the Cascade Structure II.

$$h_k^{(\text{real})}(l) = \begin{cases} \text{Re}\{\widehat{\nu}_k^{(1)}(\beta_k\beta_k^*)^{n_k^{(a)}}(\beta_k)^{-l}\}, & 0 \le l \le n_k^{(a)} - 1, \\ \text{Re}\{-\widehat{\nu}_k^{(1)}(\beta_k)^{2n_k^{(a)}-l}\}, & n_k^{(a)} \le l \le 2n_k^{(a)} - 1, \\ 0, & \text{otherwise,} \end{cases} \tag{3.54a}$$

and

$$h_k^{(\text{imag})}(l) = \begin{cases} \text{Im}\{\widehat{\nu}_k^{(1)}(\beta_k\beta_k^*)^{n_k^{(a)}}(\beta_k)^{-l}\}, & 0 \le l \le n_k^{(a)} - 1, \\ \text{Im}\{-\widehat{\nu}_k^{(1)}(\beta_k)^{2n_k^{(a)}-l}\}, & n_k^{(a)} \le l \le 2n_k^{(a)} - 1, \\ 0, & \text{otherwise.} \end{cases} \tag{3.54b}$$

The role of $\widehat{\nu}_k^{(2)}$, in turn, is to avoid overflows in $w(l)$. The corresponding unit sample response from the input to this variable is obtained from (3.54a) by using the substitution

$$\widehat{\nu}_k^{(1)} \rightarrow \widehat{\nu}_k^{(1)}\widehat{\nu}_k^{(2)}\widehat{\gamma}_k.$$

The number of real-valued multiplications, real-valued additions, and delay elements for $\widehat{E}^{(c)}(z)$ and for the overall filter are considered in Table 3.5. The role of the delay blocks $z^{-n_k^{(b)}}$, $z^{-m_k^{(b)}}$, $z^{-(D+1-2N_k-\widehat{D}_k^{(c)})}$, and $z^{-(D+1-M_k-\widehat{D}_k^{(r)})}$ is the same as for Cascade Structure I.

Similar to Cascade Structure I, the arithmetic complexity in the nonrecursive part of Cascade Structure II can be reduced by factorizing $\widehat{\Omega}_k^{(b)}(z)$. In particular, for

$$n_k^{(b)} = \prod_{j=1}^{J_k} n_k^{(j)}, \tag{3.55}$$

$\widehat{\Omega}_k^{(b)}(z)$ can be decomposed as

$$\widehat{\Omega}_k^{(b)}(z) = \prod_{j=1}^{J_k} \widehat{\Omega}_k^{(j)}(z), \tag{3.56a}$$

where

$$\widehat{\Omega}_k^{(j)}(z) = \sum_{l=0}^{2n_k^{(j)}-2} \eta_k^{(j)}(l) z^{-lN_k^{(j)}}, \tag{3.56b}$$

$$N_k^{(j)} = n_k^{(a)} \prod_{l=1}^{j-1} n_k^{(l)}, \tag{3.56c}$$

and

$$\eta_k^{(j)}(l) = \begin{cases} (r_k)^{N_k^{(j)}(2n_k^{(j)}-2-l)} \dfrac{\sin[(l+1)N_k^{(j)}\phi_k]}{\sin[N_k^{(a)}\phi_k]}, & 0 \le l \le n_k^{(j)}-1, \\ (r_k)^{N_k^{(j)}(2n_k^{(j)}-2-l)} \dfrac{\sin[(2n_k^{(j)}-1-l)N_k^{(j)}\phi_k]}{\sin[N_k^{(a)}\phi_k]}, & n_k^{(j)} \le l \le 2n_k^{(j)}-2. \end{cases} \tag{3.56d}$$

An implementation for $\widehat{\Omega}_k^{(b)}(z)$ including the scaling constants $\varphi_k^{(j)}$ is depicted in Fig. 3.13. For $j = 1, 2, \ldots J_k - 1$, the constants $\varphi_k^{(j)}$ are determined to avoid overflows at the output of $\widehat{\Omega}_k^{(j)}(z)$. The corresponding unit sample response from the filter input is obtained from

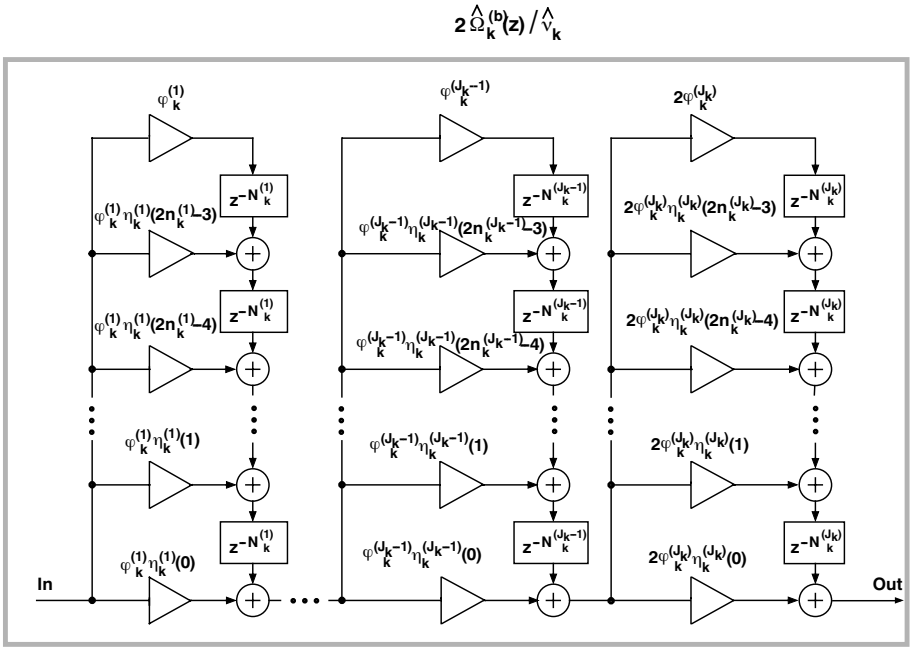

Figure 3.13: Implementation for a multistage $\widehat{\Omega}_k^{(b)}(z)$.

(3.54) by using the substitutions $n_k^{(a)} \rightarrow N_k^{(j+1)}$ and $\widehat{\nu}_k^{(1)} \rightarrow \widehat{\nu}_k^{(1)}\widehat{\nu}_k^{(2)}\widehat{\gamma}_k \prod_{l=1}^{j-1} \varphi_k^{(j)}$. Using these substitutions, $\varphi_k^{(J_k)}$ becomes

$$\varphi_k^{(J_k)} = \frac{1}{\nu_k^{(1)}\nu_k^{(2)} \prod_{j=1}^{J_k-1} \varphi_k^{(j)}}. \tag{3.57}$$

Table 3.6 compares the multistage (factorized) and single-stage implementations of $\widehat{\Omega}_k^{(b)}(z)$. It is again observed that the multistage implementation is advantageous for small values of the $n_k^{(j)}$'s.

Alternatively, as for Cascade Structure I, the use of feedback loops can be avoided by exploiting the following identity:

$$\begin{aligned}\nu_k^{(1)}\nu_k^{(2)}\widehat{\Omega}_k^{(a)}(z)\Re\Big\{\Gamma_k^{(a)}(z)\Big\} = &\Big[\sum_{l=0}^{2n_k^{(a)}-2} \nu_k^{(1)}\eta_k^{(a)}(l)z^{-l}\Big] \\ &\times\Big[\nu_k^{(2)}\,\mathrm{Re}\{\widehat{\gamma}_k\} + \nu_k^{(2)}\,\mathrm{Re}\{\widehat{\gamma}_k\beta_k^*\}z^{-1}\Big],\end{aligned} \tag{3.58a}$$

Table 3.6: Comparison between Different Implementations of $\widehat{E}_k^{(c)}(z)$ in Cascade Structure II

	Number of Real-Valued Multiplications	Number of Real-Valued Additions
$\widehat{\Omega}_k^{(b)}(z)$		
Single-Stage	$2\big[\prod_{i=1}^{J_k} n_k^{(j)}\big] - 1$	$\big[\prod_{j=1}^{J_k} n_k^{(j)}\big] - 2$
Multistage General Scaling	$\sum_{j=1}^{J_k}(2n_k^{(j)} - 1)$	$\sum_{j=1}^{J_k}(2n_k^{(j)} - 2)$
Multistage Two Scalers	$1 + \sum_{j=1}^{J_k}(2n_k^{(j)} - 2)$	$\sum_{j=1}^{J_k}(2n_k^{(j)} - 2)$
$\Gamma_k^{(a)}(z)$		
Recursive	8	10
Non-recursive	$2n_k^{(a)} + 1$	$2n_k^{(a)} - 1$

where

$$\eta_k^{(a)}(l) = \begin{cases} (r_k)^{2n_k^{(a)}-2-l}\dfrac{\sin[(l+1)\phi_k]}{\sin(\phi_k)}, & 0 \le l \le n_k^{(a)} - 1 \\ (r_k)^{2n_k^{(a)}-2-l}\dfrac{\sin[(2n_k^{(a)}-1-l)\phi_k]}{\sin(\phi_k)}, & n_k^{(a)} \le l \le 2n_k^{(a)} - 2. \end{cases} \quad (3.58b)$$

This transfer function is a cascade of two conventional FIR filter transfer functions with real coefficients and real data. As seen from Table 3.6, the nonrecursive implementation should be considered only for small values of the $n_k^{(a)}$'s.

3.4 Noise Analysis

In order to analyze the output noise due to the multiplication roundoff errors taking place in the feedback loops of $\widehat{E}_k^{(c)}(z)$, as given by (3.36b) and (3.52a) and presented in Figs. 3.6 and 3.12, the following notations are used (see Fig. 3.14):

$$\Xi_{k,\text{in}}^{(1)}(l) = \xi_{k,\text{re}}^{(1)}(l) + j\xi_{k,\text{im}}^{(1)}(l) \quad (3.59a)$$

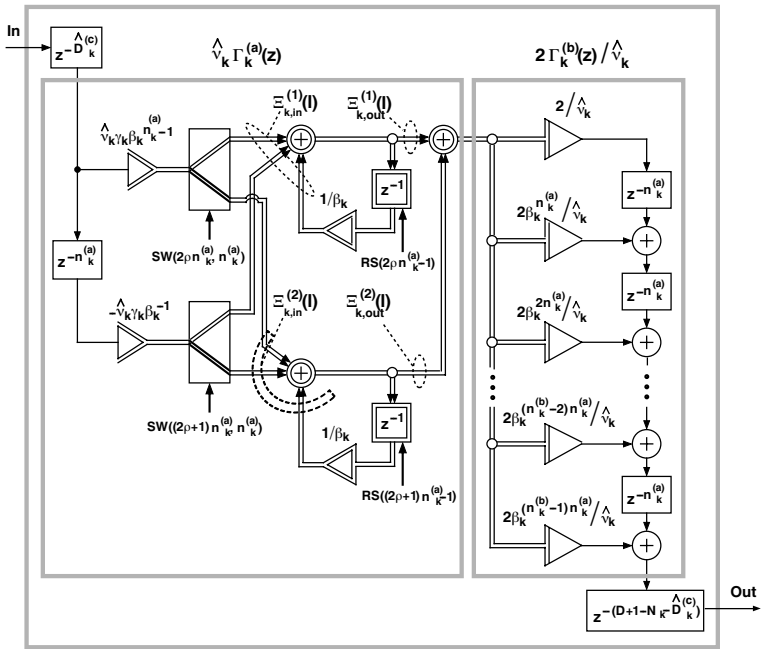

Figure 3.14: Illustration of noise signals in the implementation of $z^{-(D+1-N_k)}\,\widehat{E}_k^{(c)}(z)$.

and

$$\Xi_{k,\text{in}}^{(2)}(l) = \xi_{k,\text{re}}^{(2)}(l) + j\xi_{k,\text{im}}^{(2)}(l). \tag{3.59b}$$

Here, $\xi_{k,\text{re}}^{(1)}(l)$ and $\xi_{k,\text{im}}^{(1)}(l)$ ($\xi_{k,\text{re}}^{(2)}(l)$ and $\xi_{k,\text{im}}^{(2)}(l)$) are the multiplication roundoff errors resulting from quantizing the real and imaginary parts in the feedback loop in the upper (lower) branch respectively. In fact, the complex noises $\Xi_{k,\,\text{in}}^{(1)}(l)$ and $\Xi_{k,\,\text{in}}^{(2)}(l)$ consist of both a real time-dependent noise generated in the nonrecursive part of the above-mentioned $\widehat{E}_k^{(c)}(z)$'s, and the corresponding complex time-dependent noise generated in the recursive part of $\widehat{E}_k^{(c)}(z)$'s. For the $\widehat{E}_k^{(r)}(z)$'s as given by (3.35b) and presented in Fig. 3.7, the feedback loop consists only of the real part, and consequently $\xi_{k,\text{im}}^{(1)}(l) \equiv \xi_{k,\text{im}}^{(2)}(l) = 0$. In what follows, the roundoff noise generated in the proposed structures are analyzed in detail.

3.4.1 Cascade Structure I

Because of the switching and resetting, the (complex) roundoff error at the output of the upper feedback loop of $z^{-(D+1-N_k)}\widehat{E}_k^{(c)}(z)$ (presented in Fig. 3.6), denoted by $\Xi_{k,\text{out}}^{(1)}(l)$, is zero at the time instants

$$l = 2\rho n_k^{(a)} - 1, \tag{3.60a}$$

where ρ is an integer, whereas for $0 \le \lambda \le 2n_k^{(a)} - 2$,

$$\Xi_{k,\text{out}}^{(1)}(2\rho n_k^{(a)} + \lambda) = \sum_{i=0}^{\lambda} (1/\beta_k)^i \, \Xi_{k,\text{in}}^{(1)}(2\rho n_k^{(a)} + \lambda - i). \tag{3.60b}$$

Similarly, the output error at the output of the lower feedback loop of $z^{-(D+1-N_k)}\widehat{E}_k^{(c)}(z)$, that is, $\Xi_{k,\text{out}}^{(2)}(l)$ is zero at the time instants

$$l = (2\rho + 1)n_k^{(a)} - 1, \tag{3.61a}$$

while for $0 \le \lambda \le 2n_k^{(a)} - 2$,

$$\Xi_{k,\text{out}}^{(2)}((2\rho + 1)n_k^{(a)} + \lambda) = \sum_{i=0}^{\lambda} (1/\beta_k)^i \Xi_{k,\text{in}}^{(2)}((2\rho + 1)n_k^{(a)} + \lambda - i). \tag{3.61b}$$

Equations (3.60) and (3.61) apply to $z^{-(D+1-M_k)}\widehat{E}_k^{(r)}(z)$ by $n_k^{(a)} \to m_k^{(a)}$ and $\beta_k \to \alpha_k$.

For the structure of Fig.3.7, the overall noise at time l can be expressed as

$$\xi_k^{(\text{ove})}(l) = \sum_{i=0}^{m_k^{(b)}-1} \text{Re}\left\{ \frac{[(\alpha_k)^{m_k^{(a)}}]^{(m_k^{(b)}-1-i)}}{\widehat{\mu}_k} \times \right.$$
$$\left. \left[\Xi_k^{(1)}(l - im_k^{(a)}, 0, 2m_k^{(a)}) + \Xi_k^{(2)}(l - im_k^{(a)}, m_k^{(a)}, 2m_k^{(a)}) \right] \right\}, \tag{3.62a}$$

where

$$\Xi_k^{(m)}(l, j, n) = \begin{cases} 0, & (l - j) \bmod (n) = n - 1, \\ \displaystyle\sum_{i=0}^{(l-j)\bmod(n)} (1/\alpha_k)^i \Xi_{\text{in}}^{(m)}(l - i), & \text{otherwise.} \end{cases} \tag{3.62b}$$

By considering the effects of switching and resetting, and by taking into account the fact that $\Xi_{k,\text{in}}^{(1)}(l)$ and $\Xi_{k,\text{in}}^{(2)}(l)$ consist of the sum of the input noises from the demultiplexers and the roundoff noise generated in the coefficient of the recursive filter, the variance of the output noise at the input of $\Upsilon_k^{(b)}(z)$ at any arbitrary time l can be expressed as:

$$(\sigma^2)_k^{(\text{out})}(l) = (\sigma^2)_k^{(1)}(l) + (\sigma^2)_k^{(1)}(l - m_k^{(a)}) \tag{3.63a}$$

where

$$(\sigma^2)_k^{(1)}(l) = \begin{cases} 0, & \lambda(l) = 2m_k^{(a)} - 1, \\ \sigma_e^2 \left[\dfrac{2 - 2\alpha_k^{-2\lambda(l)-2}}{1 - \alpha_k^{-2}} - \dfrac{1}{\alpha_k^{2\lambda(l)}} \right], & \text{otherwise,} \end{cases} \tag{3.63b}$$

and

$$\lambda(l) = l \bmod \left(2m_k^{(a)}\right). \tag{3.63c}$$

In (3.63a) and (3.63b), $(\sigma^2)_k^{(1)}(l)$ is the noise variance at the output of the upper recursive filter, $(\sigma^2)_k^{(\text{out})}(l)$ is the total noise variance at the input to $\Upsilon_k^{(b)}(z)$ and σ_e^2 is the noise variance generated by each quantizer.

Based on (3.59a)–(3.63b), the total noise variance (after the transient time of $\Upsilon_k^{(b)}(z)$ has elapsed) can be expressed as:

$$T_1(\sigma^2)_k^{(\text{out})}(l) + m_k^{(b)}\sigma_e^2, \tag{3.64a}$$

where

$$T_1 = \sum_{i=0}^{m_k^{(b)}-1} \left(\frac{\alpha_k^{im_k^{(a)}}}{\hat{\mu}_k} \right)^2 = \frac{1}{\hat{\mu}_k^2} \frac{1 - \alpha_k^{2m_k^{(a)}m_k^{(b)}}}{1 - \alpha_k^{2m_k^{(a)}}}. \tag{3.64b}$$

The variance takes the highest values at $l = m_k^{(a)} - 2$.

For the case of the structure of Fig.3.6, the overall noise at time l can be expressed according to (3.62a) and (3.62b) by using the substitutions $\widehat{\mu}_k \rightarrow \widehat{\nu}_k/2, m_k^{(b)} \rightarrow n_k^{(b)}$ and

$\alpha_k \to \beta_k$. In this case, $\sigma^2_{k,\text{up}}(l)$ for any arbitrary l is given by

$$(\sigma^2)^{(1)}_k(l) = \begin{cases} 0, & \lambda = 2n_k^{(a)} - 1, \\ (2A_\lambda + A_{\lambda+1})\sigma_e^2, & \text{otherwise}, \end{cases} \tag{3.65a}$$

where

$$\lambda = l \bmod \left(2n_k^{(a)}\right), \tag{3.65b}$$

and

$$A_\lambda = \sum_{i=0}^{\lambda-1} \left| \frac{1}{\beta_k^i} \right|^2 = \sum_{i=0}^{\lambda-1} r_k^{-2i} = \frac{1 - r_k^{-2\lambda}}{1 - r_k^{-2}}. \tag{3.65c}$$

Equation (3.65a) applies both to the real and the imaginary parts of the upper branch.

Based on (3.63a), (3.65a) and (3.65c), the total noise variance (after the transient time of $\Gamma_k^{(b)}(z)$ has elapsed) is bounded by:

$$\begin{aligned} \left[\sum_{i=0}^{n_k^{(b)}-1} |c_i|^2 \right] (\sigma^2)_k^{(\text{out})}(l) + 2n_k^{(b)}\sigma_e^2 &= \left[\sum_{i=0}^{n_k^{(b)}-1} \left(r_k^{2n_k^{(a)}}\right)^i \right] \frac{4(\sigma^2)_k^{(\text{out})}(l)}{\widehat{\nu}_k^2} + 2n_k^{(b)}\sigma_e^2 \\ &= \frac{1 - r_k^{2n_k^{(a)}n_k^{(b)}}}{1 - r_k^{2n_k^{(a)}}} \frac{4(\sigma^2)_k^{(\text{out})}(l)}{\widehat{\nu}_k^2} + 2n_k^{(b)}\sigma_e^2, \end{aligned} \tag{3.66a}$$

where

$$c_i = \frac{2\beta_k^{in_k^{(a)}}}{\hat{\nu}_k} \tag{3.66b}$$

and

$$|c_i| = \frac{2r_k^{in_k^{(a)}}}{\hat{\nu}_k}. \tag{3.66c}$$

3.4.2 Cascade Structure II

Because of the switching and resetting, the (complex) roundoff error at the output of the upper feedback loop, $\Xi^{(1)}_{k,\,\text{out}}(l)$, is zero at the time instant

$$l = 4\rho n_k^{(a)} - 1, \tag{3.67a}$$

where ρ is an integer, whereas for $0 \leq \lambda \leq 4n_k^{(a)} - 2$,

$$\Xi^{(1)}_{k,\,\text{out}}(4\rho n_k^{(a)} + \lambda) = \sum_{i=0}^{\lambda} (1/\beta_k)^i \Xi^{(1)}_{k,\,\text{in}}(4\rho n_k^{(a)} + \lambda - i). \tag{3.67b}$$

Similarly, the output error at the output of the lower branch is zero at

$$l = (4\rho + 2)n_k^{(a)} - 1, \tag{3.68a}$$

and for $0 \leq \lambda \leq 4n_k^{(a)} - 2$,

$$\Xi^{(2)}_{k,\,\text{out}}((4\rho + 2)n_k^{(a)} + \lambda) = \sum_{i=0}^{\lambda} (1/\beta_k)^i \Xi^{(2)}_{k,\,\text{in}}((4\rho + 2)n_k^{(a)} + \lambda - i). \tag{3.68b}$$

For the structure of Fig.3.12, the overall noise at time l can be expressed as

$$\xi_k^{(\text{ove})}(l) = \sum_{i=0}^{2n_k^{(b)}-2} \frac{2\eta(i)}{\widehat{\nu}_k^{(1)}\widehat{\nu}_k^{(2)}} \,\text{Re}\Bigg\{ \widehat{\gamma}_k \widehat{\nu}_k^{(2)} \Big[\Xi_k^{(1)}(l - in_k^{(a)}, 0, 4n_k^{(a)}) + \Xi_k^{(2)}(l - in_k^{(a)}, 2n_k^{(a)}, 4n_k^{(a)}) \Big] \Bigg\}, \tag{3.69a}$$

where

$$\Xi_k^{(m)}(l, j, n) = \begin{cases} 0, & (l - j) \bmod (n) = n - 1, \\ \displaystyle\sum_{i=0}^{(l-j)\bmod(n)} (1/\beta_k)^i \Xi_{\text{in}}^{(m)}(l - i), & \text{otherwise.} \end{cases} \tag{3.69b}$$

The variance of the complex output noise at the input of the coefficient $\hat{\nu}_k^{(2)}\hat{\gamma}$ at any arbitrary time l can be expressed as

$$(\sigma^{(\,\text{out})}_{k,\,\text{re}})^2(l) = (\sigma^{(1)}_{k,\,\text{re}})^2(l) + (\sigma^{(1)}_{k,\,\text{re}})^2(l - 2n_k^{(a)}) \tag{3.70a}$$

and

$$(\sigma^{(\,\text{out})}_{k,\,\text{im}})^2(l) = (\sigma^{(1)}_{k,\,\text{im}})^2(l) + (\sigma^{(1)}_{k,\,\text{im}})^2(l - 2n_k^{(a)}), \tag{3.70b}$$

where $(\sigma_{k,\,\mathrm{re}}^{(\,\mathrm{out})})^2(l)$ and $(\sigma_{k,\,\mathrm{im}}^{(\,\mathrm{out})})^2(l)$ are the variances of the real and imaginary parts respectively, with

$$(\sigma_{k,\,\mathrm{re}}^{(1)})^2(l) = \begin{cases} 0, & \lambda(l) = 4n_k^{(a)} - 1, \\ \sigma_e^2 \left[\sum_{i=0}^{\lambda(l)} c_1(i) \mathrm{Re}^2(\frac{1}{\beta_k^i}) + \sum_{i=1}^{\lambda(l)} c_2(i) \left(\mathrm{Re}^2(\frac{1}{\beta_k^{i-1}}) + \mathrm{Im}^2(\frac{1}{\beta_k^{i-1}}) \right) \right] \\ = \sigma_e^2 \left[\sum_{i=0}^{\lambda(l)} c_1(i) \mathrm{Re}^2(\frac{1}{\beta_k^i}) + \sum_{i=1}^{\lambda(l)} c_2(i) \left(r_k^{-i+1} \right)^2 \right], & \text{otherwise,} \end{cases} \tag{3.70c}$$

$$(\sigma_{k,\,\mathrm{im}}^{(1)})^2(l) = \begin{cases} 0, & \lambda(l) = 4n_k^{(a)} - 1, \\ \sigma_e^2 \left[\sum_{i=0}^{\lambda(l)} c_1(i) \mathrm{Im}^2(\frac{1}{\beta_k^i}) + \sum_{i=1}^{\lambda(l)} c_2(i) \left(\mathrm{Re}^2(\frac{1}{\beta_k^{i-1}}) + \mathrm{Im}^2(\frac{1}{\beta_k^{i-1}}) \right) \right] \\ = \sigma_e^2 \left[\sum_{i=0}^{\lambda(l)} c_1(i) \mathrm{Im}^2(\frac{1}{\beta_k^i}) + \sum_{i=1}^{\lambda(l)} c_2(i) \left(r_k^{-i+1} \right)^2 \right], & \text{otherwise,} \end{cases} \tag{3.70d}$$

where

$$\lambda(l) = l \mathrm{mod}(4n_k^{(a)}), \tag{3.70e}$$

$$c_1(i) = \begin{cases} 2, & \lambda(l) - 3n_k^{(a)} < i \leq \lambda(l) - n_k^{(a)}, \\ 1, & \text{otherwise,} \end{cases} \tag{3.70f}$$

$$c_2(i) = \begin{cases} 1, & i = \lambda(l), \\ 2, & \text{otherwise.} \end{cases} \tag{3.70g}$$

In the equations (3.70a)–(3.70g), $(\sigma_{k,\,\mathrm{re}}^{(1)})^2(l)$ is the noise variance at the real part of the output of the upper recursive filter, $(\sigma_{k,\,\mathrm{im}}^{(1)})^2(l)$ is the noise variance at the imaginary branch of the output of the upper recursive filter, $(\sigma_{k,\,\mathrm{re}}^{(\mathrm{out})})^2(l)$ is the total noise variance at real branch prior to the input to $\hat{\nu}_k^{(2)}\hat{\gamma}$, $(\sigma_{k,\,\mathrm{im}}^{(\mathrm{out})})^2(l)$ is the total noise variance at imaginary branch prior to the input to $\hat{\nu}_k^{(2)}\hat{\gamma}$, and σ_e^2 is the noise variance generated by each quantizer.

The roundoff noise variance at the input to $\dfrac{2\hat{\Omega}_k^{(b)}(z)}{\hat{\nu}_k^{(1)}\hat{\nu}_k^{(1)}}$ is given by

$$(\sigma_k^{(\text{out})})^2(l) = (\sigma_{k,\,\text{re}}^{(\text{out})})^2(l)\text{Re}^2\{\hat{\nu}_k^{(2)}\ \hat{\gamma}_k\} + (\sigma_{k,\,\text{im}}^{(\text{out})})^2(l)\text{Im}^2\{\hat{\nu}_k^{(2)}\hat{\gamma}_k\} + 2\sigma_e^2. \tag{3.71}$$

Based on (3.70) and (3.71), the total noise variance (after the transient time of $\hat{\Omega}_k^{(b)}(z)$ has elapsed) is bounded by:

$$T_1(\sigma_k^{(\text{out})})^2(l) + (2n_k^{(b)} - 1)\sigma_e^2, \tag{3.72}$$

where

$$T_1 = \sum_{l=0}^{2n_k^{(b)}-2}\left(\frac{2\eta_k(l)}{\hat{\nu}_k^{(1)}\nu_k^{(2)}}\right)^2, \tag{3.73}$$

and $\eta_k(l)$ is given by (3.51d).

3.5 Performance study

This section considers the results obtained from computer simulations for the structures introduced in this chapter. Several cases have been considered, and the results have been compared with some other efficient implementation of linear-phase FIR filters. In what follows, ω_p and ω_s are the desired passband and stopband edges and δ_p and δ_s are the maximum allowable ripples in the pass- and stopband respectively.

The design is based on a procedure which obtains the optimum parameters of the proposed structure. The aforementioned parameters are essentially the coefficients a_k's, the poles (α_k's and β_k)'s and the numerator coefficients γ_k's and κ_k's, which together specify the proposed structure, as illustrated by (3.11). In other words, the coefficients of any linear-phase FIR filter, regardless how large its order is, can be completely specified by a small set of parameters.

Based on (3.14a) and (3.14b), finding γ_k's and β_k's is tantamount to finding the corresponding R_k's, r_k's, ϕ_k's and Φ_k's. In other words, with a suitable starting point, an optimization routine can be performed to yield the optimized parameters a_k's, α_k's, κ_k's, R_k's, r_k's, ϕ_k's and Φ_k's, such that the zero-phase response of the resulting filter, as given by (3.23) and determined by (3.11b), (3.13b), (3.15a) and (3.26) (or equally well by the corresponding impulse responses (3.17b), (3.17c), (3.17d) and (3.28)) meets the specifications. In the simulations presented below, the Matlab routine `fminimax`[3] has been employed to yield the optimum solution.

3.5.1 The Prototype Filter Specifications

In this section we derive the specifications for a prototype IIR filter, which is used as the starting point in the optimization routine of Section 3.5.2.

Consider the tolerance scheme for the zero-phase response of the desired linear-phase FIR filter $H(z)$, presented as

$$1 - \delta_p \leq H(\omega) \leq 1 + \delta_p, \quad 0 \leq \omega \leq \omega_p, \tag{3.74a}$$

$$-\delta_s \leq H(\omega) \leq \delta_s, \quad \omega_s \leq \omega \leq 1. \tag{3.74b}$$

Since for any IIR filter $G(z)$, the amplitude response of $|G(e^{j\omega})G(e^{-j\omega})|$ is non-negative both in the passband and stopband regions, it follows that the zero-phase response of the desired linear-phase FIR filter should first be raised by δ_s to give

$$1 - \delta_p + \delta_s \leq H(\omega) + \delta_s \leq 1 + \delta_p + \delta_s, \quad 0 \leq \omega \leq \omega_p, \tag{3.75a}$$

$$0 \leq H(\omega) + \delta_s \leq 2\delta_s, \quad \omega_s \leq \omega \leq \pi. \tag{3.75b}$$

[3]The function `fminimax` uses a Sequential Quadratic Programming (SQP) method for minimizing the maximum value of a set of multivariable functions subject to linear and nonlinear constraints.

Traditionally IIR filters are designed such that the maximum amplitude response in the passband is equal to 1, therefore the amplitude response of (3.75) should be normalized to yield

$$\frac{1-\delta_p+\delta_s}{1+\delta_p+\delta_s} \leq \frac{H(\omega)+\delta_s}{1+\delta_p+\delta_s} \leq 1, \quad 0 \leq \omega \leq \omega_p, \tag{3.76a}$$

$$0 \leq \frac{H(\omega)+\delta_s}{1+\delta_p+\delta_s} \leq \frac{2\delta_s}{1+\delta_p+\delta_s}, \quad \omega_s \leq \omega \leq \pi. \tag{3.76b}$$

$(H(\omega)+\delta_s)/(1+\delta_p+\delta_s)$ can be assumed to be the amplitude response of cascade of a prototype IIR filter $G(z)$ with its unstable counterpart $G(z^{-1})$, since according to (3.76), it is non-negative for all frequencies. That is why

$$|G(e^{j\omega})G(e^{-j\omega})| = \frac{H(\omega)+\delta_s}{1+\delta_p+\delta_s} \tag{3.77}$$

is justified.

Based on (3.76) and (3.77), the amplitude response of the prototype IIR filter $G(z)$ can be written as:

$$\sqrt{\frac{1-\delta_p+\delta_s}{1+\delta_p+\delta_s}} \leq |G(e^{j\omega})| \leq 1, \quad 0 \leq \omega \leq \omega_p, \tag{3.78a}$$

$$|G(e^{j\omega})| \leq \sqrt{\frac{2\delta_s}{1+\delta_p+\delta_s}}, \quad \omega_s \leq \omega \leq \pi. \tag{3.78b}$$

3.5.2 Simulation Results

Example 1: Consider the specifications $\omega_p = 0.4\pi$, $\omega_s = 0.41\pi$, $\delta_p = 0.01$ and $\delta_s = 0.001$. The minimum order required by an optimum linear-phase FIR filter designed by Remez algorithm to meet the above specifications is 513. This filter needs 513 delay elements and 513 adders to meet the specifications. The required number of multipliers (utilizing the symmetry of the coefficients) is 257.

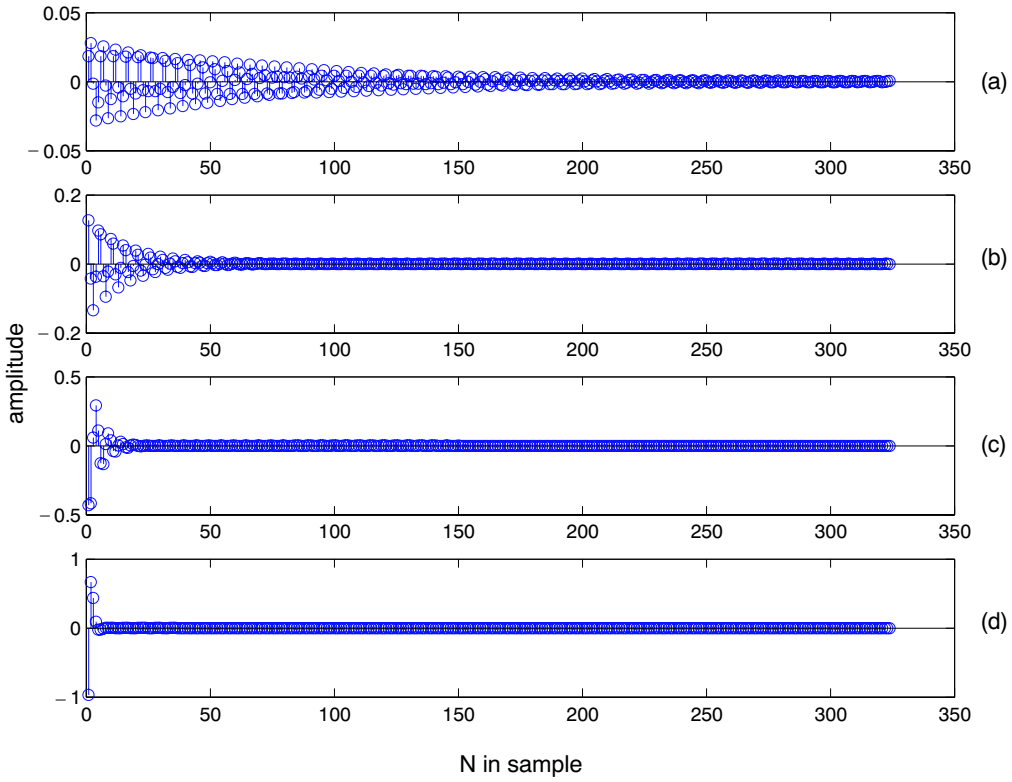

Figure 3.15: The impulse responses for the optimized Cascade Structure I for the specifications of *Example* 1. (a) First branch (b) Second branch (c) Third branch (d) Fourth branch

Cascade Structure I requires five branches to meet the specifications, where one of the branches corresponds to $A(z)$ as given by (3.11b) with a single coefficient a_0, and four others correspond to $E_k^{(c)}(z), k = 1, 2, 3, 4$ as given by (3.13c). Applying a truncation length of $N_k = 324, k = 1, 2, 3, 4$ leads to a filter structure meeting the specifications. The impulse responses of these four branches of the resulting structure is presented in Fig. 3.15. By decomposing N_k into $n_k^{(a)} = 162$ and $n_k^{(b)} = 2$, and considering the results given on Table 3.4 with recursive implementation of $\Gamma_k^{(a)}(z)$, the required delay elements,

real-valued additions and real-valued multiplications for each $E_k^{(c)}(z)$ are 654, 15, and 21 respectively.

A more careful inspection of the parameters of the optimum structure reveals that $|\beta_1| \approx 0.9866, |\beta_2| \approx 0.9353, |\beta_3| \approx 0.7926$ and $|\beta_4| \approx 0.3816$. This implies that the impulse responses of the branches corresponding to β_3 and β_4 decays quite rapidly, and therefore, some of the samples of these branches can be discarded. In fact the structure meets the specifications, if the largest 136 samples of the second branch, the largest 42 samples of the third branch and the the largest 12 samples of the fourth branch are preserved. As a result, the third and the fourth branches can essentially be implemented as FIR filters of order 42 and 12 respectively.

The above observation justifies a second round of optimization, in order to find the minimum truncation length N_k for each branch separately. Doing so, it turns out that the specifications of this example is met with $N_1 = 324$, $N_2 = 88$, $N_3 = 24$ and $N_4 = 10$. Based on the results given in Table 3.4, with $n_1^{(a)} = 162, n_2^{(a)} = 44$ and $n_1^{(b)} = n_2^{(b)} = 2$, the whole structure can be implemented using 78 real-valued additions, 90 real-valued multiplications and 654 delay elements respectively.

The number of real-valued additions and real-valued multiplications are almost 15% and 35% that of the optimum direct form design, while the number of delays has increased by less than 27%. The zero-phase response of the proposed structure is presented in Fig. 3.16.

By the above setting and based on the results of Section 3.4.1, the maximum variance of the noise generated by the first and the second branches[4] are approximately $2.7 \times 10^6 \sigma_e^2$ and $8.7 \times 10^6 \sigma_e^2$ respectively. This implies that by assigning 12 extra bits, the variance of

[4]As indicated above, the other branches are implemented as FIR filters, and therefore, their roundoff noise can be ignored.

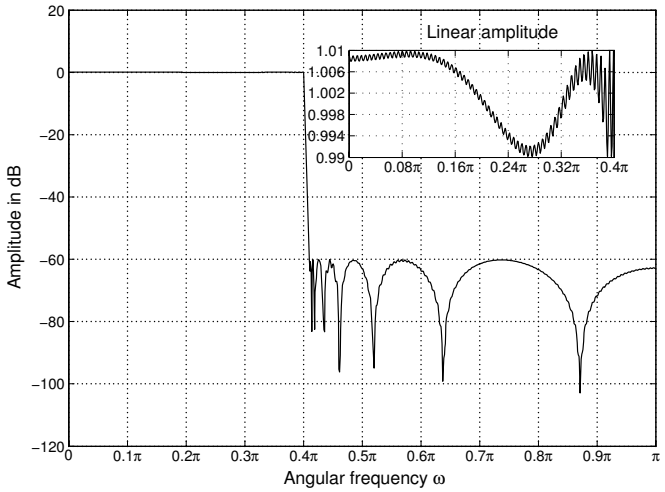

Figure 3.16: Amplitude response of the Cascade Structure I, meeting the specifications of *Example* 1.

the noise can be lowered to that of a single quantizer.

To meet the above specifications, Cascade Structure II requires five branches, where one of the branches corresponds to $A(z)$ as given by (3.11b) with a single coefficient a_0, and four others correspond to $E_k^{(c)}(z), k = 1, 2, 3, 4$ as given by (3.26). Applying a truncation length of $N_k = 172, k = 1, 2, 3, 4$ leads to a filter structure meeting the specifications. The impulse responses of the four branches of the resulting structure is presented in Fig. 3.17. By decomposing N_k into $n_k^{(a)} = 86$ and $n_k^{(b)} = 2$, and considering the results given on Table 3.5 with recursive implementation of $\Gamma_k^{(a)}(z)$, the required delay elements, real-valued additions and real-valued multiplications for each $E_k^{(c)}(z)$ are 694, 16, and 21 respectively.

For the optimum structure, $|\beta_1| \approx 0.9871, |\beta_2| \approx 0.9431, |\beta_3| \approx 0.7954$ and $|\beta_4| \approx 0.3755$. As for the Cascade Structure I, the impulse response of β_3 and β_4 decay rapidly, and therefore the specifications of this example can still be met with $N_1 = 172$, $N_2 = 51$, $N_3 = 17$ and $N_4 = 5$. Based on the results given in Table 3.5, with $n_1^{(a)} = 86, n_2^{(a)} = 17, n_1^{(b)} = 2$

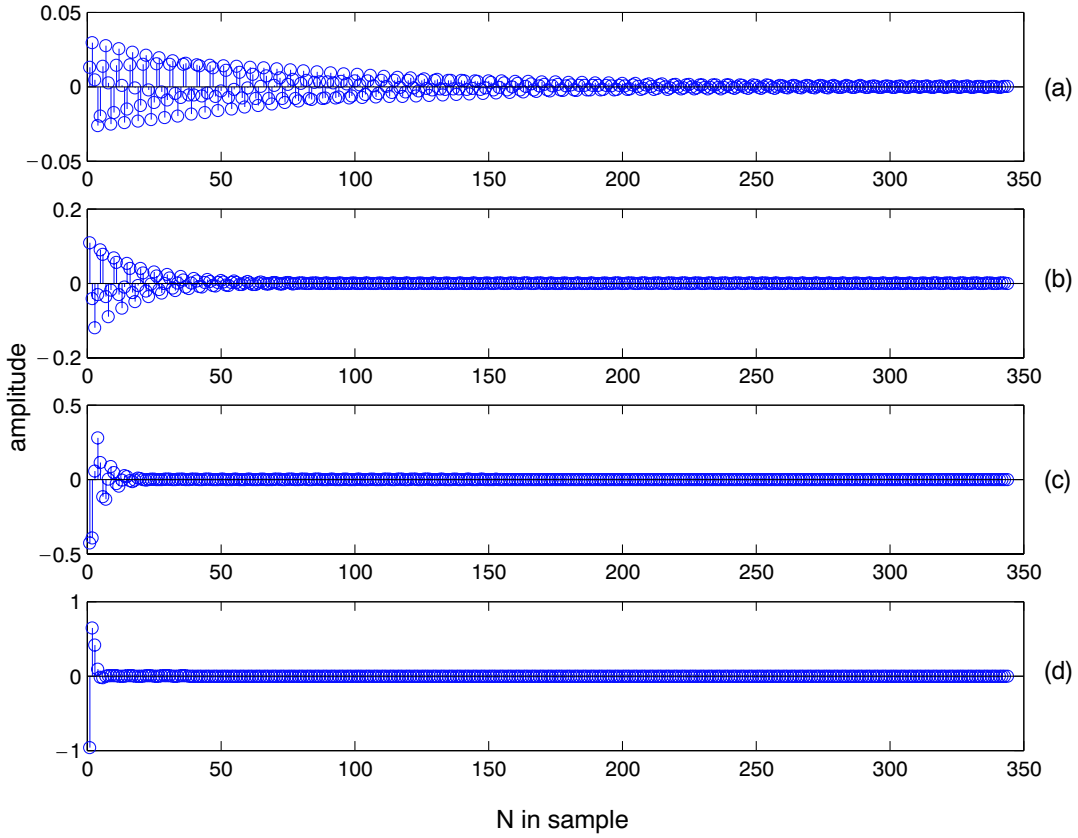

Figure 3.17: The impulse responses for the optimized Cascade Structure II for the specifications of *Example* 1. (a) First branch. (b) Second branch. (c) Third branch. (d) Fourth branch.

and $n_2^{(b)} = 3$, the whole structure can be implemented with the recursive implementation of the first and the second branches, and nonrecursive implementation of the third and the fourth branches, using 66 real-valued additions, 77 real-valued multiplications and 694 delay elements respectively.

The proposed structure shows significant savings compared to other efficient implementations of linear-phase FIR filters. For instance, to implement the above structure using the frequency masking response technique [62, 63], a total of 154 real-valued additions, 79

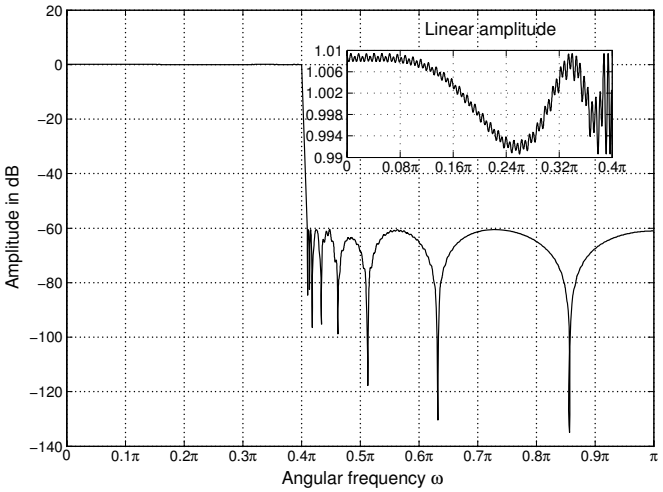

Figure 3.18: Amplitude response of the Cascade Structure II, meeting the specifications of *Example* 1.

real-valued multiplications and 564 delay elements is required. This implies that the proposed design improves the number of real-valued additions and real-valued multiplications by 57% and 3%, at the expense of 23% extra delay. The amplitude response of the proposed structure is presented in Fig. 3.18.

By the above setting and based on the results of Section 3.4.2, the maximum variance of the noise generated by the first and the second branches are $3.54 \times 10^5 \sigma_e^2$ and $1.64 \times 10^3 \sigma_e^2$ respectively. This implies that by assigning 10 extra bits, the variance of the noise can be lowered to that of a single quantizer.

Example 2: Consider the specifications: $\omega_p = 0.4\pi$, $\omega_s = 0.402\pi$, $\delta_p = 0.01$ and $\delta_s = 0.001$. The minimum order required by a linear-phase FIR filter designed by Remez algorithm to meet the above specifications is 2541 [92], implying that the filter requires 2541 delay elements, 2541 adders and 1271 multipliers to meet the specifications.

Cascade Structure I requires six branches to meet the specifications, where one of the

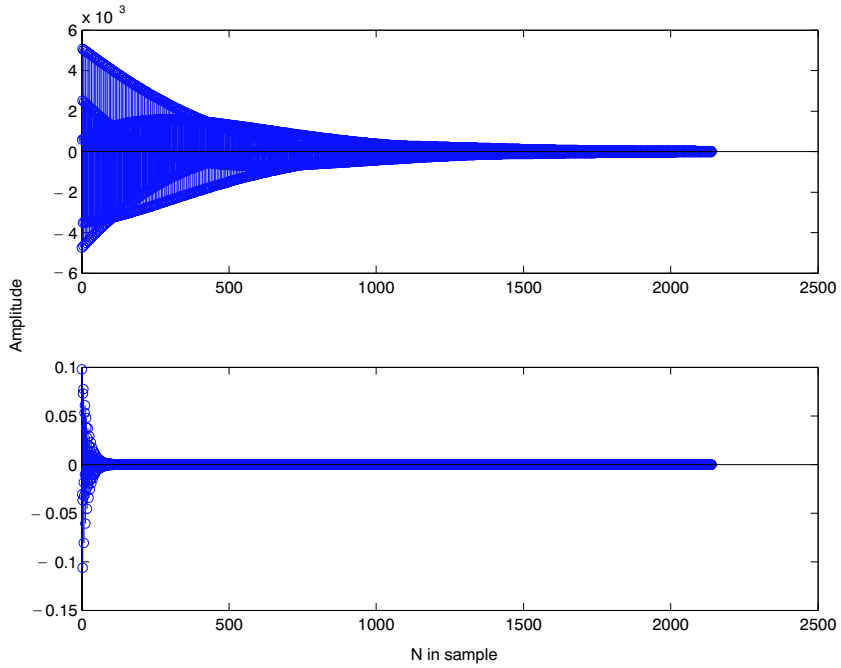

Figure 3.19: The impulse responses for the optimized Cascade Structure I for the specifications of *Example* 2. (a) First branch. (c) Third branch.

branches corresponds to $A(z)$ as given by (3.11b) with a single coefficient a_0, and five others correspond to $E_k^{(c)}(z), k = 1, 2, \ldots, 5$ as given by (3.13c). Applying a truncation length of $N_k = 2139, k = 1, 2, \ldots, 5$ leads to a filter structure meeting the specifications. The impulse responses of the first and the third branches of the resulting structure is presented in Fig. 3.19. By decomposing N_k into $n_k^{(a)} = 713$ and $n_k^{(b)} = 3$, and considering the results given on Table 3.4 with recursive implementation of $\Gamma_k^{(a)}(z)$, the required delay elements, real-valued additions and real-valued multiplications for each $E_k^{(c)}(z)$ are 4284, 17, and 23 respectively.

A more careful inspection of the parameters of the optimum structure reveals that $|\beta_1| \approx 0.9975, |\beta_2| \approx 0.9868, |\beta_3| \approx 0.9471$ and $|\beta_4| \approx 0.8032$ and $|\beta_5| \approx 0.4291$. This implies that the impulse responses of the branches corresponding to β_4 and β_5 decays quite rapidly,

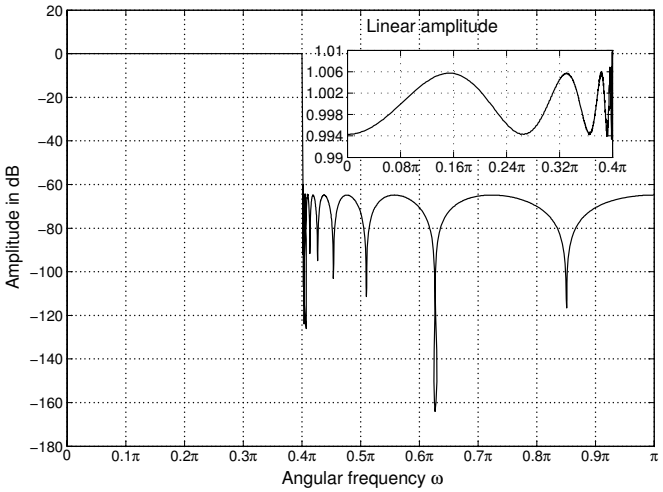

Figure 3.20: Amplitude response of the Cascade Structure I, meeting the specifications of *Example* 2.

and therefore, some of the samples of these branches can be discarded. In fact the structure meets the specifications, if the largest 417 samples of the second branch, the largest 135 samples of the third branch, the largest 38 samples of the fourth branch and the the largest 14 samples of the fifth branch are preserved. As a result, the fourth and the fifth branches can essentially be implemented as FIR filters of order 37 and 12 respectively.

The above observation justifies a second round of optimization, in order to find the minimum truncation length N_k for each branch separately. Doing so, it turns out that the specifications of this example is met with $N_1 = 2139$, $N_2 = 376$, $N_3 = 119$, $N_4 = 30$ and $N_5 = 11$. Based on the results given in Table 3.4, the whole structure can be implemented using 81 real-valued additions, 99 real-valued multiplications and 4284 delay elements respectively.

The number of real-valued additions and real-valued multiplications are almost 3% and 8% that of the optimum direct form design, while the number of delays has increased by

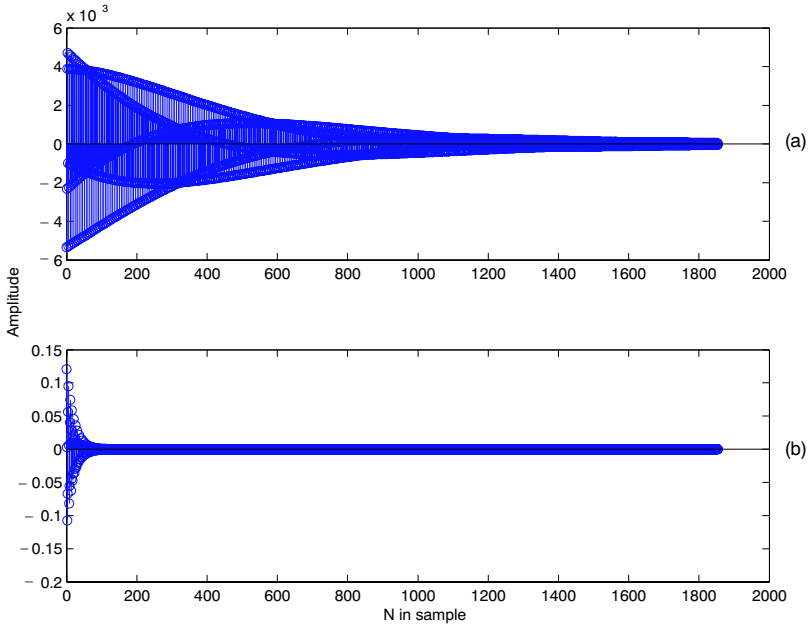

Figure 3.21: The impulse responses for the optimized Cascade Structure II for the specifications of *Example* 2. (a) First branch. (b) Third branch.

less than 69%. The amplitude response of the proposed structure is presented in Fig. 3.20.

The proposed structure shows considerable savings compared to other efficient implementation of linear-phase FIR filters. For instance, to implement the above structure using the frequency masking response technique [62, 63] a total of 332 real-valued additions, 168 real-valued multiplications and 2690 delay elements is required. This implies that the proposed design improves the number of real-valued additions and real-valued multiplications by 75% and 41%, at the expense of 59% extra delay.

By the above setting and based on the results of Section 3.4.1, the maximum variance of the noise generated by the first and the second branches are $2.04 \times 10^8 \sigma_e^2$ and $5.83 \times 10^{10} \sigma_e^2$ respectively. This implies that by assigning 18 extra bits, the variance of the noise can be lowered to that of a single quantizer.

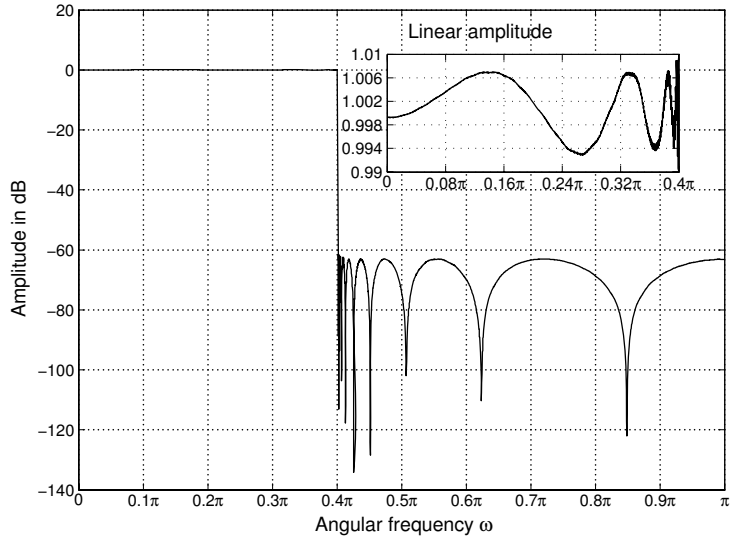

Figure 3.22: Amplitude response of the Cascade Structure II, meeting the specifications of *Example* 2.

To meet the above specifications, Cascade Structure II requires six branches, where one of the branches corresponds to $A(z)$ as given by (3.11b) with a single coefficient a_0, and five others correspond to $E_k^{(c)}(z), k = 1, 2, \ldots, 5$ as given by (3.26). Applying a truncation length of $N_k = 927, k = 1, 2, \ldots, 5$ leads to a filter structure meeting the specifications. The impulse responses of the first and the third branches of the resulting structure is presented in Fig. 3.21. By decomposing N_k into $n_k^{(a)} = 309$ and $n_k^{(b)} = 3$, and considering the results given on Table 3.5 with recursive implementation of $\Gamma_k^{(a)}(z)$, the required delay elements, real-valued additions and real-valued multiplications for each $E_k^{(c)}(z)$ are 3714, 18, and 23 respectively.

For the optimum structure, $|\beta_1| \approx 0.9976, |\beta_2| \approx 0.9875, |\beta_3| \approx 0.9500, |\beta_4| \approx 0.8082$ and $|\beta_5| \approx 0.4225$. As for the Cascade Structure I, the impulse response of β_4 and β_5 decay rapidly, and therefore the specifications of this example can still be met with $N_1 = 927$, $N_2 = 476$, $N_3 = 97$, $N_4 = 38$ and $N_5 = 9$.

Table 3.7: Summary of the Crucial Parameters of the Re-optimized Cascade Structures I and II for *Example 1* and *Example 2*

	Cascade Structure I	Cascade Structure II
Example 1	$\beta_1 = 0.2939 - 0.9417i, \gamma_1 = 0.0120 + 0.0091i$, $\beta_2 = 0.2765 - 0.8926i, \gamma_2 = 0.0524 - 0.0485i$ $\beta_3 = 0.2465 - 0.7576i, \gamma_3 = -0.2051 - 0.1798i$, $\beta_4 = 0.2240 - 0.3348i, \gamma_4 = -0.3832 + 1.2375i$, $a_0 = 1.1581$	$\beta_1 = 0.2929 - 0.9426i, \gamma_1 = 0.0065 + 0.0137i$, $\beta_2 = 0.2820 - 0.9000i, \gamma_2 = 0.0544 - 0.0400i$ $\beta_3 = 0.2694 - 0.7484i, \gamma_3 = -0.2131 - 0.1865i$, $\beta_4 = 0.2195 - 0.3047i, \gamma_4 = -0.4798 + 1.3992i$, $a_0 = 1.3615$
Example 2	$\beta_1 = 0.3063 - 0.9492i, \gamma_1 = -0.0025 + 0.0013i$, $\beta_2 = 0.3044 - 0.9379i, \gamma_2 = 0.0071 + 0.0123i$ $\beta_3 = 0.2993 - 0.8982i, \gamma_3 = 0.0519 - 0.0344i$, $\beta_4 = 0.2882 - 0.7518i, \gamma_4 = -0.2017 - 0.1811i$, $\beta_5 = 0.2784 - 0.3288i, \gamma_5 = -0.1629 + 1.0811i$, $a_0 = 0.7250$	$\beta_1 = 0.3060 - 0.9495i, \gamma_1 = -0.0027 - 0.0004i$, $\beta_2 = 0.3046 - 0.9396i, \gamma_2 = 0.0008 + 0.0134i$ $\beta_3 = 0.3024 - 0.9016i, \gamma_3 = 0.0600 - 0.0189i$, $\beta_4 = 0.2963 - 0.7561i, \gamma_4 = -0.1524 - 0.2121i$, $\beta_5 = 0.2853 - 0.3297i, \gamma_5 = -0.3633 + 1.2074i$, $a_0 = 0.9921$

Based on the results given in Table 3.5 with the recursive implementation of the first, the second, and the third branches and nonrecursive implementation of the fourth and the fifth branches, the whole structure can now be implemented using 88 real-valued additions, 103 real-valued multiplications and 3714 delay elements respectively. The amplitude response of the proposed structure is presented in Fig. 3.22.

By the above setting and based on the results of Section 3.4.2, the maximum variance of the noise generated by the first, second and third branches are $1.11 \times 10^4 \sigma_e^2$, $3.21 \times 10^8 \sigma_e^2$ and $8.58 \times 10^8 \sigma_e^2$ respectively. This implies that by assigning 15 extra bits, the variance of the noise can be lowered to that of a single quantizer. Table 3.7 presents the crucial parameters of the structures proposed in this chapter for the above examples.

Chapter 4

Parallel Structures for Generating Sharp Linear-Phase FIR Filters

In addition to the cascade structures introduced in Chapter 3, the principle of switching and resetting can be employed in a different setting to yield alternative, one-block structures. The block being considered is composed of a parallel connection of FIR filters, implemented as IIR filters.

In this chapter, three transfer functions exploiting the proposed parallel structures are derived, and their implementational aspects are discussed. In addition, the roundoff noise effects in the proposed structures are analyzed in detail, and simulation results are provided to illustrate the efficiency of the design.

4.1 Alternative Structures

Consider the transfer function $G(z)$ as given by (3.10) in Section 3.2. Multiplying $G(z)$ by $G(z^{-1})$ yields the following zero-phase IIR transfer function:

$$G(z)G(z^{-1}) = \widehat{g}\ \frac{\prod_{k=1}^{L}[(z+z^{-1}) - (q_k + q_k^{-1})]}{\prod_{k=1}^{K}[(z+z^{-1}) - (p_k + p_k^{-1})]}, \tag{4.1a}$$

where

$$\widehat{g} = g^2 \prod_{k=1}^{L}(-q_k) \Big/ \prod_{k=1}^{K}(-p_k). \tag{4.1b}$$

Since $G(z)G(z^{-1})$ is a ratio of two polynomials in $z+z^{-1}$, the partial fraction expansion in terms of $z + z^{-1}$ enables one to express $G(z)G(z^{-1})$ in the following form:

$$G(z)G(z^{-1}) = A(z) + \sum_{k=1}^{K_R} R_k(z) + \sum_{k=1}^{K_C} C_k(z), \tag{4.2a}$$

where

$$A(z) = \sum_{k=0}^{L-K} A_k (z + z^{-1})^k, \tag{4.2b}$$

$$R_k(z) = \frac{\widehat{\kappa}_k}{(z+z^{-1}) - (\alpha_k + \alpha_k^{-1})}, \tag{4.2c}$$

and

$$\begin{aligned} C_k(z) &= \frac{\widehat{\gamma}_k}{(z+z^{-1}) - (\beta_k + \beta_k^{-1})} + \frac{\widehat{\gamma}_k^*}{(z+z^{-1}) - (\beta_k^* + \beta_k^{*-1})} \\ &= 2\Re\Big\{ \frac{\widehat{\gamma}_k}{(z+z^{-1}) - (\beta_k + \beta_k^{-1})} \Big\}, \end{aligned} \tag{4.2d}$$

with

$$\widehat{\kappa}_k = [(z+z^{-1}) - (\alpha_k + 1/\alpha_k)] G(z)G(z^{-1}) \Big|_{(z+z^{-1}) = (\alpha_k + \alpha_k^{-1})}, \tag{4.2e}$$

and

$$\widehat{\gamma}_k = [(z + z^{-1}) - (\beta_k + 1/\beta_k)]G(z)G(z^{-1})\Big|_{(z+z^{-1})=(\beta_k+\beta_k^{-1})}. \tag{4.2f}$$

As in Section 3.2, $\Re\{H(z)\}$ denotes the transfer function that is the average of the transfer function $H(z)$ and the one obtained by replacing the coefficients of $H(z)$ by their complex conjugates. Furthermore, as in Section 3.2, it is assumed that there are no repeated poles in the above partial fraction expansion. In addition, the first term on the right-hand side of (4.2a) is absent if $L < K$.

Alternatively, $A(z)$, the $R_k(z)$'s, and the $C_k(z)$'s can be expressed as

$$A(z) = a_0 + \sum_{k=1}^{L-K} a_k(z^k + z^{-k}), \tag{4.3a}$$

$$R_k(z) = \frac{\kappa_k}{(1 - \alpha_k z^{-1})(1 - \alpha_k z)}, \tag{4.3b}$$

and

$$C_k(z) = 2\Re\Big\{\Gamma_k(z)\Big\}, \tag{4.3c}$$

with

$$\Gamma_k(z) = \frac{\gamma_k}{(1 - \beta_k z^{-1})(1 - \beta_k z)}. \tag{4.3d}$$

Here,

$$a_k = \begin{cases} \displaystyle\sum_{\substack{r=k \\ r \text{ even}}}^{L-K} \binom{r}{(r-k)/2} A_r, & \text{for } k \text{ even}, \\ \displaystyle\sum_{\substack{r=k \\ r \text{ odd}}}^{L-K} \binom{r}{(r-k)/2} A_r, & \text{for } k \text{ odd}, \end{cases} \tag{4.4a}$$

$$\kappa_k = -\alpha_k \widehat{\kappa}_k, \tag{4.4b}$$

and

$$\gamma_k = -\beta_k \widehat{\gamma}_k. \tag{4.4c}$$

Using the notations $\beta_k = r_k e^{j\phi_k}$ and $\gamma_k = R_k e^{j\Phi_k}$, the impulse response of $G(z)G(z^{-1})$ is expressible as

$$g(n) = g^{(a)}(n) + \sum_{k=1}^{K_R} g_k^{(r)}(n) + \sum_{k=1}^{K_C} g_k^{(c)}(n), \tag{4.5a}$$

where

$$g_a(n) = \begin{cases} a_n, & |n| \le L-K, \\ 0, & \text{otherwise}, \end{cases} \tag{4.5b}$$

$$g_k^{(r)}(n) = \frac{\kappa_k \alpha_k^{|n|}}{1-\alpha_k^2}, \tag{4.5c}$$

and

$$\begin{aligned} g_k^{(c)}(n) =& 2\,\mathrm{Re}\left\{ \frac{\gamma_k \beta_k^{|n|}}{1-\beta_k^2} \right\} \\ =& \frac{2R_k\left[r_k^{|n|}\cos(\Phi_k + |n|\phi_k) - r_k^{|n|+2}\cos(\Phi_k + (|n|-2)\phi_k)\right]}{1-2r_k^2\cos(2\phi_k)+r_k^4}. \end{aligned} \tag{4.5d}$$

As in designing FIR filters by windowing, truncating and smoothing of (4.5a) yields a zero-phase FIR transfer function, whose frequency response approximates $|G(e^{j\omega})|^2$. In what follows, three approximating zero-phase FIR filter transfer function alternatives of the form

$$F^{(0)}(z) = A(z) + \sum_{k=1}^{K_R} \widehat{R}_k(z) + \sum_{k=1}^{K_C} \widehat{C}_k(z), \tag{4.6}$$

will be proposed, giving rise to three filter structures denoted by Parallel Structure I, Parallel Structure II, and Parallel Structure III. In (4.6), the $\widehat{R}_k(z)$'s and $\widehat{C}_k(z)$'s are finite-duration approximations to the $R_k(z)$'s and $C_k(z)$'s, respectively.

4.1.1 Parallel Structure I

The transfer function for an FIR filter having the following truncated impulse response

$$\widehat{g}_k^{(r)}(n) = \begin{cases} g_k^{(r)}(n), & |n| \leq M_k - 1, \\ 0, & \text{otherwise}, \end{cases} \tag{4.7}$$

can be expressed as

$$\widehat{R}_k(z) = \frac{\kappa_k}{(1-\alpha_k z^{-1})(1-\alpha_k z)}\Big[1 + \frac{\alpha_k^{M_k+1}}{1-\alpha_k^2}[z^{M_k-1} + z^{-(M_k-1)}] - \frac{\alpha_k^{M_k}}{1-\alpha_k^2}[z^{M_k} + z^{-M_k}]\Big]. \tag{4.8}$$

Similarly, the transfer function for a filter having the impulse response

$$\widehat{g}_k^{(c)}(n) = \begin{cases} g_k^{(c)}(n), & |n| \leq N_k - 1, \\ 0, & \text{otherwise}, \end{cases} \tag{4.9}$$

is

$$\widehat{C}_k(z) = 2\Re\Big\{\widehat{\Gamma}_k(z)\Big\}, \tag{4.10a}$$

where

$$\widehat{\Gamma}_k(z) = \frac{\gamma_k}{(1-\beta_k z^{-1})(1-\beta_k z)}\Big[1 + \frac{\beta_k^{N_k+1}}{1-\beta_k^2}[z^{N_k-1} + z^{-(N_k-1)}] - \frac{\beta_k^{N_k}}{1-\beta_k^2}[z^{N_k} + z^{-N_k}]\Big]. \tag{4.10b}$$

4.1.2 Parallel Structure II

Using the approximation

$$\frac{1}{(1-x)} = \sum_{k=0}^{\infty} x^k \approx \sum_{k=0}^{N-1} x^k = \frac{1-x^N}{1-x}, \tag{4.11}$$

we obtain

$$\frac{1}{1-\beta z^{-1}} \approx \frac{1-\beta^N z^{-N}}{1-\beta z^{-1}} \tag{4.12a}$$

and

$$\frac{1}{1-\beta z} \approx \frac{1-\beta^N z^N}{1-\beta z}. \tag{4.12b}$$

Applying (4.12a) and (4.12b) with $N = M_k$ to $R_k(z)$, as given by (4.3b), gives

$$\widehat{R}_k(z) = \kappa_k \frac{(1-\alpha_k^{M_k} z^{-M_k})(1-\alpha_k^{M_k} z^{M_k})}{(1-\alpha_k z^{-1})(1-\alpha_k z)}. \tag{4.13}$$

Similarly, applying the above approximations with $N = N_k$ to $C_k(z)$, as given by (4.3c) and (4.3d), yields

$$\widehat{C}_k(z) = 2\Re\Big\{\widehat{\Gamma}_k(z)\Big\}, \tag{4.14a}$$

where

$$\widehat{\Gamma}_k(z) = \gamma_k \frac{(1-\beta_k^{N_k} z^{-N_k})(1-\beta_k^{N_k} z^{N_k})}{(1-\beta_k z^{-1})(1-\beta_k z)}. \tag{4.14b}$$

The impulse responses of the resulting $\widehat{R}_k(z)$ and $\widehat{C}_k(z)$ are given by

$$\widehat{g}_k^{(r)}(n) = \begin{cases} \dfrac{\kappa_k \alpha_k^{|n|}(1-(\alpha_k^2)^{M_k-|n|})}{1-\alpha_k^2}, & |n| \le M_k - 1, \\ 0, & \text{otherwise}, \end{cases} \tag{4.15}$$

and

$$\widehat{g}_k^{(c)}(n) = \begin{cases} 2\ \mathrm{Re}\left\{ \dfrac{\gamma_k \beta_k^{|n|}(1-(\beta_k^2)^{N_k-|n|})}{1-\beta_k^2} \right\}, & |n| \le N_k - 1, \\ 0, & \text{otherwise}, \end{cases} \tag{4.16}$$

respectively.

4.1.3 Parallel Structure III

$C_k(z)$, as given by (4.3c), can also be written as

$$C_k(z) = \frac{\eta_0 + \eta_1(z+z^{-1})}{(1-\beta_k z^{-1})(1-\beta_k^* z^{-1})(1-\beta_k z)(1-\beta_k^* z)}, \tag{4.17a}$$

where

$$\eta_0 = 2\ \mathrm{Re}\{\gamma_k\left(1+\beta_k^{*2}\right)\}, \tag{4.17b}$$

and

$$\eta_1 = -2\,\mathrm{Re}\{\gamma_k \beta_k^*\}. \tag{4.17c}$$

An approximation to $C_k(z)$ is obtained by applying (4.12a) and (4.12b) to (4.17a), yielding

$$\widehat{C}_k(z) = \frac{\Big(\eta_0 + \eta_1(z + z^{-1})\Big)\widehat{\Omega}_k(z)}{(1-\beta_k z^{-1})(1-\beta_k^* z^{-1})(1-\beta_k z)(1-\beta_k^* z)}, \tag{4.18a}$$

where

$$\widehat{\Omega}_k(z) = (1-\beta_k^{N_k} z^{-N_k})(1-(\beta_k^*)^{N_k} z^{-N_k})(1-\beta_k^{N_k} z^{N_k})(1-(\beta_k^*)^{N_k} z^{N_k}). \tag{4.18b}$$

Alternatively, $\widehat{C}_k(z)$ can be expressed as

$$\widehat{C}_k(z) = 2\widehat{\Omega}_k(z)\Re\Big\{\frac{\gamma_k}{(1-\beta_k z^{-1})(1-\beta_k z)}\Big\}. \tag{4.19}$$

The impulse response of $\widehat{C}_k(z)$ is given by

$$\widehat{g}_k^{(c)}(n) = \begin{cases} 2\,\mathrm{Re}\left\{\dfrac{\gamma_k}{1-\beta_k^2}\left((1+\beta_k^{*2N_k})(\beta_k^{|n|}-\beta_k^{2N_k-|n|}) + \beta_k^{*N_k}(\beta_k^{N_k+n}-\beta_k^{N_k-|n|})\right)\right\}, & \\ \qquad\qquad |n| \le N_k - 1, & \\ 2\,\mathrm{Re}\left\{\dfrac{\gamma_k \beta_k^{N_k}\beta_k^{*N_k}(\beta_k^{2N_k-|n|}-\beta_k^{|n|-2N_k})}{1-\beta_k^2}\right\}, & N_k \le |n| \le 2N_k - 1, \\ 0 & \text{otherwise.} \end{cases} \tag{4.20}$$

4.2 Filter Implementation

This section presents efficient implementations for the transfer functions presented in Section 4.1. Filters corresponding to Parallel Structure I are not treated, since in this case we have not found any implementation form for reasonably reducing the output roundoff error noise.

4.2.1 Parallel Structure II

Multiplying $F^{(0)}(z)$ in (4.6) by z^{-D} results in a realizable transfer function for Parallel Structure II, where

$$D = \max\{M_1 - 1, M_2 - 1, \cdots, M_{K_R} - 1, N_1 - 1, N_2 - 1, \cdots, N_{K_C} - 1\} \tag{4.21}$$

is half the order of the resulting filter. This yields

$$H(z) = z^{-(D-(L-K))}A(z) + \sum_{k=1}^{K_R} z^{-(D+1-M_k)}H^{(r)}(z) + \sum_{k=1}^{K_C} z^{-(D+1-N_k)}H^{(c)}(z), \tag{4.22a}$$

where

$$A(z) = a_0 z^{-(L-K)} + \sum_{k=1}^{L-K} a_k[z^{-(L-K+k)} + z^{-(L-K-k)}], \tag{4.22b}$$

$$H_k^{(r)}(z) = \kappa_k(\alpha_k)^{(M_k-1)}\frac{1 - [(\alpha_k)^{M_k} + (1/\alpha_k)^{M_k}]z^{-M_k} + z^{-2M_k}}{1 - [\alpha_k + 1/\alpha_k]z^{-1} + z^{-2}}, \tag{4.22c}$$

and

$$H_k^{(c)}(z) = 2\Re\Big\{\Gamma_k(z)\Big\}, \tag{4.22d}$$

with

$$\Gamma_k(z) = \gamma_k(\beta_k)^{(N_k-1)}\frac{1 - [(\beta_k)^{N_k} + (1/\beta_k)^{N_k}]z^{-N_k} + z^{-2N_k}}{1 - [\beta_k + 1/\beta_k]z^{-1} + z^{-2}}. \tag{4.22e}$$

An implementation for the overall filter is depicted in Fig. 4.1.

The output noise variance due to the multiplication roundoff errors occurring in the feedback loops of $H_k^{(r)}(z)$ and $H_k^{(c)}(z)$ can be significantly reduced by exploiting the following identity:

$$1 - x^{m_k^{(a)}m_k^{(b)}} = (1 - x^{m_k^{(a)}}) \sum_{l=0}^{m_k^{(b)}-1} [x^{m_k^{(a)}}]^l. \tag{4.23}$$

Based on this, $H_k^{(r)}(z)$ for

$$M_k = m_k^{(a)}m_k^{(b)} \tag{4.24}$$

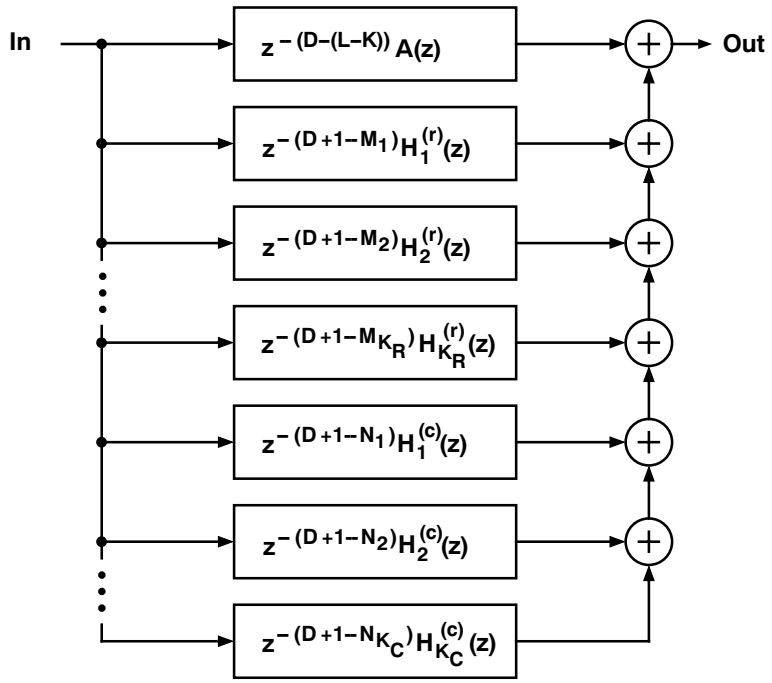

Figure 4.1: Structure for a linear-phase FIR filter obtained by using Parallel Structure II.

can be expressed as

$$H_k^{(r)}(z) = \Upsilon_k^{(a)}(z)\Upsilon_k^{(b)}(z), \tag{4.25a}$$

where

$$\Upsilon_k^{(a)}(z) = \kappa_k(\alpha_k)^{(M_k-1)}\frac{1-[(\alpha_k)^{m_k^{(a)}}+(1/\alpha_k)^{m_k^{(a)}}]z^{-m_k^{(a)}}+z^{-2m_k^{(a)}}}{1-[\alpha_k+1/\alpha_k]z^{-1}+z^{-2}}, \tag{4.25b}$$

and

$$\Upsilon_k^{(b)}(z) = \sum_{l=0}^{2m_k^{(b)}-2}\zeta_k(l)z^{-lm_k^{(a)}}, \tag{4.25c}$$

with

$$\zeta_k(l) = \begin{cases} \dfrac{[(\alpha_k)^{m_k^{(a)}}]^{(l+1)}-[(1/\alpha_k)^{m_k^{(a)}}]^{(l+1)}}{(\alpha_k)^{m_k^{(a)}}-(1/\alpha_k)^{m_k^{(a)}}}, & 0 \le l \le m_k^{(b)}-1, \\ \zeta_k(2m_k^{(b)}-2-l), & m_k^{(b)} \le l \le 2m_k^{(b)}-2. \end{cases} \tag{4.25d}$$

Similarly, $H_k^{(c)}(z)$ for

$$N_k = n_k^{(a)}n_k^{(b)} \tag{4.26}$$

can be rewritten in the form

$$H_k^{(c)}(z) = 2\Re\Big\{\Gamma_k^{(a)}(z)\Gamma_k^{(b)}(z)\Big\}, \tag{4.27a}$$

where

$$\Gamma_k^{(a)}(z) = \gamma_k(\beta_k)^{(N_k-1)}\frac{1-[(\beta_k)^{n_k^{(a)}}+(1/\beta_k)^{n_k^{(a)}}]z^{-n_k^{(a)}}+z^{-2n_k^{(a)}}}{1-[\beta_k+1/\beta_k]z^{-1}+z^{-2}} \tag{4.27b}$$

and

$$\Gamma_k^{(b)}(z) = \sum_{l=0}^{2n_k^{(b)}-2}\eta_k(l)z^{-ln_k^{(a)}}, \tag{4.27c}$$

with

$$\eta_k(l) = \begin{cases} \dfrac{[(\beta_k)^{n_k^{(a)}}]^{(l+1)}-[(1/\beta_k)^{n_k^{(a)}}]^{(l+1)}}{(\beta_k)^{n_k^{(a)}}-(1/\beta_k)^{n_k^{(a)}}}, & 0\le l\le n_k^{(b)}-1, \\ \eta_k(2n_k^{(b)}-2-l), & n_k^{(b)}\le l\le 2n_k^{(b)}-2. \end{cases} \tag{4.27d}$$

Figures 4.2 and 4.3 present, respectively, the implementations of $z^{-(D+1-N_k)}H_k^{(c)}(z)$ and $z^{-(D+1-M_k)}H_k^{(r)}(z)$, where the principle of switching and resetting is applied and the scaling constants μ_k and ν_k are included.

Ignoring the extra delay elements as well as the effect of switching and resetting, the input-output relation for $H_k^{(c)}(z)$ can be described by the following difference equations:

$$y(l) = \sum_{i=0}^{2n_k^{(b)}-2} 2\,\mathrm{Re}\Big\{(\eta_k(i)/\nu_k)W(l-in_k^{(a)})\Big\}, \tag{4.28a}$$

where

$$W(l) = \Big(\beta_k+\frac{1}{\beta_k}\Big)W(l-1)-W(l-2)+V(l), \tag{4.28b}$$

and

$$V(l) = \nu_k\chi_k(0)[x(l)+x(l-2n_k^{(a)})]-\nu_k\chi_k(1)x(l-n_k^{(a)}), \tag{4.28c}$$

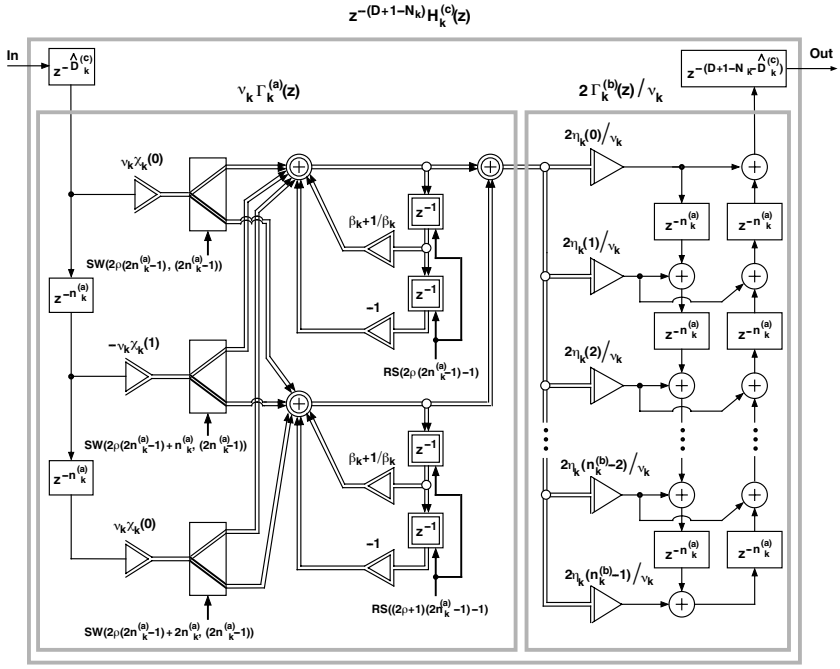

Figure 4.2: Implementation for $z^{-(D+1-N_k)} H_k^{(c)}(z)$.

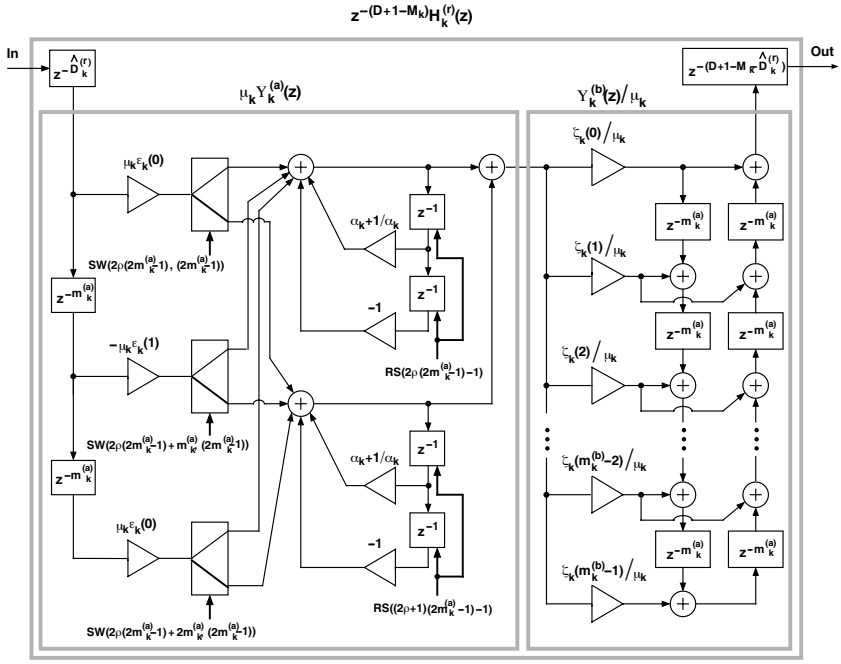

Figure 4.3: Implementation for $z^{-(D+1-M_k)} H_k^{(r)}(z)$.

Table 4.1: Real Arithmetic Input-Output Relation For $H_k^{(c)}(z)$

$$y(l) = \sum_{i=0}^{2n_k^{(b)}-2} \text{Re}\{2\eta_k(i)/\nu_k\}W_{\text{re}}(l - in_k^{(a)}) - \sum_{i=0}^{2n_k^{(b)}-2} \text{Im}\{2\eta_k(i)/\nu_k\}W_{\text{im}}(l - in_k^{(a)}),$$

where

$$W_{\text{re}}(l) = \text{Re}\{\beta_k + 1/\beta_k\}W_{\text{re}}(l-1) - \text{Im}\{\beta_k + 1/\beta_k\}W_{\text{im}}(l-1) + V_{\text{re}}(l)$$

$$W_{\text{im}}(l) = \text{Re}\{\beta_k + 1/\beta_k\}W_{\text{im}}(l-1) + \text{Im}\{\beta_k + 1/\beta_k\}W_{\text{re}}(l-1) + V_{\text{im}}(l)$$

with

$$V_{\text{re}}(l) = \text{Re}\{\nu_k\chi_k(0)\}[x(l) + x(l - 2n_k^{(a)})] - \text{Re}\{\nu_k\chi_k(1)\}x(l - n_k^{(a)})$$

$$V_{\text{im}}(l) = \text{Im}\{\nu_k\chi_k(0)\}[x(l) + x(l - 2n_k^{(a)})] - \text{Im}\{\nu_k\chi_k(1)\}x(l - n_k^{(a)})$$

with

$$\chi_k(0) = \gamma_k(\beta_k)^{N_k-1}, \tag{4.28d}$$

and

$$\chi_k(1) = \gamma_k(\beta_k)^{N_k-1}[(\beta_k)^{n_k^{(a)}} + (1/\beta_k)^{n_k^{(a)}}]. \tag{4.28e}$$

Here, $V(l)$ and $W(l)$ are complex numbers and $x(l)$ and $y(l)$ are real. The corresponding real-arithmetic input-output relation is given in Table 4.1, where $V(l) = V_{\text{re}}(l) + jV_{\text{im}}(l)$ and $W(l) = W_{\text{re}}(l)+jW_{\text{im}}(l)$. Since the order of $\Gamma_k^{(a)}(z)$ is $2(n_k^{(a)}-1)$, the demultiplexers are switched at every $2n_k^{(a)} - 1$ samples. The feedforward part is realized only once. This is made possible by synchronizing three demultiplexers in such a way that if the first demultiplexer starts giving actual data to the first feedback loop at time $n = \varrho(2n_k^{(a)} - 1)$, where ϱ is a non-negative integer, then the second and the third demultiplexers start giving data to this loop at time instants $n = \varrho(2n_k^{(a)} - 1) + n_k^{(a)}$ and $n = \varrho(2n_k^{(a)} - 1) + 2n_k^{(a)}$, respectively. The data is complex only in the feedback loops. This is made possible by

using the direct-form and the transposed structures for the feedforward parts before and after the feedbacks, respectively.

The scaling constant ν_k appearing in (4.28) is determined to avoid overflows in the real and imaginary parts of the feedback loop of Fig. 4.2. Ignoring the extra delay terms, the unit sample responses of the corresponding scaling transfer functions are given by

$$b_k^{(\text{real})}(l) = \begin{cases} \text{Re}\Big\{\gamma_k(\beta_k)^{N_k-1}\dfrac{(\beta_k)^{(l+1)} - (1/\beta_k)^{(l+1)}}{\beta_k - 1/\beta_k}\Big\}, & 0 \le l \le n_k^{(a)} - 1, \\ b_k^{(\text{real})}(2n_k^{(a)} - 2 - l), & n_k^{(a)} \le l \le 2n_k^{(a)} - 2, \\ 0, & \text{otherwise,} \end{cases} \tag{4.29a}$$

and

$$b_k^{(\text{imag})}(l) = \begin{cases} \text{Im}\Big\{\gamma_k(\beta_k)^{N_k-1}\dfrac{(\beta_k)^{(l+1)} - (1/\beta_k)^{(l+1)}}{\beta_k - 1/\beta_k}\Big\}, & 0 \le l \le n_k^{(a)} - 1, \\ b_k^{(\text{imag})}(2n_k^{(a)} - 2 - l), & n_k^{(a)} \le l \le 2n_k^{(a)} - 2, \\ 0, & \text{otherwise.} \end{cases} \tag{4.29b}$$

By $\beta_k \to \alpha_k, n_k^{(a)} \to m_k^{(a)}, \chi_k \to \varepsilon_k$, and $\nu_k \to \mu_k$, (4.28a)–(4.28e) can be modified to describe the input-output relation for $H_k^{(r)}(z)$, as given by (4.25a) and presented in Fig. 4.3. Now

$$\varepsilon_k(0) = \kappa_k(\alpha_k)^{M_k-1} \tag{4.30a}$$

and

$$\varepsilon_k(1) = \kappa_k(\alpha_k)^{M_k-1}[(\alpha_k)^{m_k^{(a)}} + (1/\alpha_k)^{m_k^{(a)}}]. \tag{4.30b}$$

Moreover, similar to the role of ν_k, the scaling constant μ_k is determined to avoid overflows in the feedback loop of Fig. 4.3, where the unit sample response of the scaling transfer function is given by

$$b_k(l) = \begin{cases} \kappa_k(\alpha_k)^{M_k-1}\dfrac{(\alpha_k)^{(l+1)} - (1/\alpha_k)^{(l+1)}}{(\alpha_k) - (1/\alpha_k)}, & 0 \le l \le m_k^{(a)} - 1, \\ b_k(2m_k^{(a)} - 2 - l), & m_k^{(a)} \le l \le 2m_k^{(a)} - 2, \\ 0, & \text{otherwise.} \end{cases} \tag{4.31}$$

The nonrecursive parts of $H_k^{(r)}(z)$ and $H_k^{(c)}(z)$ can be factorized to reduce the required number of real-valued multiplications and real-valued additions in their implementations. To elaborate this, consider the identity

$$\sum_{l=0}^{m^{(1)}m^{(2)}\cdots m^{(I)}-1} x^{lm^{(a)}} = \prod_{i=1}^{I}\left[\sum_{l=0}^{m^{(i)}-1} x^{lM^{(i)}}\right], \tag{4.32}$$

where $M^{(1)} = m^{(a)}$ and $M^{(i)} = m^{(a)}m^{(1)}m^{(2)}\cdots m^{(i-1)}$ for $i = 2, 3, \ldots, I$. Based on this identity, $\Upsilon_k^{(b)}(z)$, as given by (4.25c), for

$$m_k^{(b)} = \prod_{i=1}^{I_k} m_k^{(i)} \tag{4.33a}$$

can be expressed as

$$\Upsilon_k^{(b)}(z) = \prod_{i=1}^{I_k} \Upsilon_k^{(i)}(z), \tag{4.33b}$$

where

$$\Upsilon_k^{(i)}(z) = \sum_{l=0}^{2m_k^{(i)}-2} \zeta_k^{(i)}(l) z^{-lM_k^{(i)}}, \tag{4.33c}$$

with

$$\zeta_k(l) = \begin{cases} \dfrac{[(\alpha_k)^{M_k^{(i)}}]^{(l+1)} - [(1/\alpha_k)^{M_k^{(i)}}]^{(l+1)}}{(\alpha_k)^{M_k^{(i)}} - (1/\alpha_k)^{M_k^{(i)}}}, & 0 \le l \le m_k^{(i)} - 1, \\ \zeta_k(2m_k^{(i)} - 2 - l), & m_k^{(i)} \le l \le 2m_k^{(i)} - 2, \end{cases} \tag{4.33d}$$

and

$$M_k^{(i)} = m_k^{(a)} \prod_{l=1}^{i-1} m_k^{(l)}. \tag{4.33e}$$

Similarly, $\Gamma_k^{(b)}(z)$, as given by (4.27c), for

$$n_k^{(b)} = \prod_{j=1}^{J_k} n_k^{(j)} \tag{4.34a}$$

can be written as

$$\Gamma_k^{(b)}(z) = \prod_{j=1}^{J_k} \Gamma_k^{(j)}(z), \tag{4.34b}$$

where

$$\Gamma_k^{(j)}(z) = \sum_{l=0}^{2n_k^{(j)}-2} \eta_k^{(j)}(l) z^{-lN_k^{(j)}}, \tag{4.34c}$$

with

$$\eta_k^{(j)}(l) = \begin{cases} \dfrac{[(\beta_k)^{N_k^{(j)}}]^{(l+1)} - [(1/\beta_k)^{N_k^{(j)}}]^{(l+1)}}{(\beta_k)^{N_k^{(i)}} - (1/\beta_k)^{N_k^{(j)}}}, & 0 \le l \le n_k^{(j)} - 1, \\ \eta_k(2n_k^{(j)} - 2 - l), & n_k^{(j)} \le l \le 2n_k^{(j)} - 2, \end{cases} \tag{4.34d}$$

and

$$N_k^{(j)} = n_k^{(a)} \prod_{l=1}^{j-1} n_k^{(l)}. \tag{4.34e}$$

The scaled implementations for the above transfer functions are depicted in Figs. 4.4 and 4.5. For $\Gamma_k^{(b)}(z)$, the constants $\varphi_k^{(j)}$ for $j = 1, 2 \ldots, J_k - 1$ are determined to avoid the overflows at the outputs of the $\Gamma_k^{(j)}(z)$'s based on the fact that the unit sample responses of the scaling transfer functions to the real and imaginary parts of these outputs are obtained from (4.29a) and (4.29b) by using the substitutions

$$n_k^{(a)} \rightarrow n_k^{(a)} \prod_{l=1}^{j} n_k^{(j)} \tag{4.35a}$$

and

$$\nu_k \rightarrow \nu_k \prod_{i=1}^{j-1} \varphi_k^{(j)}. \tag{4.35b}$$

$\varphi_k^{(J_k)}$ is then given by

$$\varphi_k^{(J_k)} = 1 \Big/ \Big[\nu_k \prod_{j=1}^{J_k-1} \varphi_k^{(j)} \Big]. \tag{4.36}$$

The scaling constants $\psi_k^{(i)}$ for $\Upsilon_k^{(b)}(z)$ are determined in a similar manner.

Table 4.2 compares the number of real-valued multiplications[1] and real-valued additions required by multistage implementations to those of single-stage equivalents in two

[1] $\eta_k^{(j)}(0) = 1$ for $j = 1, 2, \ldots J_k$ so that the $\varphi_k^{(j)}\eta_k^{(j)}(0)$'s are real-valued.

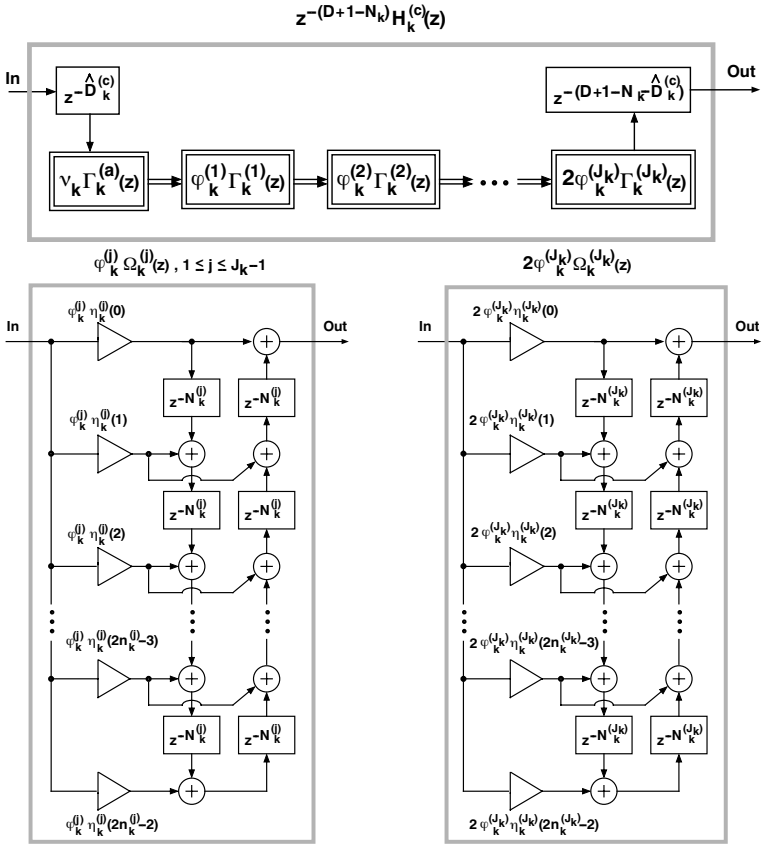

Figure 4.4: Generalized implementation for $z^{-(D+1-M_k)}H_k^{(c)}(z)$.

cases. In the first case, the filters are scaled as shown in Figs. 4.4 and 4.5, whereas, in the second case, the filter scalings are taken care of by a single constant μ_k or ν_k and the fact that now $\psi_k^{(i)}\zeta_k^{(j)}(0) = 1$ for $i = 1, 2, \ldots, I_k - 1$ and $\varphi_k^{(j)}\eta_k^{(j)}(0) = 1$ for $j = 1, 2, \ldots, J_k - 1$. It is seen from the table that the multistage cases provide significant savings when the $m_k^{(i)}$'s and the $n_k^{(j)}$'s are relatively small (2 or 3).

If desired, the scaled $\Upsilon_k^{(a)}(z)$ and $\Gamma_k^{(a)}(z)$ can also be implemented in the following non-recursive forms:

$$\mu_k \Upsilon_k^{(a)}(z) = \sum_{l=0}^{2m_k^{(a)}-2} \mu_k \zeta_k^{(a)}(l) z^{-l} \tag{4.37a}$$

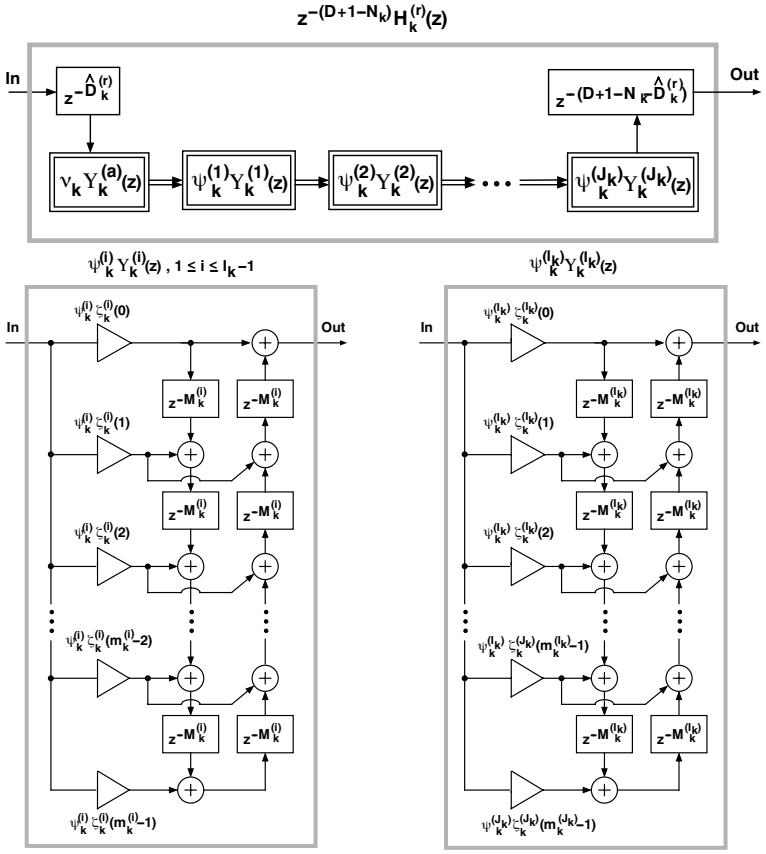

Figure 4.5: Generalized implementation for $z^{-(D+1-N_k)}H_k^{(r)}(z)$.

and

$$\nu_k \Gamma_k^{(a)}(z) = \sum_{l=0}^{2n_k^{(a)}-2} \nu_k \eta_k^{(a)}(l) z^{-l}, \tag{4.37b}$$

where

$$\zeta_k^{(a)}(l) = \begin{cases} \kappa_k(\alpha_k)^{M_k-1} \dfrac{(\alpha_k)^{(l+1)} - (1/\alpha_k)^{(l+1)}}{\alpha_k - 1/\alpha_k}, & 0 \le l \le m_k^{(a)} - 1, \\ \zeta_k^{(a)}(2m_k^{(a)} - 2 - l), & m_k^{(a)} \le l \le 2m_k^{(a)} - 2, \end{cases} \tag{4.37c}$$

Table 4.2: Comparison Between Different Implementations of $H_k^{(r)}(z)$ and $H_k^{(c)}(z)$ in Parallel Structure II

	Number of Real-Valued Multiplications	Number of Real-Valued Additions
$\Upsilon_k^{(b)}(z)$		
Single-Stage	$\prod_{i=1}^{I_k} m_k^{(i)}$	$2\big[\prod_{i=1}^{I_k} m_k^{(i)}\big] - 2$
Multistage General Scaling	$\sum_{i=1}^{I_k} m_k^{(i)}$	$\sum_{i=1}^{I_k}(2m_k^{(i)} - 2)$
Multistage One Scaler	$1 + \sum_{i=1}^{I_k}(m_k^{(i)} - 1)$	$\sum_{i=1}^{I_k}(2m_k^{(i)} - 2)$
$\Upsilon_k^{(a)}(z)$		
Recursive	4	5
Non-recursive	$m_k^{(a)}$	$2m_k^{(a)} - 2$
$\Gamma_k^{(b)}(z)$		
Single-Stage	$2\big[\prod_{j=1}^{J_k} n_k^{(j)}\big] - 1$	$2\big[\prod_{j=1}^{J_k} n_k^{(j)}\big] - 2$
Multistage General Scaling	$2n_k^{(J_k)} - 1 + \sum_{j=1}^{J_k-1}(4n_k^{(j)} - 2)$	$2n_k^{(J_k)} - 2 + \sum_{j=1}^{J_k-1} 4(n_k^{(j)} - 1)$
Multistage One Scaler	$2n_k^{(J_k)} - 1 + \sum_{j=1}^{J_k-1} 4(n_k^{(j)} - 1)$	$2n_k^{(J_k)} - 2 + \sum_{j=1}^{J_k-1} 4(n_k^{(j)} - 1)$
$\Gamma_k^{(a)}(z)$		
Recursive	10	20
Non-recursive	$2n_k^{(a)} - 1$	$2n_k^{(a)} - 2$

and

$$\eta_k^{(a)}(l) = \begin{cases} \gamma_k(\beta_k)^{N_k-1}\dfrac{(\beta_k)^{(l+1)} - (1/\beta_k)^{(l+1)}}{\beta_k - 1/\beta_k}, & 0 \le l \le n_k^{(a)} - 1, \\ \eta_k^{(a)}(2n_k^{(a)} - 2 - l), & n_k^{(a)} \le l \le 2n_k^{(a)} - 2. \end{cases} \tag{4.37d}$$

From Table 4.2, it is seen that the non-recursive implementations as shown in Fig. 4.6 become more attractive than the recursive equivalents for small values of $m_k^{(a)}$ and $n_k^{(a)}$.

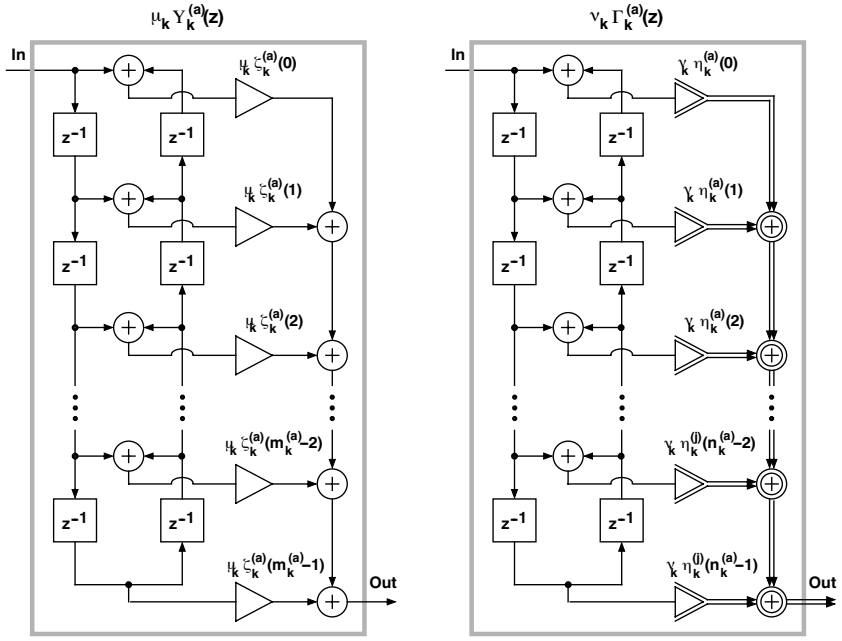

Figure 4.6: Non-recursive implementations for $\Upsilon_k^{(a)}(z)$ and $\Gamma_k^{(a)}(z)$.

Implementation of Delays

The number of delay elements required by the overall structure of Fig. 4.1 can be minimized by determining

$$\widetilde{D}_{\text{in}} = \max\Big\{ \max_{1\le k\le K_R}\{2m_k^{(a)}\}, \max_{1\le k\le K_C}\{2n_k^{(a)}\}\Big\}, \tag{4.38a}$$

$$\widetilde{D}_{\text{out}} = \max\Big\{ \max_{1\le k\le K_R}\{2m_k^{(a)}(m_k^{(b)}-1)\}, \max_{1\le k\le K_C}\{2n_k^{(a)}(n_k^{(b)}-1)\}\Big\}, \tag{4.38b}$$

$$\widehat{D}_k^{(r)} = \min\{D+1-2M_k, \widetilde{D}_{\text{in}} - 2m_k^{(a)}\}, \quad k=1,2,\ldots,K_R, \tag{4.38c}$$

and

$$\widehat{D}_k^{(c)} = \min\{D+1-2N_k, \widetilde{D}_{\text{in}} - 2n_k^{(a)}\}, \quad k=1,2,\ldots,K_C. \tag{4.38d}$$

This minimization can be performed by using the common delay blocks $z^{-\widetilde{D}_{\text{in}}}$ and

$z^{-\widetilde{D}_{\text{out}}}$ at the filter input and output, respectively. With these selections, the block delays $z^{-\widehat{D}_k^{(c)}}$ and $z^{-\widehat{D}_k^{(r)}}$ in Figs. 4.2 and 4.3 as well as the $z^{-n_k^{(a)}}$'s and $z^{-m_k^{(a)}}$'s before the feedback loops can be obtained from $z^{-\widetilde{D}_{\text{in}}}$, whereas $z^{-\widetilde{D}_{\text{out}}}$ provides delays for the $z^{-(D+1-N_k-\widehat{D}_k^{(c)})}$'s and $z^{-(D+1-M_k-\widehat{D}_k^{(r)})}$'s as well as for the $z^{-n_k^{(a)}}$'s and $z^{-m_k^{(a)}}$'s after the feedback loops. In most practical cases, $\widetilde{D}_{\text{in}}$ and $\widetilde{D}_{\text{out}}$ are determined by one or two outermost complex pole pairs. In this case, $\widetilde{D}_{\text{in}} + \widetilde{D}_{\text{out}}$, the delays required by feedforward parts, is equal to $2D + 2$, which is two more than the overall filter order. Including the delays in the feedback loops, the number of delay elements required by the overall filter is $2D+4K_R+8K_C+2$. When implementing the two copies of the feedback loops by demultiplexing the same filter and exploiting the symmetries in the coefficients after the feedback loops, the implementations of $z^{-(D+1-M_k)}H_k^{(r)}(z)$ and $z^{-(D+1-N_k)}H_k^{(c)}(z)$ require $m_k^{(b)}+4$ and $2n_k^{(b)}+9$ real-valued multiplications, respectively.[2] The corresponding number of real adders are $2n_k^{(a)}+6$ and $2m_k^{(b)}+2$. $A(z)$ requires $L-K+1$ multipliers and $L-K$ adders. For the overall filter, the number of real-valued multiplications and real-valued additions are

$$\sum_{k=1}^{K_R} m_k^{(a)} + \sum_{k=1}^{K_C} 2n_k^{(a)} + 4K_R + 9K_C + L - K + 1$$

and

$$2\sum_{k=1}^{K_R} m_k^{(a)} + 2\sum_{k=1}^{K_C} 2n_k^{(a)} + 2K_R + 6K_C + L - K,$$

respectively.

[2] $\eta_k(0) = 1$ so that $2\eta_k(0)/\nu_k$ is real-valued.

4.2.2 Parallel Structure III

For the implementation of Parallel Structure III, $F^{(0)}(z)$ in (4.6) is multiplied by z^{-D} where

$$D = \max\{M_1 - 1, M_2 - 1, \cdots, M_{K_R} - 1, 2N_1 - 1, 2N_2 - 1, \cdots, 2N_{K_C} - 1\}, \quad (4.39)$$

yielding

$$H(z) = z^{-(D-(L-K))}A(z) + \sum_{k=1}^{K_R} z^{-(D+1-M_k)}H_k^{(r)}(z) + \sum_{k=1}^{K_C} z^{-(D+1-2N_k)}H_k^{(c)}(z), \quad (4.40a)$$

where $H^{(r)}(z)$ and $A(z)$ are the same as those for Parallel Structure II, whereas

$$H_k^{(c)}(z) = 2\Omega_k(z)\Re\Big\{\frac{\widehat{\gamma}_k}{1 - [\beta_k + 1/\beta_k]z^{-1} + z^{-2}}\Big\}, \quad (4.40b)$$

where

$$\widehat{\gamma}_k = -(\beta_k\beta_k^*)^{N_k}\gamma_k/\beta_k \quad (4.40c)$$

and

$$\Omega_k(z) = [1 + z^{-4N_k}] - \chi_k(0)[z^{-N_k} + z^{-3N_k}] + \chi_k(1)z^{-2N_k} \quad (4.40d)$$

with

$$\chi_k(0) = (\beta_k)^{N_k} + (\beta_k^*)^{N_k} + (1/\beta_k)^{N_k} + (1/\beta_k^*)^{N_k} \quad (4.40e)$$

and

$$\chi_k(1) = 2 + \Big((\beta_k)^{N_k} + (1/\beta_k)^{N_k}\Big)\Big((\beta_k^*)^{N_k} + (1/\beta_k^*)^{N_k}\Big). \quad (4.40f)$$

For

$$N_k = n_k^{(a)}n_k^{(b)} \quad (4.41a)$$

with

$$n_k^{(b)} = \prod_{j=1}^{J_k} n_k^{(j)}, \quad (4.41b)$$

$\Omega_k(z)$, as given by (4.40d), can be factorized in the form

$$\Omega_k(z) = \Omega_k^{(a)}(z)\Omega_k^{(b)}(z), \tag{4.41c}$$

where

$$\Omega_k^{(b)}(z) = \prod_{j=1}^{J_k} \Omega_k^{(j)}(z). \tag{4.41d}$$

Here,

$$\Omega_k^{(a)}(z) = [1 + z^{-4n_k^{(a)}}] - \varepsilon_k(0)[z^{-n_k^{(a)}} + z^{-3n_k^{(a)}}] + \varepsilon_k(1) z^{-2n_k^{(a)}} \tag{4.42a}$$

and

$$\Omega_k^{(j)}(z) = \sum_{l=0}^{4n_k^{(j)}-4} \eta_k^{(j)}(l) z^{-lN_k^{(j)}}, \tag{4.42b}$$

where

$$\varepsilon_k(0) = (\beta_k)^{n_k^{(a)}} + (\beta_k^*)^{n_k^{(a)}} + (1/\beta_k)^{n_k^{(a)}} + (1/\beta_k^*)^{n_k^{(a)}}, \tag{4.42c}$$

$$\varepsilon_k(1) = 2 + \Big((\beta_k)^{n_k^{(a)}} + (1/\beta_k)^{n_k^{(a)}}\Big)\Big((\beta_k^*)^{n_k^{(a)}} + (1/\beta_k^*)^{n_k^{(a)}}\Big), \tag{4.42d}$$

$$N_k^{(j)} = n_k^{(a)} \prod_{l=1}^{j-1} n_k^{(l)}, j = 1 \dots J \tag{4.42e}$$

and

$$\eta_k^{(j)}(l) = \begin{cases} -\vartheta_k(l/2)\vartheta_k^*(l/2) + 2\,\mathrm{Re}\Big\{\sum_{i=0}^{l/2} \vartheta_k(i)\vartheta_k^*(l-i)\Big\}, & l \text{ even}, 0 \le l \le 2n_k^{(j)} - 2, \\ 2\,\mathrm{Re}\Big\{\sum_{i=0}^{(l-1)/2} \vartheta_k(i)\vartheta_k^*(l-i)\Big\}, & l \text{ odd}, 0 \le l \le 2n_k^{(j)} - 2, \\ \eta_k^{(j)}(l) & 2n_k^{(a)} - 1 \le l \le 4n_k^{(a)} - 4, \end{cases} \tag{4.42f}$$

with

$$\vartheta_k(l) = \begin{cases} \dfrac{[(\beta_k)^{N_k^{(j)}}]^{(l+1)} - [(1/\beta_k)^{N_k^{(j)}}]^{(l+1)}}{(\beta_k)^{N_k^{(j)}} - (1/\beta_k)^{N_k^{(j)}}}, & 0 \le l \le n_k^{(j)} - 1, \\ \vartheta_k(2n_k^{(j)} - 2 - l), & n_k^{(j)} \le l \le 2n_k^{(j)} - 2. \end{cases} \tag{4.42g}$$

From (4.41), it follows that if $J = 1$, then $N_k^{(j)} = n_k^{(a)}$ and $n_k^{(j)} = n_k^{(b)}$. For this special case, $\eta_k^{(1)} \equiv \eta_k^{(b)}$, and $\Omega_k^{(b)}(z)$ can be expressed as:

$$\Omega_k^{(b)}(z) = \sum_{l=0}^{4n_k^{(b)}-4} \eta_k^{(b)}(l) z^{-l n_k^{(a)}}. \tag{4.43}$$

Based on the factorization of (4.42), $H_k^{(c)}(z)$ can be written as

$$H_k^{(c)}(z) = \Gamma_k^{(a)}(z) \left[2 \prod_{j=1}^{J_k} \Omega_k^{(j)}(z) \right], \tag{4.44a}$$

where

$$\Gamma_k^{(a)}(z) = \Omega_k^{(a)}(z) \cdot \Re\Big\{ \frac{\widehat{\gamma}_k}{1 - [\beta_k + 1/\beta_k] z^{-1} + z^{-2}} \Big\}. \tag{4.44b}$$

An implementation for the overall filter is shown in Fig. 4.7, whereas a scaled implementation for $z^{-(D+1-2N_k)} H_k^{(c)}(z)$ is depicted in Fig. 4.8. In order to determine the scaling constants $\nu_k^{(1)}$, $\nu_k^{(2)}$, $\varphi_k^{(1)}$, $\varphi_k^{(2)}$, $\ldots$, and $\varphi_k^{(J_k)}$, the following unit sample response is defined:

$$b(l, N, \upsilon) = \begin{cases} \dfrac{\upsilon \tau^{(1)}(l)}{\beta_k - 1/\beta_k}, & 0 \le l \le N-1, \\ \dfrac{\upsilon \tau^{(2)}(l)}{\beta_k - 1/\beta_k}, & N \le l \le 2N-1, \\ b(4N-2-l, N, \upsilon), & 2N \le l \le 4N-2, \\ 0, & \text{otherwise}, \end{cases} \tag{4.45a}$$

where

$$\tau^{(1)}(l) = (\beta_k)^{(l+1)} - (\beta_k)^{-(l+1)} \tag{4.45b}$$

and

$$\begin{aligned} \tau^{(2)}(l) = & -(\beta_k)^{(l-N+1)} [(\beta_k^*)^N + (\beta_k^*)^{-N} + (\beta_k)^{-N}] \\ & + (\beta_k)^{-(l-N+1)} [(\beta_k^*)^N + (\beta_k^*)^{-N} + (\beta_k)^N]. \end{aligned} \tag{4.45c}$$

$\nu_k^{(1)}$ is determined to avoid overflows at the outputs of the feedback loops, based on the fact that the unit sample responses to the real and imaginary parts are, respectively, $\text{Re}\{b(l, n_k^{(a)}, \nu_k^{(1)}\}$ and $\text{Im}\{b(l, n_k^{(a)}, \nu_k^{(1)}\}$. $\nu_k^{(2)}$ is used to scale the output of $\nu_k^{(1)}\nu_k^{(2)}\Gamma_k^{(a)}(z)$. The corresponding unit sample response is $\text{Re}\{b(l, n_k^{(a)}, \nu_k^{(1)}\nu_k^{(2)}\widehat{\gamma}_k\}$. The $\varphi_k^{(j)}$'s for $j = 1, 2 \ldots J_k - 1$ scale the outputs of the corresponding $\Omega_k^{(j)}(z)$'s. The unit sample response from the overall filter input to these outputs is

$$\text{Re}\left\{ b(l, N_k^{(j+1)}, \nu_k^{(1)}\nu_k^{(2)}\widehat{\gamma}_k \prod_{l=1}^{j-1} \varphi_k^{(l)} \right\}. \tag{4.46}$$

Finally, $\varphi_k^{(J_k)}$ is used to compensate the scalings and is given by

$$\varphi_k^{(J_k)} = \frac{1}{\nu_k^{(1)}\nu_k^{(2)} \prod_{l=1}^{J_k-1} \varphi_k^{(l)}}. \tag{4.47}$$

Table 4.3 gives the number of real-valued multiplications and real-valued additions for the above $\Omega_k^{(b)}(z)$ in two cases in a manner similar to Parallel Structure II. In the first case, all the scaling constants are used and, in the second case, only $\nu_k^{(1)}$ and $\nu_k^{(2)}$ take care of the filter scaling. In the latter case, $\varphi^{(j)}\eta_k^{(j)}(0) = 1$ for $j = 1, 2, \ldots, J_k - 1$. For comparison purposes, the data for the single-stage $\Omega_k^{(b)}(z)$ is also included.

Alternatively, the scaled $\Gamma_k^{(a)}(z)$ can be expressed non-recursively as

$$\begin{aligned} \nu_k^{(1)}\nu_k^{(2)}\Gamma_k^{(a)}(z) = &\Big[\sum_{l=0}^{4n_k^{(a)}-4} \nu_k^{(1)}\eta_k^{(a)}(l) z^{-l} \Big] \times \\ &\Big[\nu_k^{(2)} \,\text{Re}\{\widehat{\gamma}_k\}[1 + z^{-2}] - \nu_k^{(2)} \,\text{Re}\{\widehat{\gamma}_k(\beta_k^* + 1/\beta_k^*)\} z^{-1} \Big], \end{aligned} \tag{4.48}$$

where the $\eta_k^{(a)}(l)$'s can be determined from (4.42f) and (4.42g) using $N_k^{(j)} = 1$ and $n_k^{(j)} = n_k^{(a)}$. The above transfer function is a cascade of two conventional linear-phase FIR filters of orders $4n_k^{(a)} - 4$ and 2, respectively. Table 4.3 compares the non-recursive and recursive implementations of $\Gamma_k^{(a)}(z)$.

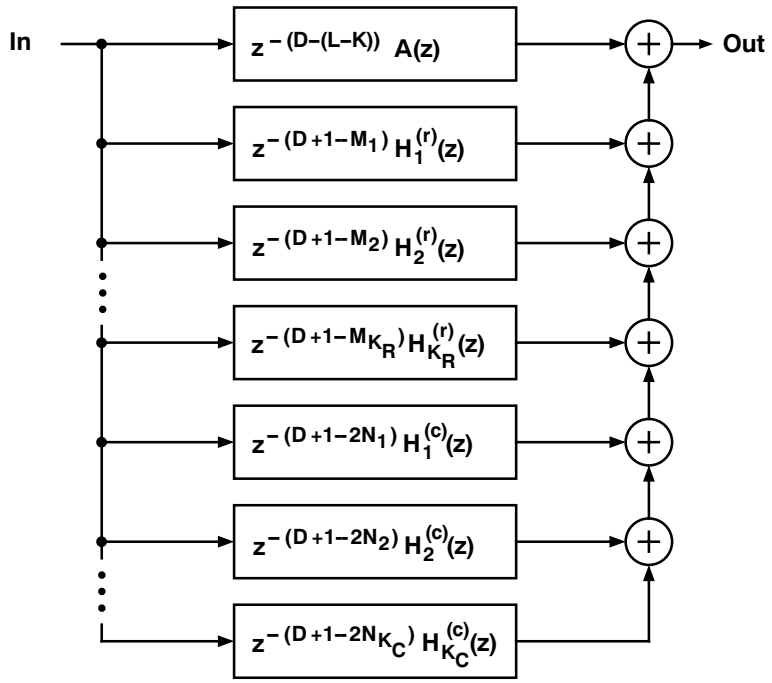

Figure 4.7: Structure for a linear-phase FIR filter obtained by using the Parallel Structure III.

Of special interest is the case, where $\Gamma_k^{(a)}(z)$ is implemented non-recursively and $n_k^{(j)} = 2$ for $j = 1, 2, \ldots, J_k$ so that, $N_k = n_k^{(a)} 2^{J_k}$. By this setting,

$$\Omega_k^{(j)}(z) = [1 + z^{-4N_k^{(j)}}] + \eta_k^{(j)}(1)[z^{-N_k^{(j)}} + z^{-3N_k^{(j)}}] + \eta_k^{(j)}(2) z^{-2N_k^{(j)}}. \tag{4.49}$$

When two scaling constants are used, this implementation requires only $2n_k^{(a)}+2+2J_k$ real-valued multiplications and $4n_k^{(a)} - 2 + 4J_k$ real-valued additions to generate a linear-phase FIR filter response of length $2n_k^{(a)} 2^{J_k} - 2$.

4.3 Noise Analysis

In order to analyze the output noise due to the multiplication roundoff errors taking place in the feedback loops of $\widehat{H}_k^{(c)}(z)$ and $\widehat{H}_k^{(r)}(z)$, the notations of (3.59) are applied to $\widehat{H}_k^{(c)}(z)$

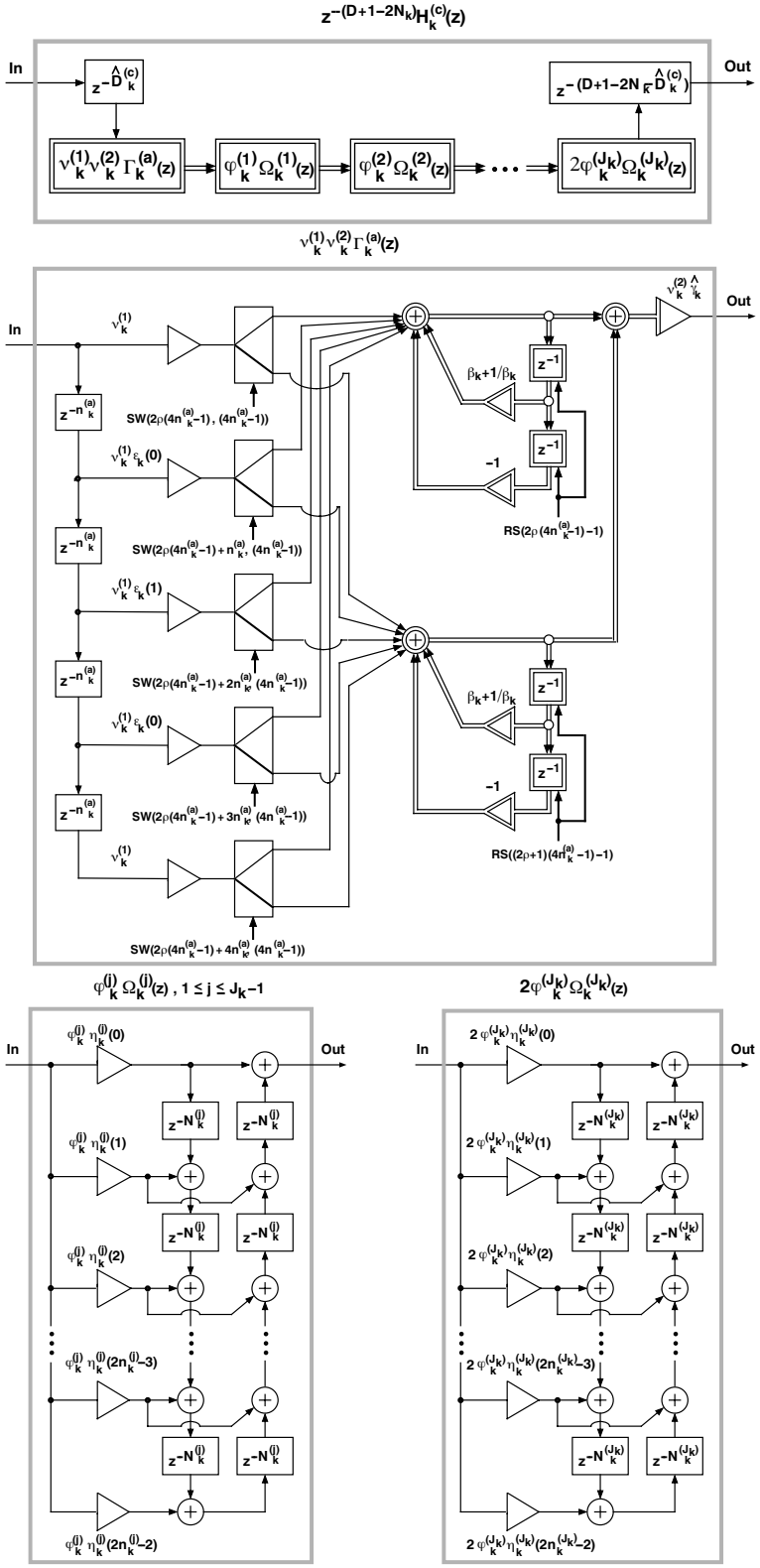

Figure 4.8: Generalized implementation for $z^{-(D+1-N_k)}\,\widehat{H}_k^{(c)}(z)$.

Table 4.3: Comparison Between Different Implementations of $H^{(c)}(z)$ in Parallel Structure III

	Number of Real-Valued Multiplications	Number of Real-Valued Additions
$\Omega_k^{(b)}(z)$		
Single-Stage	$2\Big[\prod_{j=1}^{J_k} n_k^{(j)}\Big] - 1$	$4\Big[\prod_{j=1}^{J_k} n_k^{(j)}\Big] - 2$
Multistage General Scaling	$\sum_{j=1}^{J_k} \big(2n_k^{(j)} - 1\big)$	$\sum_{j=1}^{J_k} \big(4n_k^{(j)} - 4\big)$
Multistage Two Scalers	$1 + \sum_{j=1}^{J_k} \big(2n_k^{(j)} - 2\big)$	$\sum_{j=1}^{J_k} \big(4n_k^{(j)} - 4\big)$
$\Gamma_k^{(a)}(z)$		
Recursive	11	12
Non-recursive	$2n_k^{(a)} + 1$	$4n_k^{(a)} - 2$

and $\widehat{H}_k^{(r)}(z)$ in the same sense that they were applied to $\widehat{E}_k^{(c)}(z)$ and $\widehat{E}_k^{(r)}(z)$. In what follows, the roundoff noise generated in the proposed structures are analyzed in detail.

4.3.1 Parallel Structure II

Because of the switching and resetting, the (complex) roundoff error at the output of the upper feedback loop of $z^{-(D+1-N_k)}H_k^{(c)}(z)$, $\Xi_{k,\,\text{out}}^{(1)}(l)$, is zero at the time instant

$$l = 2\rho(2n_k^{(a)} - 1) - 1, \tag{4.50a}$$

where ρ is an integer, whereas for $0 \le \lambda \le 4n_k^{(a)} - 4$,

$$\Xi_{k,\,\text{out}}^{(1)}(2\rho(2n_k^{(a)} - 1) + \lambda) = \sum_{i=0}^{\lambda} \frac{\beta_k^{(\lambda+1-i)} - (1/\beta_k)^{(\lambda+1-i)}}{\beta_k - 1/\beta_k} \Xi_{k,\,\text{in}}^{(1)}(2\rho(2n_k^{(a)} - 1) + i). \tag{4.50b}$$

Similarly the (complex) roundoff error at the output of the lower feedback loop of $z^{-(D+1-N_k)}H_k^{(c)}(z)$, $\Xi_{k,\,\text{out}}^{(2)}(l)$, is zero at the time instant

$$l = (2\rho - 1)(2n_k^{(a)} - 1) - 1, \tag{4.51a}$$

whereas for $0 \le \lambda \le 4n_k^{(a)} - 4$,

$$\Xi_{k,\,\text{out}}^{(2)}((2\rho-1)(2n_k^{(a)}-1)+\lambda) = \sum_{i=0}^{\lambda} \frac{\beta_k^{(\lambda+1-i)} - (1/\beta_k)^{(\lambda+1-i)}}{\beta_k - 1/\beta_k} \times \Xi_{k,\,\text{in}}^{(2)}((2\rho-1)(2n_k^{(a)}-1)+i). \tag{4.51b}$$

Equations (4.50) and (4.51) can be expressed more compactly by defining

$$\Xi_{k,\,\text{out}}^{(m)}(l,j,n) = \begin{cases} 0, & \lambda = n-1, \\ \displaystyle\sum_{i=0}^{\lambda} \frac{\beta_k^{(\lambda+1-i)} - (1/\beta_k)^{(\lambda+1-i)}}{\beta_k - 1/\beta_k} \Xi_{k,\,\text{in}}^{(m)}(l-\lambda+i), & \text{otherwise,} \end{cases} \tag{4.52a}$$

where

$$\lambda = (l-j)\text{mod}(n). \tag{4.52b}$$

For the upper branch, $m = 1, j = 0$ and $n = 4n_k^{(a)} - 2$, while for the lower branch, $m = 2, j = n = 2n_k^{(a)} - 1$ and $n = 4n_k^{(a)} - 2$. Equations (4.50)–(4.52) apply to $z^{-(D+1-M_k)}H_k^{(r)}(z)$ by $n_k^{(a)} \to m_k^{(a)}$ and $\beta_k \to \alpha_k$.

Using the notation of (4.52a), the overall output noise generated in the structure of Fig. 4.3 at time l can be expressed as

$$\xi_k^{(\text{ove})}(l) = \sum_{i=0}^{2m_k^{(b)}-2} \text{Re}\left\{ \frac{\zeta_k(i)}{\mu_k} \left[\Xi_{k,\,\text{out}}^{(1)}(l - im_k^{(a)}, 0, 4m_k^{(a)} - 2) + \Xi_{k,\,\text{out}}^{(2)}(l - im_k^{(a)}, 2m_k^{(a)} - 1, 4m_k^{(a)} - 2) \right] \right\}. \tag{4.53}$$

By considering the effects of switching and resetting, and by taking into account the fact that $\Xi_{k,\,\text{in}}^{(1)}(l)$ and $\Xi_{k,\,\text{in}}^{(2)}(l)$ consist of the sum of the input noises from the demultiplexers

and the round-off noise generated in the coefficient of the recursive filter, the variance of the output noise at the input of $\Upsilon_k^{(b)}(z)/\mu_k$ at any arbitrary time l can be expressed as:

$$(\sigma^2)_k^{(\text{out})}(l) = (\sigma^2)_k^{(1)}(l) + (\sigma^2)_k^{(1)}(l - 2m_k^{(a)} + 1), \tag{4.54a}$$

where

$$(\sigma^2)_k^{(1)}(l) = \begin{cases} 0, & \delta(l) = 4m_k^{(a)} - 3, \\ \sigma_e^2 \left[\sum_{i=0}^{\delta(l)} c(i) T_i^2 + \sum_{i=0}^{\delta(l)-1} T_i^2 \right], & \text{otherwise.} \end{cases} \tag{4.54b}$$

The polynomials T_n in (4.54b) are defined recursively as $T_{-1} = 0$, $T_0 = 1$, and

$$T_n = (\alpha_k + 1/\alpha_k) T_{n-1} - T_{n-2}, \tag{4.54c}$$

where

$$c(i) = \begin{cases} 2, & \delta(l) - 2m_k^{(a)} + 2 \le i \le \delta(l) - m_k^{(a)} \\ & \text{or } \delta(l) - 3m_k^{(a)} + 2 \le i \le \delta(l) - 2m_k^{(a)}, \\ 1, & \text{otherwise,} \end{cases} \tag{4.54d}$$

with

$$\delta(l) = l \bmod (4m_k^{(a)} - 2). \tag{4.55}$$

In (4.54a)–(4.54b), $(\sigma^2)_k^{(1)}(l)$ is the noise variance at the output of the upper recursive filter, $(\sigma^2)_k^{(\text{out})}(l)$ is the total noise variance at the input to $\Upsilon_k^{(b)}(z)$ and σ_e^2 is the noise variance generated by each quantizer.

Based on (4.54a)–(4.54d), the total noise variance at any arbitrary time l (after the transient time of $\Upsilon_k^{(b)}(z)$ has elapsed) is bounded by:

$$U_1 \times (\sigma^2)_k^{(\text{out})}(l) + (2m_k^{(b)} - 1)\sigma_e^2 \tag{4.56a}$$

where

$$U_1 = \sum_{i=0}^{2m_k^{(b)}-2} \left(\frac{\zeta(i)}{\mu_k} \right)^2 \tag{4.56b}$$

For the case of the structure of Fig. 4.2, the overall noise at time l can be expressed according to Equation (4.53) by using the substitutions $\mu_k \rightarrow \nu_k$ and $\zeta_k \rightarrow 2\eta_k$. Now the noise variance of both the real and the imaginary branches of the upper branch for any arbitrary time l can be expressed as:

$$(\sigma^2)_k^{(1)}(l) = \begin{cases} 0, & \delta(l) = 4m_k^{(a)} - 3, \\ \sigma_e^2 \left[\sum_{i=0}^{\delta(l)} c(i)(\Re\{P_i\}^2 + \Im\{P_i\}^2) + \sum_{i=0}^{\delta(l)-1} \Re\{P_i\}^2 + \Im\{P_i\}^2 \right], & \\ & \text{otherwise.} \end{cases} \tag{4.57}$$

The polynomials P_i are derived from T_i defined on (4.54c) with $(\alpha_k + 1/\alpha_k) \rightarrow (\beta_k + 1/\beta_k)$. The coefficients $c(i)$ are given in (4.54d).

Based on (4.54)–(4.57), the total noise variance at any arbitrary time l (after the transient time of $\Gamma_k^{(b)}(z)$ has elapsed) is bounded by:

$$2 \left[\sum_{i=0}^{2n_k^{(b)}-2} \text{Re}^2\{c_i\} + \sum_{i=0}^{2n_k^{(b)}-2} \text{Im}^2\{c_i\} \right] \sigma_{\text{out}}^{(2)}(l) + (2n_k^{(b)} - 2)\sigma_e^2, \tag{4.58a}$$

where

$$c_i = \frac{2\eta(i)}{\nu_k}. \tag{4.58b}$$

4.3.2 Parallel Structure III

The noise analysis for the structure of Fig. 4.8 is very similar to that of Fig. 4.2. There are, however, two major differences.

- The resetting time of the upper and lower recursive filters for the structure of Fig. 4.8 are $2\rho(4n_k^{(a)} - 1) - 1$ and $(2\rho + 1)(4n_k^{(a)} - 1) - 1$ respectively.

- The number of demultiplexers are increased from two to five, and the time slot in which each multiplexer feeds each recursive filter has changed.

Considering the above facts, and using the notations of (3.59) and (4.43), the overall output noise error at any arbitrary time l for the single stage implementation of $\Omega_k^{(b)}(z)$ is given by

$$\xi_k^{(\text{ove})}(l) = \sum_{i=0}^{4n_k^{(b)}-4} 2\eta_k^{(b)}(i)\,\text{Re}\Bigg\{\nu_k^{(2)}\hat{\gamma}_k\Big[\Xi_{k,\,\text{out}}^{(1)}(l-in_k^{(a)},0,8n_k^{(a)}-2) + \Xi_{k,\,\text{out}}^{(2)}(l-in_k^{(a)},4n_k^{(a)}-1,8n_k^{(a)}-2)\Big]\Bigg\}, \quad (4.59)$$

where $\Xi_{k,\,\text{out}}^{(m)}(l,j,n)$ is defined in (4.52a) and (4.52b).

The variance of the complex output noise at the input of $\Omega_k^{(b)}(z)$ at any arbitrary time l can be expressed as

$$(\sigma^2)_{k,\,\text{real}}^{(\text{out})}(l) = (\sigma^2)_{k,\,\text{real}}^{(1)}(l) + (\sigma^2)_{k,\,\text{real}}^{(1)}(l+4n_k^{(a)}-1), \quad (4.60a)$$

and

$$(\sigma^2)_{k,\,\text{imag}}^{(\text{out})}(l) = (\sigma^2)_{k,\,\text{imag}}^{(1)}(l) + (\sigma^2)_{k,\,\text{imag}}^{(1)}(l+4n_k^{(a)}-1), \quad (4.60b)$$

with

$$\delta(l) = l \bmod (8n_k^{(a)}-2), \quad (4.60c)$$

$$(\sigma^2)_{k,\,\text{real}}^{(1)}(l) = \begin{cases} 0, & \delta(l) = 8n_k^{(a)}-3, \\ \sigma_e^2\left[\sum_{i=0}^{\delta(l)} c(i)\Re\{P_i\}^2 + \sum_{i=0}^{\delta(l)-1} \Re\{P_i\}^2 + \Im\{P_i\}^2\right], & \text{otherwise}, \end{cases} \quad (4.60d)$$

$$(\sigma^2)_{k,\,\text{imag}}^{(1)}(l) = \begin{cases} 0, & \delta(l) = 8n_k^{(a)}-3, \\ \sigma_e^2\left[\sum_{i=0}^{\delta(l)} c(i)\Im\{P_i\}^2 + \sum_{i=0}^{\delta(l)-1} \Re\{P_i\}^2 + \Im\{P_i\}^2\right], & \text{otherwise}, \end{cases} \quad (4.60e)$$

and

$$c(i) = \begin{cases} 2, & n_k^{(a)} \le \delta(l) - k \le 2n_k^{(a)} - 1 \text{ or } 6n_k^{(a)} - 1 \le \delta(l) - k \le 7n_k^{(a)} - 2, \\ 3, & 2n_k^{(a)} \le \delta(l) - k \le 3n_k^{(a)} - 1 \text{ or } \delta(l) - k = 4n_k^{(a)} - 1, \\ 4, & 3n_k^{(a)} \le \delta(l) - k \le 4n_k^{(a)} - 2 \text{ or } 4n_k^{(a)} \le \delta(l) - k \le 5n_k^{(a)} - 2, \\ 1, & \text{otherwise.} \end{cases} \tag{4.60f}$$

In (4.60a)–(B.41), $(\sigma^2)^{(1)}_{k,\,\text{real}}(l)$ and $(\sigma^2)^{(1)}_{k,\,\text{imag}}(l)$ are the noise variances at the output of the real and the imaginary branches of the upper recursive filter respectively, $(\sigma^2)^{(\text{out})}_{k,\,\text{real}}(l)$ and $(\sigma^2)^{(\text{out})}_{k,\,\text{imag}}(l)$ are the total noise variance at the input to the multiplier $\nu_k^{(2)}\widehat{\gamma}_k$ and σ_e^2 is the noise variance generated by each quantizer. The polynomials P_i are derived from T_i defined in (4.54c) with $\alpha_k + 1/\alpha_k \to \beta_k + 1/\beta_k$.

Based on (4.59)–(4.60), the total noise variance at any arbitrary time l (after the transient time of $\Omega_k^{(b)}(z)$ has elapsed) is bounded by:

$$\left(\sum_{i=0}^{4n_k^{(b)}-4} c_i^2 \right) (\sigma^2)^{(\text{out})}_{k,\Gamma}(l) + (4n_k^{(b)} - 5)\sigma_e^2, \tag{4.61a}$$

where

$$(\sigma^2)^{(\text{out})}_{k,\Gamma}(l) = (\sigma^2)^{(\text{out})}_{k,\,\text{real}}(l)\text{Re}\{\nu_k^{(2)}\widehat{\gamma}_k\}^2 + (\sigma^2)^{(\text{out})}_{k,\,\text{imag}}(l)\text{Im}\{\nu_k^{(2)}\widehat{\gamma}_k\}^2 + 2\sigma_e^2 \tag{4.61b}$$

and

$$c_i = \frac{2\eta_k^{(b)}(i)}{\nu_k^{(1)}\nu_k^{(2)}}. \tag{4.61c}$$

4.4 Performance study

This section presents the results obtained from computer simulation for the structures introduced in this chapter. The prototype IIR filter specification and the algorithm in use are already presented in Section 3.5. Several cases have been considered, and the results have been compared with other efficient implementations of linear-phase FIR filters.

4.4.1 Simulation Results

Example 1: Consider the specifications of *Example 1* on page 93, i.e., $\omega_p = 0.4\pi$, $\omega_s = 0.41\pi$, $\delta_p = 0.01$ and $\delta_s = 0.001$. Parallel Structure II requires five branches to meet the specifications, where one of the branches corresponds to $A(z)$ as given by (4.22b) with a single coefficient a_0, and four others correspond to $H_k^{(c)}(z), k = 1, 2, 3, 4$ as given by (4.22d). Applying a truncation length[3] of $N_k = 288, k = 1, 2, 3, 4$ leads to a filter structure meeting the specifications. The impulse responses of these four branches of the resulting structure is presented in Fig. 4.9. By decomposing N_k into $n_k^{(a)} = 144$ and $n_k^{(b)} = 2$, and considering the results given on Table 4.2 with recursive implementation of $\Gamma_k^{(a)}(z)$, the required delay elements, real-valued additions and real-valued multiplications for each $H_k^{(c)}(z)$ are 584, 22, and 14 respectively.

A more careful inspection of the parameters of the optimum structure reveals that $|\beta_1| \approx 0.9866, |\beta_2| \approx 0.9399, |\beta_3| \approx 0.7868$ and $|\beta_4| \approx 0.3751$. This implies that the impulse responses of the branches corresponding to β_3 and β_4 decay quite rapidly, and therefore, some of the samples of these branches can be discarded. In fact the structure meets the specifications, if the largest 269 samples of the second branch, the largest 61 samples of the third branch and the largest 19 samples of the fourth branch are preserved. As a result, the third and the fourth branches can essentially be implemented as symmetric linear-phase FIR filters of order 61 and 19 respectively.

The above observation justifies a second round of optimization, in order to find the minimum truncation length N_k for each branch separately. Doing so, it turns out that the specifications of this example is met with $N_1 = 288$, $N_2 = 80$, $N_3 = 21$ and $N_4 = 10$. Based on the results given in Table 4.2, the whole structure can be implemented using 81

[3]The minimum truncation length to meet the specifications is the prime number $N_k = 283$, which cannot be factorized and therefore is not interesting in terms of implementation.

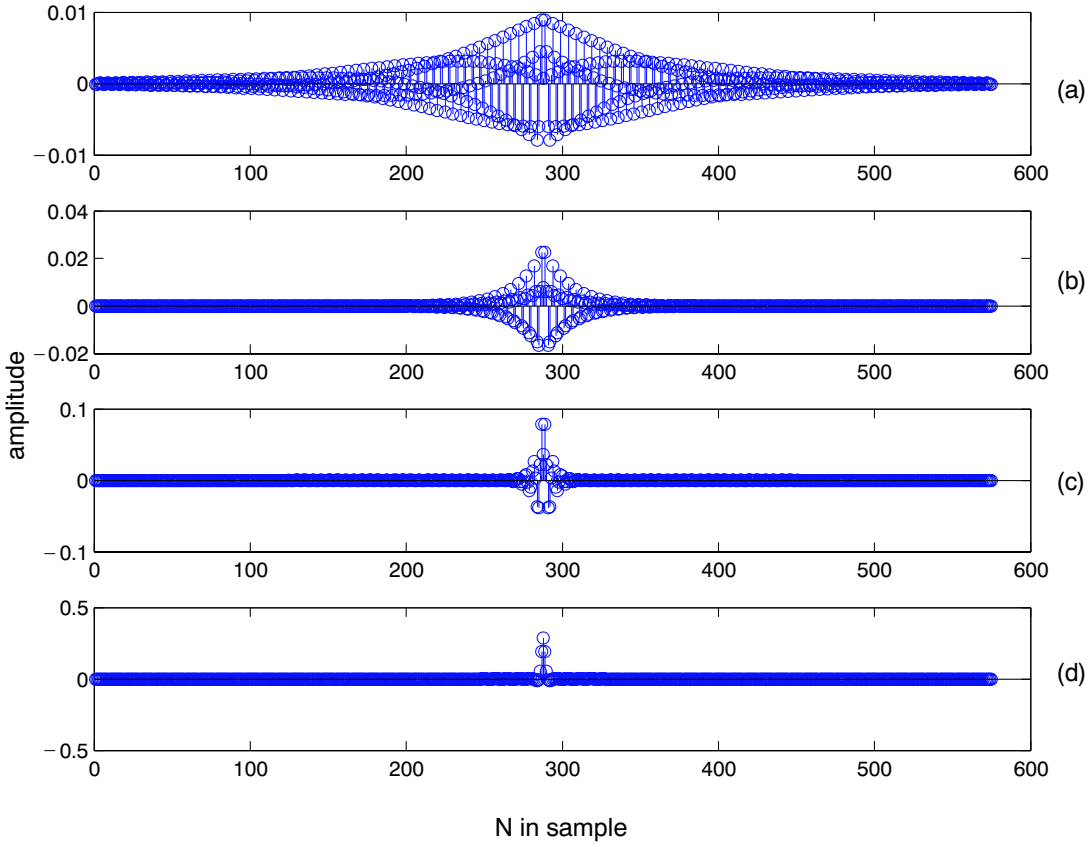

Figure 4.9: The impulse responses for the optimized Parallel Structure II for the specifications of *Example* 1. (a) First branch. (b) Second branch. (c) Third branch. (d) Fourth branch.

real-valued additions, 49 real-valued multiplications and 584 delay elements respectively. The number of real-valued additions and real-valued multiplications are almost 19% and 16% that of the optimum direct form design, while the number of delays has increased by less than 14%. The amplitude response of the proposed structure is presented in Fig. 4.10.

The proposed structure shows significant savings compared to other efficient implementation of linear-phase FIR filters. For instance, to implement the above structure using the frequency masking response technique [62, 63] a total of 154 real-valued additions, 79

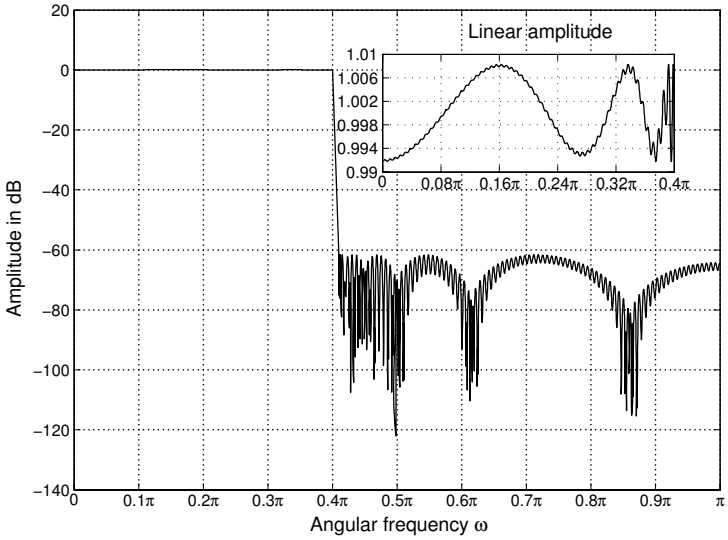

Figure 4.10: Amplitude response of the Parallel Structure II, meeting the specifications of *Example* 1.

real-valued multiplications and 564 delay elements is required. This implies that the proposed design improves the number of real-valued additions and real-valued multiplications by 47% and 38%, at the expense of 4% extra delay.

By the above setting and based on the results of Section 4.3.1, the maximum variance of the noise generated by the first and the second branches are $4.53 \times 10^6 \sigma_e^2$ and $8.13 \times 10^7 \sigma_e^2$ respectively. This implies that by assigning 13 extra bits, the variance of the noise can be lowered to that of a single quantizer.

To meet the above specifications, Parallel Structure III requires five branches, where one of the branches corresponds to $A(z)$ as given by (4.22b) with a single coefficient a_0, and four others correspond to $H_k^{(c)}(z), k = 1, 2, 3, 4$ as given by (4.40b). Applying a truncation length of $N_k = 244, k = 1, 2, 3, 4$ leads to a filter structure meeting the specifications. The impulse responses of these four branches of the resulting structure is presented in Fig. 4.11. By decomposing N_k into $n_k^{(a)} = 122$ and $n_k^{(b)} = 2$, and considering the results

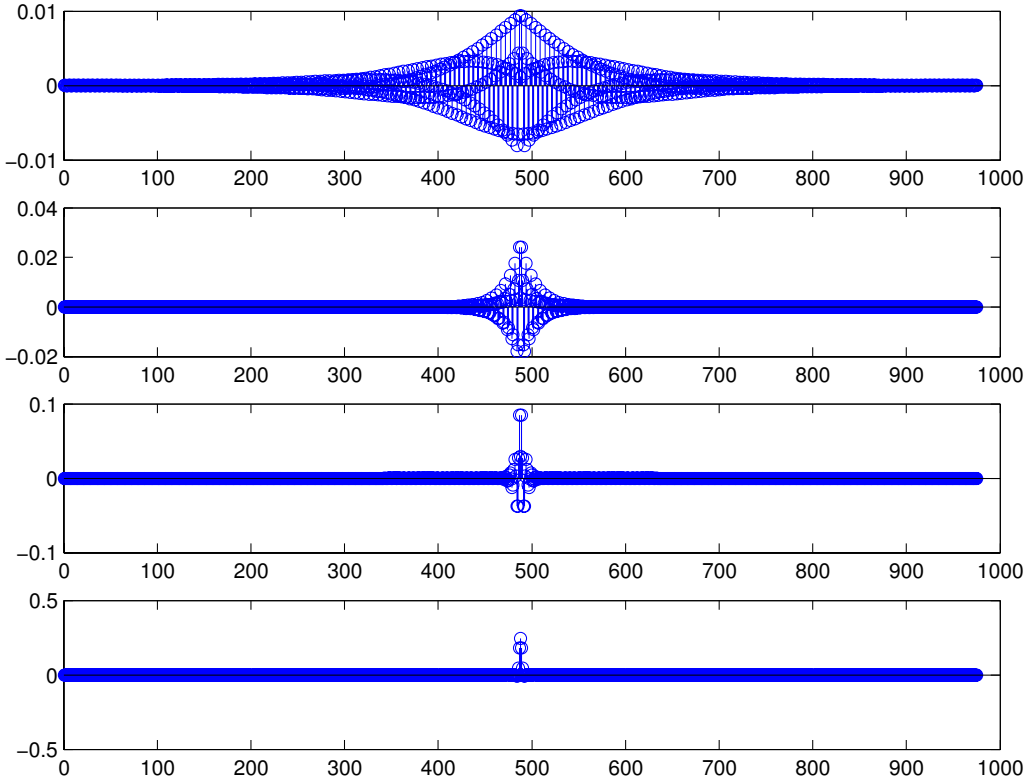

Figure 4.11: The impulse responses for the optimized Parallel Structure III for the specifications of *Example* 1. (a) First branch. (b) Second branch. (c) Third branch. (d) Fourth branch.

given on Table 4.3 with recursive implementation of $\Gamma_k^{(a)}(z)$, the required delay elements, real-valued additions and real-valued multiplications for each $H_k^{(c)}(z)$ are 984, 18, and 15 respectively.

For the optimum structure, $|\beta_1| \approx 0.9871, |\beta_2| \approx 0.9376, |\beta_3| \approx 0.7997$ and $|\beta_4| \approx 0.4207$. As for the Parallel Structure II, the impulse response of β_3 and β_4 decay rapidly, and therefore the specifications of this example can still be met with $N_1 = 244$, $N_2 = 64$, $N_3 = 18$ and $N_4 = 5$. Based on the results given in Table 4.3 with the recursive

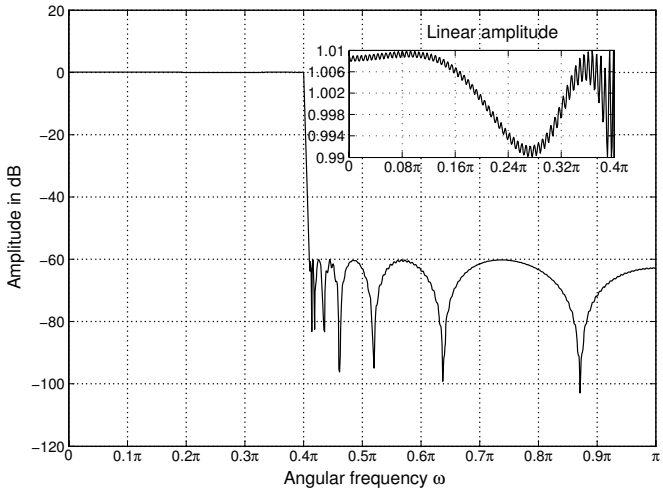

Figure 4.12: Amplitude response of the Parallel Structure III, meeting the specifications of *Example* 1.

implementation of the first and the second branches, and nonrecursive implementation of the third and the fourth branches, the whole structure can now be implemented using 78 real-valued additions, 55 real-valued multiplications and 984 delay elements respectively. The zero-phase response of the proposed structure is presented in Fig. 4.12.

By the above setting and based on the results of Section 4.3.1, the maximum variance of the noise generated by the first and the second branches are $3.91 \times 10^8 \sigma_e^2$ and $5.24 \times 10^{10} \sigma_e^2$ respectively. This implies that by assigning 18 extra bits, the variance of the noise can be lowered to that of a single quantizer.

Example 2: Consider the specifications of *Example 2* on page 93, i.e., $\omega_p = 0.4\pi$, $\omega_s = 0.402\pi$, $\delta_p = 0.01$ and $\delta_s = 0.001$. Parallel Structure III requires six branches to meet the specifications, where one of the branches corresponds to $A(z)$ as given by (4.22b) with a single coefficient a_0, and five others correspond to $H_k^{(c)}(z), k = 1, 2, \ldots, 5$ as given by (4.22d). Applying a truncation length of $N_k = 1398, k = 1, 2, \ldots, 5$ leads

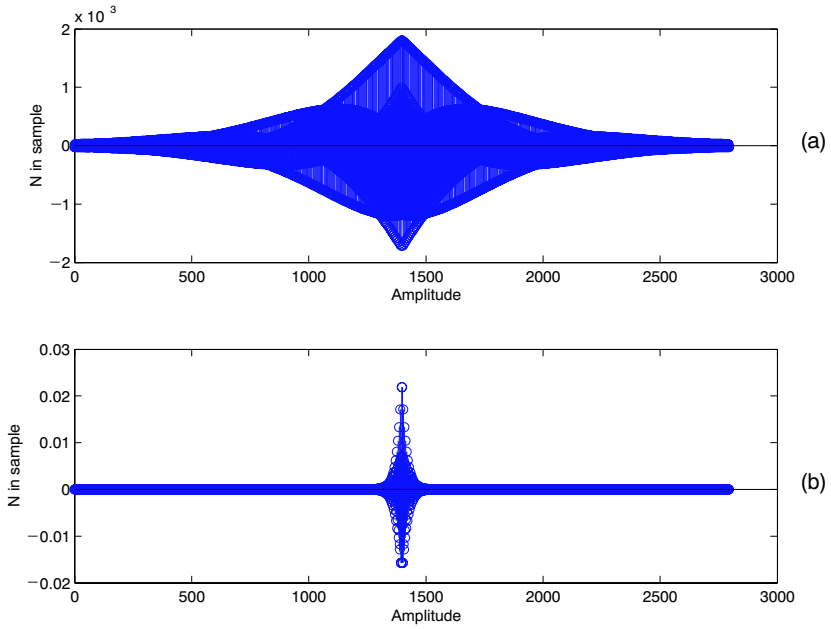

Figure 4.13: The impulse responses for the optimized Parallel Structure II for the specifications of *Example* 2. (a) First branch. (c) Third branch.

to a filter structure meeting the specifications. The impulse responses of the first and the third branches of the resulting structure is presented in Fig. 4.13. By decomposing N_k into $n_k^{(a)} = 233$, $n_k^{(1)} = 2$ and $n_k^{(2)} = 3$, and considering the results given on Table 4.2 with recursive implementation of $\Gamma_k^{(a)}(z)$, the required delay elements, real-valued additions and real-valued multiplications for each $H_k^{(c)}(z)$ are 2804, 28, and 22 respectively.

A more careful inspection of the parameters of the optimum structure reveals that $|\beta_1| \approx 0.9973$, $|\beta_2| \approx 0.9877$, $|\beta_3| \approx 0.9525$ and $|\beta_4| \approx 0.8111$ and $|\beta_5| \approx 0.4093$. This implies that the impulse responses of the branches corresponding to β_4 and β_5 decays quite rapidly, and therefore, some of the samples of these branches can be discarded. In fact the structure meets the specifications, if the largest 1417 samples of the second branch, the largest 401 samples of the third branch, the largest 87 samples of the fourth branch and the largest 23

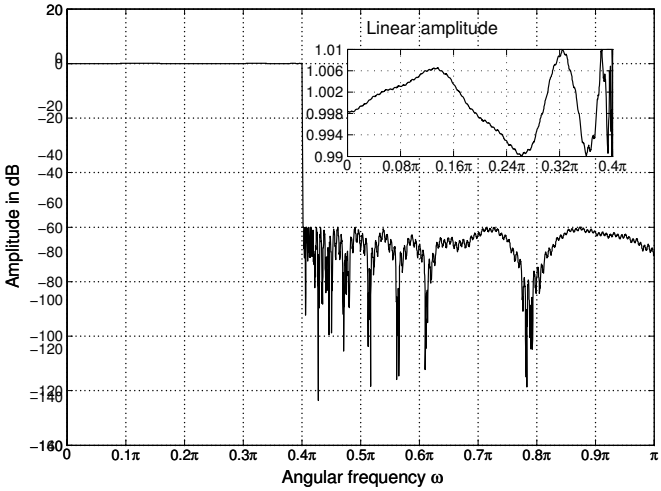

Figure 4.14: Amplitude response of the Parallel Structure II, meeting the specifications of *Example* 2.

samples of the fourth branch are preserved. As a result, the third and the fourth branches can be implemented as symmetric linear-phase FIR filters of order 87 and 23 respectively.

A second round of optimization reveals that with $N_1 = 1398$, $N_2 = 308$, $N_3 = 104$, $N_4 = 22$ and $N_5 = 7$, the specifications of this example are still met. Based on the results given in Table 4.2, the whole structure can be implemented using 115 real-valued additions, 74 real-valued multiplications and 2804 delay elements respectively. The number of real-valued additions and real-valued multiplications are almost 5% of that of the optimum direct form design, while the number of delays has increased by less than 9%. The amplitude response of the proposed structure is presented in Fig. 4.14.

The proposed structure shows considerable savings compared to other efficient implementation of linear-phase FIR filters. For instance, to implement the above structure using the frequency masking response technique [62, 63] a total of 332 real-valued additions, 168

real-valued multiplications and 2690 delay elements is required. This implies that the proposed design improves the number of real-valued additions and real-valued multiplications by 65% and 32%, at the expense of 4% extra delay.

By the above setting and based on the results of Section 4.3.1, the maximum variance of the noise $5.47 \times 10^{10}\sigma_e^2$ is generated by the third branch. This implies that by assigning 18 extra bits, the variance of the noise can be lowered to that of a single quantizer.

To meet the above specifications, Parallel Structure III also requires six branches, where one of the branches corresponds to $A(z)$ as given by (4.22b) with a single coefficient a_0, and four others correspond to $H_k^{(c)}(z), k = 1, 2, \ldots, 5$ as given by (4.40b). Applying a truncation length of $N_k = 1194, k = 1, 2, \ldots, 5$ leads to a filter structure meeting the specifications. The impulse responses of the first and the third branches of the resulting structure is presented in Fig. 4.15. By decomposing N_k into $n_k^{(a)} = 597$ and $n_k^{(b)} = 2$, and considering the results given on Table 4.3 with recursive implementation of $\Gamma_k^{(a)}(z)$, the required delay elements, real-valued additions and real-valued multiplications for each $H_k^{(c)}(z)$ are 4784, 18, and 15 respectively.

For the optimum structure, $|\beta_1| \approx 0.9971, |\beta_2| \approx 0.9854, |\beta_3| \approx 0.9441$, $|\beta_4| \approx 0.7864$ and $|\beta_5| \approx 0.4076$. As for the Parallel Structure II, the impulse response of β_4 and β_5 decay rapidly, and therefore the specifications of this example can still be met with $N_1 = 1194$, $N_2 = 246$, $N_3 = 69$, $N_4 = 25$ and $N_5 = 5$. Based on the results given in Table 4.3 with the recursive implementation of the first, second and third branches, and nonrecursive implementation of the fourth and the fifth branches, the whole structure can now be implemented using 96 real-valued additions, 67 real-valued multiplications and 4784 delay elements respectively. This implies that compared to Parallel Structure II, Parallel Structure III is even more efficient by 9% and 17% in terms of the number of real-valued multiplications and

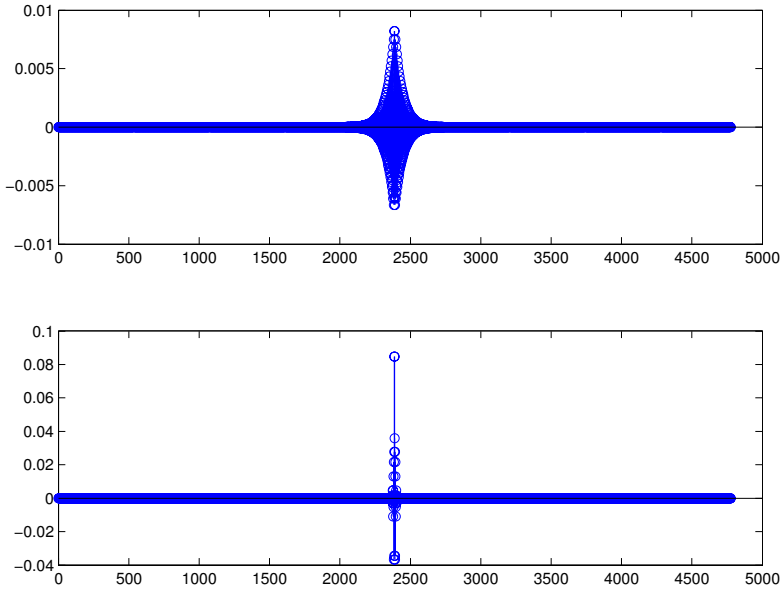

Figure 4.15: The impulse responses for the optimized Parallel Structure III for the specifications of *Example* 2. (a) First branch. (c) Third branch.

real-valued additions respectively. However, this comes at the expense of 68% extra delay elements. The amplitude response of the proposed structure is presented in Fig. 4.16.

By the above setting and based on the results of Section 4.3.1, the maximum variance of the noise generated by the first and the second branches are $3.08 \times 10^9 \sigma_e^2$ and $5.24 \times 10^{11} \sigma_e^2$ respectively. This implies that by assigning 20 extra bits, the variance of the noise can be lowered to that of a single quantizer.

Table 4.4 presents the crucial parameters of the structures proposed in this chapter for the above examples. The required number of elements and the generated noise variance for different filter structures meeting the specifications of *Example 1* and *Example 2* have been summarized in Tables 4.5 and 4.6 respectively.

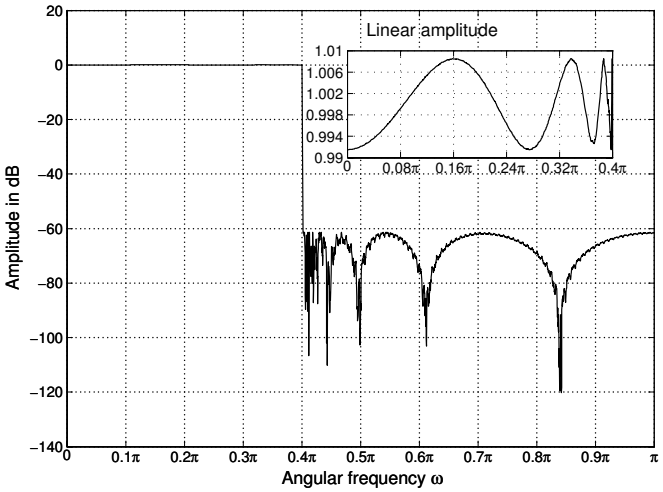

Figure 4.16: Amplitude response of the Parallel Structure III, meeting the specifications of *Example* 2.

Table 4.4: Summary of the Crucial Parameters of the Re-optimized Parallel Structures II and III for *Example 1* and *Example 2*

	Parallel Structure II	Parallel Structure III
Example 1	$\beta_1 = 0.2928 + 0.9423i, \gamma_1 = -0.0019 - 0.0085i$, $\beta_2 = 0.2896 + 0.8979i, \gamma_2 = 0.0008 - 0.0215i$ $\beta_3 = 0.2933 + 0.7548i, \gamma_3 = 0.0068 - 0.0746i$, $\beta_4 = 0.2934 + 0.3312i, \gamma_4 = 0.1149 - 0.1932i$, $a_0 = 0.0705$	$\beta_1 = 0.2941 + 0.9424i, \gamma_1 = -0.0013 - 0.0088i$, $\beta_2 = 0.2934 + 0.8909i, \gamma_2 = 0.0019 - 0.0262i$, $\beta_3 = 0.2930 + 0.7218i, \gamma_3 = 0.0020 - 0.0858i$, $\beta_4 = 0.2285 + 0.2871i, \gamma_4 = 0.0958 - 0.2319i$, 0.1145
Example 2	$\beta_1 = 0.3056 + 0.9492i, \gamma_1 = -0.0004 - 0.0020i$, $\beta_2 = 0.3038 + 0.9374i, \gamma_2 = -0.0007 - 0.0059i$ $\beta_3 = 0.2998 + 0.8953i, \gamma_3 = -0.0016 - 0.0239i$, $\beta_4 = 0.2929 + 0.7298i, \gamma_4 = 0.0070 - 0.0842i$, $\beta_5 = 0.2763 + 0.2996i, \gamma_5 = 0.1125 - 0.1948i$, $a_0 = 0.0721$	$\beta_1 = 0.3059 + 0.9493i, \gamma_1 = -0.0004 - 0.0018i$, $\beta_2 = 0.3037 + 0.9385i, \gamma_2 = -0.0009 - 0.0052i$ $\beta_3 = 0.3019 + 0.9033i, \gamma_3 = 0.0002 - 0.0195i$, $\beta_4 = 0.3002 + 0.7648i, \gamma_4 = 0.0043 - 0.0736i$, $\beta_5 = 0.2931 + 0.3379i, \gamma_5 = 0.1115 - 0.1982i$, $a_0 = 0.0762$

Table 4.5: The Required Number of Elements and the Noise Generated for Different Filter Structures Implementing Specifications of *Example 1*

Structure	Number of Required Adders	Number of Required Multipliers	Number of Required Delays	Generated Noise Variance in σ_e^2
Cascade Structure I	78	90	654	8.70×10^6
Cascade Structure II	66	77	694	3.54×10^5
Parallel Structure II	81	49	584	8.13×10^7
Parallel Structure III	78	55	984	5.24×10^{10}
Frequency Response Masking	154	79	564	–
Conventional FIR, Direct Form	513	257	513	–

Table 4.6: The Required Number of Elements and the Noise Generated for Different Filter Structures Implementing Specifications of *Example 2*

Structure	Number of Required Adders	Number of Required Multipliers	Number of Required Delays	Generated Noise Variance in σ_e^2
Cascade Structure I	81	99	4284	5.83×10^{10}
Cascade Structure II	88	103	3714	8.58×10^8
Parallel Structure II	115	74	2804	5.47×10^{10}
Parallel Structure III	96	67	4784	5.24×10^{11}
Frequency Response Masking	332	168	2690	–
Conventional FIR, Direct Form	2541	1271	2541	–

Chapter 5

Concluding Remarks

Efficient decimator and interpolator structures have been the subject of the first part of this work. The problem of designing 1S2F was shown to boil down to a systematic algorithm for solving an optimization problem. The algorithm used the fact that the filters of a 1S2F shape different parts of the amplitude response of a filter, and consequently, the amplitude response of the filter at the end of each stage of the algorithm is an indication for the sufficiency of the orders of the constituent filters. In other words, a superfluous ripple in each band can be compensated by an increase in the order of the filter taking care of that particular band. Order estimation schemes, especially helpful for narrow transition-band decimators, were developed to facilitate the algorithm.

MBD's, which are another class of efficient decimators were subsequently introduced. By spelling out the zero-phase response of an MBD, and using the result of Proposition 1, the design of these decimators was converted to an algorithm for solving an optimization problem. By combining MBD's and 1S2F's, and adding extra constraints and using multiplierless filters, very efficient decimators excelling the known designs were derived.

The presented schemes prove to be the only systematic and the most precise among the

known schemes, supported by the simulation results presented in Chapter 2.

The focus of the second and third parts of the thesis was on developing efficient structures for narrow transition-band specifications. The idea was based on different designs for approximating a zero-phase IIR filter. Four different decomposition schemes were proposed to alleviate the noise and/or reduce the required number of the components. Noise analysis for all the above-mentioned schemes, taking all the possible decompositions into account was performed.

The simulation results provided in this work clearly indicate the capabilities and potentials of the proposed schemes. The number of the real-valued multiplications required to implement the design were a fraction of that of an optimum minimax design. Moreover, compared to many designs, the proposed design proved to be superior; the design can be implemented online, its group delay is slightly higher than that of a minimax design, it can virtually be applied to any set of specifications, and it doesn't require any time reversal circuitry.

The simulations proved that the noise generated in the structure is tolerable for a number of different cases. The flexibility of the design allows the efficiency to be traded by a higher noise tolerance if needed.

To improve the design, the optimum truncation length for each constituent branch of the filter can be addressed. As already illustrated in Section 4.4.1, different branches are of different significance to the overall zero-phase of the design. The optimum length of each branch is a subject of future topic.

Appendix A

Derivation of Some Formulae in Chapter 3

A.1 Derivation of (3.17d)

$$\begin{aligned}
E_k^{(c)}(z) =& 2\Re\left\{\frac{\gamma_k\left(1-\beta_k^{N_k}z^{-N_k}\right)}{1-\beta_k z^{-1}}\right\} \text{(By (3.13c))} \\
=& \frac{\gamma_k\left(1-\beta_k^{N_k}z^{-N_k}\right)}{1-\beta_k z^{-1}} + \frac{\gamma_k^*\left(1-(\beta_k^*)^{N_k}z^{-N_k}\right)}{1-\beta_k^* z^{-1}} \\
=& \gamma_k\sum_{n=0}^{N_k-1}(\beta_k)^n z^{-n} + \gamma_k^*\sum_{n=0}^{N_k-1}(\beta_k^*)^n z^{-n} = \sum_{n=0}^{N_k-1}\left(\gamma_k(\beta_k)^n + \gamma_k^*(\beta_k^*)^n\right)z^{-n} \\
=& \sum_{n=0}^{N_k-1} 2\text{Re}\left\{\gamma_k(\beta_k)^n\right\}z^{-n}
\end{aligned}$$

A.2 Derivation of (3.24)**:**

The numerator of (3.24) is obtained by

$$\begin{aligned}
2\Re\left\{\gamma_k\left(1-\beta_k^* z^{-1}\right)\right\} &= \left(\gamma_k - \gamma_k\beta_k^* z^{-1}\right) + \left(\gamma_k^* - \gamma_k^*\beta_k z^{-1}\right) \\
&= 2\text{Re}\left(\gamma_k\right) - 2\text{Re}\left(\gamma_k\beta_k\right)z^{-1} \\
&= 2R_k\cos(\Phi_k) - 2R_k r_k\cos(\Phi_k-\phi_k)z^{-1}
\end{aligned}$$

The denominator of (3.24) follows from

$$\begin{aligned}\left(1-\beta_k z^{-1}\right)\left(1-\beta_k^* z^{-1}\right) =& 1-2\text{Re}\{\beta_k\}z^{-1}+|\beta_k|^2 z^{-2}\\ =& 1-2r_k\cos(\phi_k)z^{-1}+r_k^2 z^{-2}\end{aligned} \tag{A.1}$$

A.3 Derivation of (3.26)**:**

As shown in (A.1) above:

$$\begin{aligned}&\left(1-\beta_k^{N_k} z^{-N_k}\right)\left(1-\beta_k^{*N_k} z^{-N_k}\right) = 1-2\text{Re}\{\beta_k^{N_k}\}z^{-N_k}+\left|\beta_k^{N_k}\right|^2 z^{-2N_k}\\ &= 1-2r_k^N\cos(N_k\phi_k)z^{-N_k}+(r_k)^2 z^{-2N_k}\end{aligned} \tag{A.2}$$

Now

$$\begin{aligned}E_k^{(c)}(z) =& G_k^{(c)}(z)\left(1-\beta_k^{*N_k} z^{-N_k}\right)\left(1-\beta_k^{N_k} z^{-N_k}\right) \text{ (By (3.24) and (3.25))}\\ =& \frac{2\Re\left\{\gamma_k\left(1-\beta_k^* z^{-1}\right)\right\}\left(1-\beta_k^{N_k} z^{-N_k}\right)\left(1-\beta_k^{*N_k} z^{-N_k}\right)}{\left(1-\beta_k z^{-1}\right)\left(1-\beta_k^* z^{-1}\right)}\\ =& \frac{2\Re\left\{\gamma_k\left(1-\beta_k^* z^{-1}\right)\right\}\Omega_k(z)}{\left(1-\beta_k z^{-1}\right)\left(1-\beta_k^* z^{-1}\right)} \text{ (By (A.2) and (3.27))}\\ =& \frac{\Omega_k(z)\left\{\gamma_k\left(1-\beta_k^* z^{-1}\right)+\gamma_k^*\left(1-\beta_k z^{-1}\right)\right\}}{\left(1-\beta_k z^{-1}\right)\left(1-\beta_k^* z^{-1}\right)}\\ =& \Omega_k(z)\left[\frac{\gamma_k}{\left(1-\beta_k z^{-1}\right)}+\frac{\gamma_k^*}{\left(1-\beta_k^* z^{-1}\right)}\right]\\ =& \Omega_k(z)2\Re\left\{\frac{\gamma_k}{\left(1-\beta_k z^{-1}\right)}\right\}\end{aligned}$$

A.4 Derivation of (3.28)**:**

$$\begin{aligned}E_k^{(c)}(z) =& 2\Omega_k(z)\Re\left\{\frac{\gamma_k}{1-\beta_k z^{-1}}\right\} \text{ (By (3.26))}\\ =& (1-\beta_k^{N_k})(1-\beta_k^{*N_k})2\Re\left\{\frac{\gamma_k}{1-\beta_k z^{-1}}\right\} \text{ (By (A.2) and (3.27))}\\ =& 2\Re\left\{\gamma_k(1-\beta_k^{*N_k})\sum_{n=0}^{N_k-1}(\beta_k)^n z^{-n}\right\}\end{aligned} \tag{A.3}$$

The last equality is established by a similar argument presented in the Derivation of (3.17d). The corresponding impulse response as given by (3.28) follows directly from (A.3).

A.5 Derivation of (3.30b)

To prove the second equality, we note that

$$\begin{aligned}
&2\Re\left\{\frac{\hat{\gamma_k}}{1-\beta_k^{-1}z^{-1}}\right\} = 2\Re\left\{\frac{\gamma_k}{z^{-1}-\beta_k}\right\} \text{(By (3.30d))} \\
=&\frac{\gamma_k}{z^{-1}-\beta_k} + \frac{\gamma_k^*}{z^{-1}-\beta_k^*} \\
=&\frac{-2\text{Re}(\beta_k\gamma_k^*) + 2\text{Re}(\gamma_k)z^{-1}}{r_k^2 - 2\text{Re}(\beta_k)z^{-1}+z^{-2}} = \frac{-2r_kR_k\cos(\Phi_k-\phi_k)+2R_k\cos(\Phi_k)z^{-1}}{r_k^2-2r_k\cos(\phi_k)z^{-1}+z^{-2}} \\
=&\frac{-2R_kr_k^{-1}\cos(\Phi_k-\phi_k)+2R_kr_k^{-2}\cos(\Phi_k)z^{-1}}{1-2r_k^{-1}\cos(\phi_k)z^{-1}+r_k^{-2}z^{-2}}
\end{aligned} \tag{A.4}$$

From (3.26):

$$\begin{aligned}
E_k^{(c)}(z^{-1}) =& \left(1-2r_k^{N_k}\cos(N_k\phi_k)z^{N_k}+r_k^{2N_k}z^{2N_k}\right).2\Re\left\{\frac{\gamma_k}{1-\beta_kz}\right\} \\
=&r_k^{2N_k}z^{2N_k}\left(1-2r_k^{-N_k}\cos(N_k\phi_k)z^{-N_k}+r_k^{-2N_k}z^{-2N_k}\right)\left\{\frac{\gamma_k}{1-\beta_kz}+\frac{\gamma_k^*}{1-\beta_k^*z}\right\} \\
=&\widehat{\Omega}_k(z)z^{2N_k}\left\{\frac{\gamma_k}{1-\beta_kz}+\frac{\gamma_k^*}{1-\beta_k^*z}\right\}\text{(By (3.30c))} \\
=&\widehat{\Omega}_k(z)z^{2N_k-1}\left\{\frac{\hat{\gamma}_k}{1-\beta_k^{-1}z^{-1}}+\frac{\hat{\gamma}_k^*}{1-\beta_k^{*-1}z^{-1}}\right\}\text{(By (3.30d))} \\
=& z^{2N_k-1}\widehat{E}_k^{(c)}(z),
\end{aligned}$$

as required by (3.29) and (3.30b).

A.6 Derivation of (3.32)

In what follows, the scaling factor ν_k has been ignored. By (3.13c)

$$E_k^{(c)}(z) = 2\Re\left\{\frac{\gamma_k\left(1-\beta_k^{N_k}z^{-N_k}\right)}{1-\beta_kz^{-1}}\right\},$$

implying that to implement $E_k^{(c)}(z)$, it suffices to implement $H(z) = \frac{\gamma_k\left(1-\beta_k^{N_k}z^{-N_k}\right)}{1-\beta_kz^{-1}}$. Thus the output of $E_k^{(c)}(z)$ is equal to the output of the real branch of $H(z)$, multiplied by 2.

(This fact is also illustrated by Fig. 3.4). The impulse response of the aforementioned $H(z)$, denoted by $h[n]$ is given by

$$h[n] = \begin{cases} \gamma_k \beta_k^l, & 0 \leq l \leq N_k - 1, \\ 0, & \text{otherwise}, \end{cases}$$

as indicated by (3.32a) and (3.32b).

A.7 Derivation of (3.35b) **and** (3.36b)

For (3.35b), consider:

$$\begin{aligned}
&\alpha_k^{M_k} - z^{-M_k} = \alpha_k^{m_k^{(a)} m_k^{(b)}} - z^{-m_k^{(a)} m_k^{(b)}} \quad \text{(By (3.35a))} \\
&= \alpha_k^{m_k^{(a)} m_k^{(b)}} \left(1 - \left(\frac{1}{\alpha_k z}\right)^{m_k^{(a)} m_k^{(b)}}\right) \\
&= \alpha_k^{m_k^{(a)}} \, \alpha_k^{m_k^{(a)}\left(m_k^{(b)}-1\right)} \left(1 - \left(\frac{1}{\alpha_k z}\right)^{m_k^{(a)}}\right) \left(\sum_{l=0}^{m_k^{(b)}-1} \left(\frac{1}{\alpha_k z}\right)^{l m_k^{(a)}}\right) \text{(By (3.34))} \\
&= \left(\alpha_k^{m_k^{(a)}} - z^{-m_k^{(a)}}\right) \sum_{l=0}^{m_k^{(b)}-1} \left(\alpha_k^{m_k^{(a)}}\right)^{m_k^{(b)}-1-l} z^{-l m_k^{(a)}} \qquad \text{(A.5)}
\end{aligned}$$

Now consider (3.21c):

$$\begin{aligned}
\widehat{E}_k^{(r)}(z) &= \frac{\kappa_k \alpha_k^{-1} \left(\alpha_k^{M_k} - z^{-M_k}\right)}{1 - \frac{1}{\alpha_k} z^{-1}} \\
&= \underbrace{\frac{\kappa_k \alpha_k^{-1} \left(\alpha_k^{m_k^{(a)}} - z^{-m_k^{(a)}}\right)}{1 - \frac{1}{\alpha_k} z^{-1}}}_{\Upsilon_k^{(a)}(z)} \cdot \underbrace{\sum_{l=0}^{m_k^{(b)}-1} \left(\alpha_k^{m_k^{(a)}}\right)^{m_k^{(b)}-1-l} z^{-l m_k^{(a)}}}_{\Upsilon_k^{(b)}(z)}
\end{aligned}$$

The last equality follows from (A.5). A similar argument goes for (3.36b).

A.8 Derivation of (3.38a) **and** (3.38b)

According to (3.36c):

$$\hat{\nu_k}\Gamma_k^{(a)}(z) = \frac{\hat{\nu_k}\gamma_k\beta_k^{-1}\left(\beta_k^{n_k^{(a)}} - z^{-n_k^{(a)}}\right)}{1-\beta_k^{-1}z^{-1}} = \hat{\nu_k}\gamma_k \underbrace{\beta_k^{-1}\beta_k^{n_k^{(a)}}}_{\beta_k^{n_k^{(a)}-1}} \cdot \underbrace{\frac{1-\beta_k^{-n_k^{(a)}}z^{-n_k^{(a)}}}{1-\beta_k^{-1}z^{-1}}}_{G(z)}.$$

$g[l]$, the impulse response of $G(z)$ is given by:

$$g\,[l] = \begin{cases} \beta_k^{-l}, & 0 \le l \le n_k^{(a)} - 1, \\ 0, & \text{otherwise,} \end{cases}$$

from which the impulse response of $\hat{\nu_k}\Gamma_k^{(a)}(z)$ (Equations (3.38a) and (3.38b)) follows.

A.9 Derivation of (3.41c) **and** (3.42c)

First the validity of

$$\sum_{i=1}^{I_k}\left(m_k^{(i)}-1\right)\prod_{l=1}^{i-1}m_k^{(l)} = m_k^{(b)} - 1, \tag{A.6}$$

is proved. To this end, $\sum_{i=1}^{I_k}\left(m^{(i)}-1\right)\prod_{l=1}^{i-1}m_k^{(l)}$ is written down term by term:

$$i = I_k : \left(m_k^{(I_k)}-1\right)\prod_{l=1}^{I_k-1}m_k^{(l)} = m_k^{(I_k)}\prod_{l=1}^{I_k-1}m_k^{(l)} - \prod_{l=1}^{I_k-1}m_k^{(l)} = m_k^{(b)} - \prod_{l=1}^{I_k-1}m_k^{(l)} \text{ (By (3.41a))}$$

$$i = I_k - 1 : \left(m_k^{(I_k-1)}-1\right)\prod_{l=1}^{I_k-2}m_k^{(l)} = m_k^{(I_k-1)}\prod_{l=1}^{I_k-2}m_k^{(l)} - \prod_{l=1}^{I_k-2}m_k^{(l)} = \prod_{l=1}^{I_k-1}m_k^{(l)} - \prod_{l=1}^{I_k-2}m_k^{(l)}$$

$$\vdots$$

$$i = 1 : \left(m_k^{(1)}-1\right)\prod_{l=1}^{0}m_k^{(l)} = m_k^{(1)}\prod_{l=1}^{0}m_k^{(l)} - \prod_{l=1}^{0}m_k^{(l)} = \prod_{l=1}^{1}m_k^{(l)} - 1$$

Obviously the second term of the ith row is cancelled by the first term of $i-1$th row, and consequently the whole sum is equal to $m_k^{(b)} - 1$.

Next the validity of

$$\prod_{i=1}^{I_k} \alpha_k^{M_k^{(i)} m_k^{(i)} - M_k^{(i)}} = \alpha_k^{m_k^{(a)}\left(m_k^{(b)}-1\right)} \tag{A.7}$$

is established as follows:

$$\begin{aligned}
\prod_{i=1}^{I_k} \alpha_k^{M_k^{(i)} m_k^{(i)} - M_k^{(i)}} &= \prod_{i=1}^{I_k} \alpha_k^{\left(m_k^{(a)} \prod_{l=1}^{i-1} m_k^{(l)}\right)\left(m_k^{(i)}-1\right)} \quad \left(\text{By (3.41d), definition of } M_k^{(i)}\right) \\
&= \alpha_k^{m_k^{(a)} \sum_{i=1}^{I_k} \prod_{l=1}^{i-1} m_k^{(l)}\left(m_k^{(i)}-1\right)} \\
&= \alpha_k^{m_k^{(a)} \sum_{i=1}^{I_k} \left(m_k^{(i)}-1\right) \prod_{l=1}^{i-1} m_k^{(l)}} = \alpha_k^{m_k^{(a)}\left(m_k^{(b)}-1\right)} \text{ (By (A.6))}
\end{aligned}$$

Finally by (3.40), (3.41a) and (3.41d)

$$\sum_{l=0}^{m_k^{(b)}-1} \left(\alpha_k^{-1} z^{-1}\right)^{l m_k^{(a)}} = \prod_{i=1}^{I_k} \sum_{l=0}^{m_k^{(i)}-1} \left(\alpha_k^{-1} z^{-1}\right)^{l M_k^{(i)}}. \tag{A.8}$$

Now

$$\begin{aligned}
\Upsilon_k^{(b)}(z) &= \sum_{l=0}^{m_k^{(b)}-1} \left[\alpha_k^{m_k^{(a)}}\right]^{\left(m_k^{(b)}-1-l\right)} z^{-l m_k^{(a)}} = \alpha_k^{m_k^{(a)}\left(m_k^{(b)}-1\right)} \sum_{l=0}^{m_k^{(b)}-1} \left(\alpha_k^{-1} z^{-1}\right)^{l m_k^{(a)}} \\
&= \prod_{j=1}^{I_k} \alpha_k^{M_k^{(i)} m_k^{(i)} - M_k^{(i)}} \prod_{i=1}^{I_k} \sum_{l=0}^{m_k^{(i)}-1} \left(\alpha_k^{-1} z^{-1}\right)^{l M_k^{(i)}} \text{ (By (A.7) and (A.8))} \\
&= \prod_{i=i}^{I_k} \sum_{l=0}^{m_k^{(i)}-1} \left[\alpha_k^{M_k^{(i)}}\right]^{\left(m_k^{(i)}-1-l\right)} z^{-l M_k^{(i)}} \\
&= \prod_{i=1}^{I_k} \Upsilon_k^{(i)}(z) \text{ (By (3.41c))}
\end{aligned}$$

A.10 Derivation of (3.44a) **and** (3.44b)

Ignoring the scaling coefficients $\widehat{\upsilon}_k \left[\prod_{i=1}^{j} \varphi_k^{(i)}\right] \gamma_k$, an inductive proof (on index j, $j = 0, \cdots, J$) is provided.

1. *Inductive base:* With $j = 0$, the cascade of $\Gamma_k^{(a)}(z)$ with no $\Gamma_k^{(j)}(z)$ is implied, in other words $j = 0$ indicates the impulse response of $\Gamma_k^{(a)}(z)$. If $j = 0$, $N_k^{(1)} = n_k^{(a)}$ as given by (3.42d). Consequently, (3.44) boils down to the impulse response of $h_k(j, l)$, as given by

$$h_k(j,l) = \begin{cases} \beta_k^{(n_k^{(a)}-1-l)}, & 0 \le l \le n_k^{(a)} - 1, \\ 0, & \text{otherwise.} \end{cases} \tag{A.9}$$

   This is exactly the impulse response of $\Gamma_k^{(a)}(z)$, as given by (3.36c).

2. *Inductive hypothesis:* For any $j - 1$, we assume that the impulse response of the cascade of $\Gamma_k^{(a)}(z)$ with the first $\Gamma_k^{(i)}(z), i = 1, \cdots, j-1$ is given by

$$h_k(j-1,l) = \begin{cases} \beta_k^{(N_k^{(j)}-1-l)}, & 0 \le l \le N_k^{(j)} - 1, \\ 0, & \text{otherwise.} \end{cases} \tag{A.10}$$

3. *Inductive step:* By (3.42c), the transfer function of $\Gamma_k^{(j)}(z)$ is given by

$$\Gamma_k^{(j)}(z) = \sum_{l=0}^{n_k^{(j)}-1} (\beta_k^{N_k^{(j)}})^{n_k^{(j)}-1-l} z^{-lN_k^{(j)}}. \tag{A.11}$$

The terms $z^{-lN_k^{(j)}}$ in (A.11) indicate that there are $N_k^{(j)} - 1$ zeros between each nonzero impulse response samples of $\Gamma_k^{(j)}(z)$. This in turn implies that if the filter with the impulse response of (A.10) is placed in cascade with the filter with transfer function of (A.11), the impulse response of the composite filter is a scaled and shifted versions of (A.10). In other words, the transfer function of the composite filter $\widetilde{H}(z)$ is given by

$$\begin{aligned} \widetilde{H}(z) =& (\beta_k^{N_k^{(j)}})^{n_k^{(j)}-1} \sum_{l=0}^{N_k^{(j)}-1} \beta_k^{(N_k^{(j)}-1-l)} z^{-l} + (\beta_k^{N_k^{(j)}})^{n_k^{(j)}-2} z^{-N_k^{(j)}} \sum_{l=0}^{N_k^{(j)}-1} \beta_k^{(N_k^{(j)}-1-l)} z^{-l} + \cdots \\ &+ z^{-N_k^{(j)}(n_k^{(j)}-1)} \sum_{l=0}^{N_k^{(j)}-1} \beta_k^{(N_k^{(j)}-1-l)} z^{-l}, \end{aligned} \tag{A.12}$$

but (3.42d) implies $N_k^{(j+1)} = N_k^{(j)} n_k^{(j)}$ and therefore (A.12) can be expressed as

$$\widetilde{H}(z) = \sum_{l=0}^{N_k^{(j)}-1} \beta_k^{(N_k^{(j+1)}-1-l)} z^{-l} + \sum_{l=0}^{N_k^{(j)}-1} \beta_k^{(N_k^{(j+1)}-N_k^{(j)}-1-l)} z^{-(l+N_k^{(j)})} + \cdots + \sum_{l=0}^{N_k^{(j)}-1} \beta_k^{(N_k^{(j)}-1-l)} z^{-(l+N_k^{(j+1)}-N_k^{(j)})}. \tag{A.13}$$

Finally changing the variables in the summation indices in (A.13) yields

$$\begin{aligned}\widetilde{H}(z) = &\sum_{l=0}^{N_k^{(j)}-1} \beta_k^{(N_k^{(j+1)}-1-l)} z^{-l} + \sum_{l=N_k^{(j)}}^{2N_k^{(j)}-1} \beta_k^{(N_k^{(j+1)}-1-l)} z^{-l} + \cdots \\ &+ \sum_{l=N_k^{(j+1)}-N_k^{(j)}}^{N_k^{(j+1)}-1} \beta_k^{(N_k^{(j+1)}-1-l)} z^{-l} \\ = &\sum_{l=0}^{N_k^{(j+1)}-1} \beta_k^{(N_k^{(j+1)}-1-l)} z^{-l},\end{aligned}$$

from which the impulse responses of (3.44) follows.

A.11 Derivation of (3.45a)

Starting from (3.35c) and ignoring the scaling term $\widehat{\mu}_k$, we have:

$$\begin{aligned}\Upsilon_k^{(a)}(z) = &\frac{\kappa_k \alpha_k^{-1} \left(\alpha_k^{m_k^{(a)}} - z^{-m_k^{(a)}} \right)}{1 - 1/\alpha_k z^{-1}} \\ = &\kappa_k \alpha_k^{m_k^{(a)}-1} \frac{\left(1 - \alpha_k^{-m_k^{(a)}} z^{-m_k^{(a)}} \right)}{1 - \alpha_k^{-1} z^{-1}} = \kappa_k \alpha_k^{m_k^{(a)}-1} \sum_{l=0}^{m_k^{(a)}-1} \left(\alpha_k^{-1} z^{-1} \right)^l \\ = &\sum_{l=0}^{m_k^{(a)}-1} \kappa_k \alpha_k^{m_k^{(a)}-1-l} z^{-l}\end{aligned}$$

A.12 Derivation of (3.49a) **and** (3.49b)

The impulse response of $E_k^{(c)}$ is given by (3.28). Equations (3.49a) and (3.49b) are valid representations of the impulse response of the part prior to the multiplier γ_k/ν_k, as seen

from Fig. 3.11. The reason is that (taking the scaling factor ν_k into account), multiplying (3.49a) and (3.49b) by γ_k/ν_k yields (3.28), as required.

A.13 Derivation of (3.50)**–**(3.51)

Let $x = \left(r_k e^{j\phi_k} z\right)^{-1}$. Now by (3.35a)

$$\frac{1-x^{N_k}}{1-x^{n_k^{(a)}}} = \sum_{l=0}^{n_k^{(b)}-1} \left(x^{n_k^{(a)}}\right)^l = \sum_{l=0}^{n_k^{(b)}-1} \left(r_k e^{j\phi_k} z\right)^{-n_k^{(a)} l}, \tag{A.14}$$

in other words, $\left(1 - r_k^{-N_k} e^{-jN_k\phi_k} z^{-N_k}\right)$ is divisible by $\left(1 - r_k^{-n_k^{(a)}} e^{-jn_k^{(a)}\phi_k} z^{-n_k^{(a)}}\right)$. By the same argument, $\left(1 - r_k^{-N_k} e^{jN_k\phi_k} z^{-N_k}\right)$ is divisible by $\left(1 - r_k^{-n_k^{(a)}} e^{jn_k^{(a)}\phi_k} z^{-n_k^{(a)}}\right)$ and the quotient is

$$\sum_{l=0}^{n_k^{(b)}-1} \left(r_k e^{-j\phi_k} z\right)^{-n_k^{(a)} l}. \tag{A.15}$$

On the other hand, $\widehat{\Omega}_k(z)$ as given by (3.30c) can be factorized as:

$$\begin{aligned}\widehat{\Omega}_k(z) =& r_k^{2N_k}\left(1 - 2r_k^{-N_k}\cos(N_k\phi_k) z^{-N_k} + r_k^{-2N_k} z^{-2N_k}\right)\\ =& r_k^{2N_k}\left(1 - r_k^{-N_k} e^{-jN_k\phi_k} z^{-N_k}\right)\left(1 - r_k^{-N_k} e^{jN_k\phi_k} z^{-N_k}\right)\end{aligned} \tag{A.16}$$

and similarly, $\widehat{\Omega}_k^{(a)}(z)$ as given by (3.51b) can be factorized as:

$$\widehat{\Omega}_k^{(a)}(z) = r_k^{2n_k^{(a)}}\left(1 - r_k^{-n_k^{(a)}} e^{-jn_k^{(a)}\phi_k} z^{-n_k^{(a)}}\right)\left(1 - r_k^{-n_k^{(a)}} e^{jn_k^{(a)}\phi_k} z^{-n_k^{(a)}}\right). \tag{A.17}$$

It follows from (A.14), (A.15),(A.16), and (A.17), that $\widehat{\Omega}_k(z)$ is divisible by $\widehat{\Omega}_k^{(a)}(z)$:

$$\frac{\widehat{\Omega}_k^N}{\widehat{\Omega}_k^{n_k^{(a)}}} = \frac{r_k^{2N}}{r_k^{2n_k^{(a)}}} G_k(z) = r_k^{n_k^{(a)}\left(2n_k^{(b)}-2\right)} G_k(z), \tag{A.18}$$

where

$$G_k(z) = \sum_{l=0}^{n_k^{(b)}-1} \left(e^{-j\phi_k n_k^{(a)}} \left(r_k z\right)^{-n_k^{(a)}}\right)^l \sum_{l=0}^{n_k^{(b)}-1} \left(e^{j\phi_k n_k^{(a)}} \left(r_k z\right)^{-n_k^{(a)}}\right)^l. \tag{A.19}$$

But $G_k(z)$ according to my notes on (4.32)–(4.34e) for the parallel case (with $e^{j\phi_k n_k^{(a)}} \rightarrow \alpha_k$ and $r_k z \rightarrow z$) can be expressed as

$$\widetilde{\Omega}_k^{(b)}(z) = \sum_{l=0}^{2n_k^{(b)}-2} \tilde{\eta}_k(l) r^{-n_k^{(a)} l} z^{-l n_k^{(a)}} \tag{A.20}$$

where

$$\tilde{\eta}_k(l) = \begin{cases} \dfrac{\left(e^{j\phi_k n_k^{(a)}}\right)^{l+1} - \left(e^{-j\phi_k n_k^{(a)}}\right)^{l+1}}{\left(e^{j\phi_k n_k^{(a)}}\right) - \left(e^{j\phi_k n_k^{(a)}}\right)}, & 0 \le l \le n_k^{(b)} - 1, \\ \tilde{\eta}_k(2n_k^{(b)} - 2 - l), & n_k^{(b)} \le l \le 2n_k^{(b)} - 2. \end{cases} \tag{A.21}$$

Since $2j\sin(\alpha) = e^{j\alpha} - e^{-j\alpha}$, (A.21) can also be expressed as

$$\tilde{\eta}_k(l) = \begin{cases} \dfrac{\sin\left[(l+1)n_k^{(a)}\phi_k\right]^{l+1}}{\sin\left[n_k^{(a)}\phi_k\right]}, & 0 \le l \le n_k^{(b)} - 1, \\ \dfrac{\sin\left[\left(2n_k^{(b)} - 1 - l\right) n_k^{(a)}\phi_k\right]^{l+1}}{\sin\left[n_k^{(a)}\phi_k\right]}, & n_k^{(b)} \le l \le 2n_k^{(b)} - 2. \end{cases} \tag{A.22}$$

Incorporating $r^{-n_k^{(a)} l}$ in (A.20) and $r_k^{n_k^{(a)}\left(2n_k^{(b)}-2\right)}$ in (A.18) into $\tilde{\eta}_k(l)$ in (A.22) yields $\eta_k(l) = r_k^{n_k^{(a)}\left(2n_k^{(b)}-2-l\right)} \tilde{\eta}_k(l)$, which is exactly what is given in (3.51d).

A.14 Derivation of (3.54a) **and** (3.54b)

The transfer function of $H_k(z)$ is (see the left boxed area of Fig. 3.12, prior to the premultiplier $\widehat{\nu}_k^{(2)}\widehat{\gamma}_k$)

$$H_k(z) = \widehat{\nu}_k^{(1)}\widehat{\Omega}_k^{(a)}(z)\Gamma_k^{(a)}(z)/\widehat{\gamma}_k = \nu_k^{(1)}\widehat{\Omega}_k^{(a)}(z)\frac{1}{1-\beta_k^{-1}z^{-1}}. \text{ (By (3.52b))} \tag{A.23}$$

$\Omega_k^{(a)}(z)$ as given by (A.17) can be alternatively expressed as:

$$\Omega_k^{(a)}(z) = (\beta_k\beta_k^*)^{n_k^{(a)}} \left(1 - \beta_k^{-n_k^{(a)}} z^{-n_k^{(a)}}\right)\left(1 - \beta_k^{*-n_k^{(a)}} z^{-n_k^{(a)}}\right). \tag{A.24}$$

Plugging (A.24) in (A.23) yields

$$\begin{aligned} H_k(z) =& \widehat{\nu}_k^{(1)} (\beta_k \beta_k^*)^{n_k^{(a)}} \left(1 - \beta_k^{*-n_k^{(a)}} z^{-n_k^{(a)}}\right) \frac{\left(1 - \beta_k^{-n_k^{(a)}} z^{-n_k^{(a)}}\right)}{1 - \beta_k^{-1} z^{-1}} \\ =& \widehat{\nu}_k^{(1)} (\beta_k \beta_k^*)^{n_k^{(a)}} \left(1 - \beta_k^{*-n_k^{(a)}} z^{-n_k^{(a)}}\right) \sum_{l=0}^{n_k^{(a)}-1} {\beta_k}^{-l} z^{-l} \\ =& \left(\widehat{\nu}_k^{(1)} \left(\beta_k \beta_k^*\right)^{n_k^{(a)}} - {\beta_k}^{n_k^{(a)}} z^{-n_k^{(a)}}\right) \sum_{l=0}^{n_k^{(a)}-1} {\beta_k}^{-l} z^{-l} \end{aligned} \tag{A.25}$$

The corresponding impulse responses as given by (3.54a) and (3.54b) follow directly from the above equation.

A.15 Derivation of (3.56b) **and** (3.56d)

Based on the definition of $N_k^{(j)}$ (as given by (3.56c)),

$$N_k^{(j+1)} = n_k^{(a)} \prod_{l=1}^{j} n_k^{(l)} = \left(n_k^{(a)} \prod_{l=1}^{j-1} n_k^{(l)}\right) n_k^{(j)} = N_k^{(j)} n_k^{(j)}.$$

This implies that (3.56d) is another version of (3.51d), with $N_k \rightarrow N_k^{(j+1)}, n_k^{(a)} \rightarrow N_k^{(j)}$ and $n_k^{(b)} \rightarrow n_k^{(j)}$.

Finally by the same reasoning presented in the Derivation of (3.50)–(3.52), decomposibility of N_k as given by (3.50), (3.55), and (3.56d) leads to a corresponding decomposibility of $\Omega_k^{(b)}(z)$, as given by (3.56a) and (3.56b). For instance, assume

$$\begin{aligned} N_k =& n_k^{(a)} n_k^{(1)} n_k^{(2)} \\ =& N_k^{(1)} n_k^{(2)} \quad \text{By (3.56c)} \end{aligned} \tag{A.26}$$

Since $N_k^{(1)}$ divides N_k, then by (A.18) and (A.19), $\widehat{\Omega}_k^{(a,1)}(z)$ divides $\widehat{\Omega}_k(z)$, where

$$\widehat{\Omega}_k^{(a,1)}(z) = r_k^{2N_k^{(1)}} \left(1 - r_k^{-N_k^{(1)}} e^{-jN_k^{(1)}\phi_k} z^{-N_k^{(1)}}\right) \left(1 - r_k^{-N_k^{(1)}} e^{jN_k^{(1)}\phi_k} z^{-N_k^{(1)}}\right).$$

As proven above, the transfer function of the quotient, denoted by $\widehat{\Omega}_k^{(2)}(z)$, is in the form of (3.56b). But by the same argument, $\widehat{\Omega}_k^{(a,1)}(z)$ is divisible by $\widehat{\Omega}_k^{(a)}(z)$ as given by (3.51b), and the quotient, denoted by $\widehat{\Omega}_k^{(1)}(z)$, is in the form of (3.56b). In other words, it has been shown that

$$\widehat{\Omega}_k(z) = \widehat{\Omega}_k^{(a)}(z)\widehat{\Omega}_k^{(1)}(z)\widehat{\Omega}_k^{(2)}(z)$$

as required.

A.16 Derivation of (3.58a)

Factorizing $n_k^{(a)}$ as $n_k^{(a)} = 1.n_k^{(a)}$ and applying the decomposition of (3.51a) to $\widehat{\Omega}_k^{(a)}(z)$ as given by (3.51a) yields

$$\widehat{\Omega}_k^{(a)}(z) = \widetilde{\Omega}_k^{(a)}(z)\widetilde{\Omega}_k^{(b)}(z) \tag{A.27}$$

where

$$\begin{aligned}\widetilde{\Omega}_k^{(a)} =& r_k^2 - 2r_k\cos(n\phi_k)z^{-1} + z^{-2}\\ =& r_k^2\left(1 - r_k^{-1}e^{-j\phi_k}z^{-1}\right)\left(1 - r_k^{-1}e^{j\phi_k}z^{-1}\right)\\ =& r_k^2\left(1 - \beta_k^{-1}z^{-1}\right)\left(1 - \beta_k^{*-1}z^{-1}\right),\end{aligned} \tag{A.28}$$

and

$$\widetilde{\Omega}_k^{(b)}(z) = \sum_{l=0}^{2n_k^{(a)}-2}\eta_k^{(a)}(l)z^{-l}, \tag{A.29}$$

with $\eta_k^{(a)}(l)$ defined in (3.58b).

Now

$$\begin{aligned}\widehat{\Omega}_k^{(a)}(z)\Re\left\{\Gamma_k^{(a)}(z)\right\} =& \widetilde{\Omega}_k^{(b)}(z)\widetilde{\Omega}_k^{(a)}(z)\Re\left\{\Gamma_k^{(a)}(z)\right\} \text{ (By (A.27))}\\ =& r_k^2\sum_{l=0}^{2n_k^{(a)}-2}\eta_k^{(a)}(l)z^{-l}\Re\left\{\widehat{\gamma}_k\left(1 - \beta_k^{*-1}\right)\right\}\\ & \text{(By (A.28), (A.29) and (3.52b))}\end{aligned}$$

from which (3.58a) follows.

A.17 Derivation of (3.58b)

Follows from (3.50) and (3.51d), with $N_k \rightarrow n_k^{(a)}, n_k^{(a)} \rightarrow 1$ and $n_k^{(b)} \rightarrow n_k^{(a)}$.

A.18 Derivation of (3.62a) **and** (3.62b)

To appreciate these equations, an example is in order. Assuming $m_k^{(a)}$=3 and $m_k^{(b)}$=2, by (3.35c), the order of $\Upsilon_k^{(a)}(z)$ is found to be $m_k^{(a)} - 1 = 2$. This implies that the input signal $x(n)$ is parsed into blocks of length $m_k^{(a)} = 3$.

First, the upper branch of the structure of Figs. 3.6 and 3.7 is considered. The upper demultiplexer has been denoted by Up and the lower one by Lo. The following table indicates the samples $x(n)$ fed to the upper branch at each time instant n. The table also signifies which demultiplexer feeds the sample in question.

Time n	0	1	2	3	4	5	6	7	8	9	10	11	12	...
Up	$x(0)$	$x(1)$	$x(2)$	–	–	–	$x(6)$	$x(7)$	$x(8)$	–	–	–	$x(12)$	$x(13)$
Lo	–	–	–	$x(0)$	$x(1)$	$x(2)$	–	–	–	$x(6)$	$x(7)$	$x(8)$	–	–

Table A.1: The flow of the input samples in the upper branch of the structure of Figs. 3.6 and 3.7.

Obviously the upper branch is in charge of filtering the samples $x(0), x(1), x(2), x(6), x(7), x(8), x(12), x(13), x(14), \ldots$, and the lower branch is in charge of filtering the samples $x(3), x(4), x(5), x(9), x(10), x(11), x(15), x(16), x(17), \ldots$. The time instance $n = 5$ is very important, since then, the output will be computed, and at the same time, the memory elements will be set to zero before the beginning of a new cycle, that is, the time instance $n = 6$. Note that ideally, the output at time $n = 5$ should be zero, since the order of the FIR filter $\Upsilon_k^{(a)}(z)$ is 2. Filtering of 3 samples (such as $x(0), x(1), x(2)$) by such filter must yield zero at $n = 5$. That explains why the memory (delay) elements are set to zero

at such time instant.

Based on the above-mentioned example, the effects of the recursive filter on the input noise $x(0) \equiv \Xi_{k,\text{in}}^{(1)}(0)$ (as given by (3.59a)) is considered next. As seen from Fig. 3.6, $x(0)$ is filtered by the recursive IIR filter

$$H(z) = \frac{1}{1-(1/\beta_k)z^{-1}}, \tag{A.30}$$

which defines the difference equation

$$y[n] = x[n] + (1/\beta_k)\, y[n-1]. \tag{A.31}$$

Based on (A.38), for this noise input sample ($x(0)$), the output of the recursive filter of the upper branch $\Xi_{k,\text{out}}^{(1)}$ between the time instants $n = 0$ and $n = 5$ is given in Table A.2.

n	0	1	2	3	4	5 (resetting time)
$\Xi_{k,\text{out}}^{(1)}$	$x(0)$	$(1/\beta_k)\, x(0)$	$\left(1/\beta_k^2\right) x(0)$	$\left(1/\beta_k^3\right) x(0)$	$\left(1/\beta_k^4\right) x(0)$	0

Table A.2: The output of the recursive filter of the upper branch for $n = 0 \cdots n = 5$.

As noted from Table A.2, at time $n = 5$, the filter is reset, implying that the cumulative (recursive) effect of the filter on the noise sample $\Xi_{k,\text{in}}^{(1)}(0)$ has been removed (in fact this cancellation applies to other noises in the filter, that is, $\Xi_{k,\text{in}}^{(1)}(1)$ and $\Xi_{k,\text{in}}^{(1)}(2)$. Note that $\Xi_{k,\text{in}}^{(1)}(3)$, $\Xi_{k,\text{in}}^{(1)}(4)$ or $\Xi_{k,\text{in}}^{(1)}(5)$ are not considered here, since the upper demultiplexer is feeding the lower recursive filter at those time instances).

The next cycle starts at $n = 6$ and ends at $n = 11$, following exactly the same pattern as that for $n = 0$ up to $n = 5$. The output noise is zero at $n = 5$, $n = 11$, etc, justifying (3.60a). Compared to the input noise, the output noise is equally amplified at $n = 0$ and $n = 6$, $n = 1$ and $n = 7$, etc. This justifies the periodicity of (3.60b), and also explains

why only the time instances $\lambda \equiv n$, $0 \leq \lambda \leq 2n_k^{(a)} - 2$ are considered. (In the above table, this corresponds to the time instances $0 \ldots 4$).

Finally the validity of (3.60b) for each cycle defined above is illustrated. The impulse response of the $H(z)$ as given by (A.30) is

$$h(n) = (1/\beta_k)^n, \quad n \geq 0. \tag{A.32}$$

The output noise is the convolution sum of the input noise $\Xi_{k,\text{in}}^{(1)}(2\rho n_k^{(a)} + n)$ with (A.32), which is exactly (3.60b) for the cycle $0 \leq i \leq n$ (note that in (3.60b), $\lambda \equiv n$). In other words, (3.60b) can be interpreted as multiplying the latest input noise $\Xi_{k,\text{in}}^{(1)}(2\rho n_k^{(a)} + \lambda)$ by

$$(1/\beta_k)^0 = 1,$$

the previous one $i = \lambda - 1 \rightarrow \Xi_{k,\text{in}}^{(1)}(2\rho n_k^{(a)} + \lambda - 1)$ by

$$(1/\beta_k)^1 = 1/\beta_k,$$

still the previous one $i = \lambda - 2 \rightarrow \Xi_{k,\text{in}}^{(1)}(2\rho n_k^{(a)} + \lambda - 2)$ by

$$(1/\beta_k)^2 = 1/\beta_k^2,$$

etc.

By considering the fact that the lower branch lags the upper branch by $n_k^{(a)}$, (3.61a) and (3.61b) follow from (3.60a) and (3.60b).

A.19 Derivation of (3.62b)

The cyclic nature of the output noise indicates that a "mod" operator can facilitate the noise calculation. In the sequel, only the upper branch is considered.

By (3.60a), $\Xi_{k,\text{out}}^{(1)}(l) \equiv \Xi_{k,\text{out}}^{(1)}(2\rho n_k^{(a)} + \lambda)$, therefore $\lambda = l \bmod (2n_k^{(a)}) \equiv (l - j) \bmod (n)$, with $j \equiv 0$, $n \equiv 2n_k^{(a)}$.

By (3.61a), $\Xi^{(1)}_{k,\text{out}}(l) \equiv \Xi^{(1)}_{k,\text{out}}((2\rho+1)n^{(a)}_k + \lambda)$, therefore $\lambda = (l - n^{(a)}_k) \bmod (2n^{(a)}_k) \equiv (l-j) \bmod (n)$ with $j \equiv n^{(a)}_k$, $n \equiv 2n^{(a)}_k$.

In (3.62b), the variable "j" denotes the lag between the upper and lower branches, in other words, if the upper branch is at time l of the above-mentioned noise cycle, the lower branch is at time $l - n^{(a)}_k$ of the noise cycle. Using j, a single symbol $\Xi^{(m)}_{k,\text{out}}(l, j, n)$ encompasses the noise cycle in both the upper and the lower branches.

In other words, to obtain the overall (cyclic) noise of the structure of Fig. 3.7, the time instant (l), the lag (j) between the upper and the lower branches, and the base of the $\bmod$ operation (n) are needed, and moreover, the branch in question (m) should be specified. The function $\Xi^{(m)}_k(l, j, n)$ contains all these features effectively, and when convolved by the input noise as governed by the modular convolution of (3.62b), yields the overall output noise.

A.20 Derivation of (3.62a)

In words, (3.62a) states that "add the output noises of the first and the second branches, $\Xi^{(1)}_{k,\text{out}}(l)+\Xi^{(2)}_{k,\text{out}}(l)$, using the notation of (3.62b), and convolve that by $\Upsilon^{(b)}_k(z)/\widehat{\mu}_k$, as shown in Fig. 3.7".

A.21 Derivation of (3.63a)**–**(3.64b) **for Fig. 3.7**

Taking the upper branch into account, two separate noise sources in the structure are distinguished, i.e., the noise generated by the multipliers before the demultiplexers, and those generated by the upper recursive filter coefficients.

The noise generated by the multipliers before the demultiplexers

Denoting the input noise sample to the upper branch at time instant i by e_i, the output noise $\Xi^{(1)}_{k,\text{out}}(n)$ of the upper branch of Fig. 3.7 is presented in Table A.3.

n	0	1	2	...	$2m_k^{(a)} - 1$
$\Xi_{k,\text{out}}^{(1)}(n)$	$E_0 = e_0$	$E_1 = e_1 + (1/\alpha_k)\, E_0$ $= e_1 + (1/\alpha_k)\, e_0$	$E_2 = e_2 + (1/\alpha_k)\, E_1$ $=(1/\alpha_k)^2\, e_0 + (1/\alpha_k)\, e_1 + e_2$	...	0 (=resetting time)

Table A.3: The output noise of the upper branch of Fig. 3.7, generated by the multipliers before the demultiplexers.

Table A.3 is in fact a generalization of Table A.2, where all the input samples (instead of only e_0) are considered, and is therefore governed by the same difference equation ((A.32)). In other words,

$$E_n = \sum_{i=0}^{n} T_{n-i} e_i, \quad 0 \leq n \leq 2m_k^{(a)} - 2, \tag{A.33}$$

where the coefficients T_n are defined recursively as:

$$T_0 = 1, \text{ and } T_n = 1/\alpha_k T_{n-1}, \tag{A.34}$$

as indicated by the transfer function of the recursive filter $H(z)$, defined in (A.30). Equation (A.35) can alternatively be expressed as

$$E_n = \sum_{i=0}^{n} (1/\alpha_k)^n\, e_i, \quad 0 \leq n \leq 2m_k^{(a)} - 2. \tag{A.35}$$

The noise generated by the upper recursive filter

The noise generated by the upper recursive filter lags that generated by the multipliers before the demultiplexers by one sample (since it takes one multiplication operation before this noise is generated in the recursive part of the filter), but otherwise, it is subject to the same recursion. The generation of noise originated by the upper recursive filter is illustrated in Table A.4.

n	0	1	2	3	...	$2m_k^{(a)}-1$
$\Xi_{k,\text{out}}^{(1)}(n)$	0 (=zero still in the memory)	$F_1 = f_1$	$F_2 = f_2 + (1/\alpha_k) F_1 = f_2 + (1/\alpha_k) f_1$	$F_3 = f_3 + (1/\alpha_k) F_2 = f_3 + (1/\alpha_k) f_2 + (1/\alpha_k)^2 f_1$	...	0 (=resetting time)

Table A.4: Generation of noise originated by the upper recursive filter in Fig 3.7.

Comparing Table A.4 to Table A.3 reveals that $F_n = \sum_{i=1}^{n} T_{n-i} f_i$, where T_i is defined in (A.34). As seen above, the lower bound of index i lags that of E_n by 1.

n	1	2	3	4	5	...	$2\rho n_k^{(a)}$
	0 (=zero in the memory)	f_1	$f_2 + \frac{1}{\beta_k} f_1$	$f_3 + \frac{1}{\beta_k} f_2 + \frac{1}{\beta_k^2} f_1$	$f_4 + \frac{1}{\beta_k} f_3 + \frac{1}{\beta_k^2} f_2 + \frac{1}{\beta_k^3} f_1$	...	0 (=resetting time)

The total noise

Considering the independency of $f_i(n)$ and $e_i(n)$, and appreciating the fact that the noise variances of both f_i and e_i is σ_e^2, the total variance of the noise generated at time n in the upper branch is:

$$V_n = \left[\underbrace{\sum_{i=0}^{n} T_{n-i}^2}_{\text{The part related to } E_i} + \underbrace{\sum_{i=1}^{n} T_{n-i}^2}_{\text{The part related to } F_i} \right] \sigma_e^2$$

$$\overset{k=n-i}{=} \left[\underbrace{\sum_{k=0}^{n} T_k^2}_{\text{The part related to } E_i} + \underbrace{\sum_{k=0}^{n-1} T_k^2}_{\text{The part related to } F_i} \right] \sigma_e^2,$$

$$= \left[\sum_{k=0}^{n-1} (1/\alpha_k)^2 + \sum_{k=0}^{n} (1/\alpha_k)^2 \right] \sigma_e^2, \tag{A.36}$$

where $\sigma_e^2 = \frac{2^{-2b}}{12}$ is the variance of a single noise source.

The geometric series of (A.36) can be written in the following closed form:

$$V_n = \left[2 \sum_{i=0}^{n-1} \frac{1}{\alpha_k^{2i}} - \frac{1}{\alpha_k^{2n-2}} \right] \sigma_e^2 = \left[2 \frac{1 - \alpha_k^{-2n}}{\alpha_k^{-2}} - \frac{1}{\alpha_k^{2n-2}} \right] \sigma_e^2. \tag{A.37}$$

The output of $\Upsilon_k^{(a)}(z)$

Considering the fact that the variance of the noise at each branch is periodic with the period of $2m_k^{(a)}$, and also the fact that the variance of the noise at the time instant $n + m_k^{(a)}$ for the lower branch is equal to that of the upper branch at the time instant n, the variance of the noise at any time instant n at the output of $\Upsilon_k^{(a)}(z)$ is:

$$V_{\text{tot}} = V_n + V_{n+m_k^{(a)}}, \tag{A.38}$$

where V_n is given by (A.37).

The output of $\widehat{E}_k^{(r)}(z)$

As seem from Fig. 3.7, the noise at the output of $\Upsilon_k^{(a)}(z)$ is fed to the FIR filter $\Upsilon_k^{(b)}(z)/\widehat{\mu}_k$, whose transfer function is given by (3.35d). The total noise variance at the filter output is consequently

$$\left[\sum_{i=0}^{m_k^{(b)}-1} \left(\frac{\alpha_k^{i m_k^{(a)}}}{\widehat{\mu}_k} \right)^2 \right] V_{\text{tot}} + m_k^{(b)} \sigma_e^2,$$

where V_{tot} is given by (A.38).

A.22 Derivation of (3.63a)**–**(3.66c) **for Fig. 3.6**

The derivation of (3.63a)–(3.64b) for Fig. 3.6 follows the same principle as that stated for Fig. 3.7. The only difference is the complex nature of the noise and the coefficients. In what follows, the upper branch is considered first.

The noise generated by the multipliers before the demultiplexers

The output noise $\Xi^{(1)}_{k,\text{out}}(n)$ of the upper branch of Fig. 3.6 is presented in Table A.5, which is essentially Table A.3 of Fig. 3.7 with $\alpha_k \mapsto \beta_k$ and $m_k^{(a)} \mapsto n_k^{(a)}$.

n	0	1	2	...	$2n_k^{(a)}-1$
$\Xi^{(1)}_{k,\text{out}}(n)$	$E_0 = e_0$	$E_1 = e_1 + (1/\beta_k)\, E_0$ $= e_1 + (1/\beta_k)\, e_0$	$E_2 = e_2 + (1/\beta_k)\, E_1$ $=(1/\beta_k)^2\, e_0 + (1/\beta_k)\, e_1 + e_2$	...	0 (=resetting time)

Table A.5: The output noise of the upper branch of Fig. 3.6, generated by the multipliers before the demultiplexers.

As for the case of Fig. 3.7, in general:

$$E_n = \sum_{i=0}^{n} T_{n-i} e_i, \quad 0 \le n \le 2n_k^{(a)} - 2,$$

where the coefficients T_n are defined according to (A.34), with $\alpha_k \mapsto \beta_k$.

Since the noise E_n is complex, the real and the imaginary parts of E_n are

$$\begin{aligned} \text{Re}\,\{E_n\} =& \text{Re}\left\{\sum_{i=0}^{n} T_{n-i} e_i\right\} = \sum_{i=0}^{n} \text{Re}\,\{T_{n-i} e_i\} \\ =& \sum_{i=0}^{n} \left(\text{Re}\,\{T_{n-i}\}\,\text{Re}\,\{e_i\} - \text{Im}\,\{T_{n-i}\}\,\text{Im}\,\{e_i\}\right) \end{aligned} \tag{A.39a}$$

and

$$\begin{aligned}\text{Im}\left\{E_n\right\} =&\text{Im}\left\{\sum_{i=0}^{n} T_{n-i}e_i\right\} = \sum_{i=0}^{n}\text{Im}\left\{T_{n-i}e_i\right\}\\ =&\sum_{i=0}^{n}\left(\text{Re}\left\{T_{n-i}\right\}\text{Im}\left\{e_i\right\}+\text{Im}\left\{T_{n-i}\right\}\text{Re}\left\{e_i\right\}\right)\end{aligned} \tag{A.39b}$$

respectively.

Considering the independency of Re $\{e_i\}$ and Im $\{e_i\}$, and appreciating the fact that the noise variance of both Re$\{e_i\}$ and Im$\{e_i\}$ is σ_e^2, the variances of the real and the imaginary parts of E_n are followed from (A.39a) and (A.39b) to be

$$\text{var}\left(\text{Re}\left\{E_n\right\}\right) =\text{var}\left(\text{Im}\left\{E_n\right\}\right) = A_{n+1}\sigma_e^2 \tag{A.40a}$$

where

$$\begin{aligned}A_{n+1} =&\sum_{i=0}^{n}\text{Re}^2\left\{T_{n-i}\right\}+\text{Im}^2\left\{T_{n-i}\right\}.\\ =&\sum_{i=0}^{n}\left|\frac{1}{\beta_k^i}\right|^2 \quad \text{(By (A.35))}\end{aligned} \tag{A.40b}$$

The noise generated by the upper recursive filter

The generation of noise originated by the upper recursive filter is illustrated in Table A.6.

n	0	1	2	3	...	$2n_k^{(a)}-1$
$\Xi_{k,\text{out}}^{(1)}(n)$	0 (=zero still in the memory)	$F_1 = f_1$	$F_2 = f_2 + (1/\beta_k)\,F_1 = f_2 + (1/\beta_k)\,f_1$	$F_3 = f_3 + (1/\beta_k)\,F_2 = (1/\beta_k)^2\,f_1 + (1/\beta_k)\,f_2 + f_3$	...	0 (=re-setting time)

Table A.6: Generation of noise originated by the upper recursive filter in Fig. 3.6.

As seen from Table A.6, the noise generated by the upper recursive filter lags that generated by the multipliers before the demultiplexers by one sample (since it takes one

multiplication operation before this noise is generated in the recursive part of the filter), but otherwise, it is subject to the same recursion. Consequently, the variance of the real and complex parts of the noise F_n are simple modification of (A.40a), as given by

$$\text{var}\left(\text{Re}\left\{F_n\right\}\right) = \text{var}\left(\text{Im}\left\{F_n\right\}\right) = 2A_n\sigma_e^2 \tag{A.41}$$

The premultiplier "2" in (A.41) stems from the fact that both the real and the imaginary parts of the complex noise f_i are equivalent to two noise sources, since they are generated by two multiplications.

The total noise

Considering the independencies of the real and imaginary parts of the complex noises f_i and e_i, the total variance of the noise generated at time n in the upper branch follows from (A.40a) and (A.41) to be

$$V_n = \left(2A_n + A_{n+1}\right)\sigma_e^2. \tag{A.42}$$

The variance V_n as given by (A.42) applies to both the real and the imaginary parts of the upper branch.

The output of $\Gamma_k^{(a)}(z)$

Considering the fact that the variance of the noise at each branch is periodic with the period of $2n_k^{(a)}$, and also the fact that the variance of the noise at the time instant $n + n_k^{(a)}$ for the lower branch is equal to that of the upper branch at the time instant n, the variance of the noise at any time instant n at the output of $\Gamma_k^{(a)}(z)$ is:

$$V_{\text{tot}} = V_n + V_{n+n_k^{(a)}}, \tag{A.43}$$

where V_n is given by (A.42). The total variance V_{tot} as given by (A.43) applies to both the real and the imaginary parts of the output of $\Gamma_k^{(a)}(z)$.

The output of $\widehat{E}_k^{(r)}(z)$

As seen from Fig. 3.6, the noise at the output of $\Gamma_k^{(a)}(z)$ is fed to the FIR filter $\Gamma_k^{(b)}(z)/\widehat{\nu}_k$, whose transfer function is given by (3.36d). The total noise variance at the filter output is consequently

$$\left[\sum_{i=0}^{n_k^{(b)}-1}(b_i)^2\right] V_{\text{tot}} + (2n_k^{(b)}-1)\sigma_e^2, \tag{A.44}$$

where V_{tot} is given by (A.43) and

$$b_i = \text{abs}\left(\frac{2\beta_k^{in_k^{(a)}}}{\widehat{\nu}_k}\right).$$

Note that each complex multiplier $\frac{2\beta_k^{in_k^{(a)}}}{\widehat{\nu}_k}$ is realized by 2 real multipliers except for the real multiplier b_0. Hence the total number of real multipliers to implement all $\frac{2\beta_k^{in_k^{(a)}}}{\widehat{\nu}_k}$'s is $2n_k^{(b)}-1$, explaining the second term of (A.44).

A.23 Derivation of (3.69a) **and** (3.69b)

To appreciate these equations, an example is in order. Assuming $n_k^{(a)}$=2 and $n_k^{(b)}$=3, by (3.51b), (3.52a) and (3.52b), the order of $\left[\widehat{\Omega}_k^{(a)}(z)\Re\left\{\Gamma_k^{(a)}(z)\right\}\right]$ is found to be $2n_k^{(a)}-1=3$. This implies that the input signal $x(n)$ is parsed into blocks of length $2n_k^{(a)}=4$.

First, the upper branch of the structure of Fig. 3.12 is considered. The upper demultiplexer has been denoted by Up, the middle demultiplexer by Mi and the lower one by Lo. The following table indicates the samples $x(n)$ fed to the upper branch at each time instant n. The table also signifies which demultiplexer feeds the sample in question.

Obviously the upper branch is in charge of filtering the samples $x(0), x(1), x(2), x(3), x(8), x(9), x(10), x(11), x(16), \ldots$, and the lower branch is in charge of filtering the samples $x(4), x(5), x(6), x(7), x(12), x(13), x(14), x(15), x(20), \ldots$. The time instance $n=7$ is very important, since then, the output will be computed, and at the same time, the

Time n	0	1	2	3	4	5	6	7	8	9	10	11	12	...
Up	$x(0)$	$x(1)$	$x(2)$	$x(3)$	–	–	–	–	$x(8)$	$x(9)$	$x(10)$	$x(11)$	–	–
Mi	–	–	$x(0)$	$x(1)$	$x(2)$	$x(3)$	–	–	–	–	$x(8)$	$x(9)$	$x(10)$	$x(11)$
Lo	–	–	–	–	$x(0)$	$x(1)$	$x(2)$	$x(3)$	–	–	–	–	$x(8)$	$x(9)$

Table A.7: The flow of the input samples in the upper branch of the structure of Fig. 3.12.

memory elements will be set to zero before the beginning of a new cycle, that is, the time instance $n = 8$. Note that ideally, the output at time $n = 7$ should be zero, since the order of the FIR filter $\left[\widehat{\Omega}_k^{(a)}(z)\Re\left\{\Gamma_k^{(a)}(z)\right\}\right]$ is 3. Filtering of 4 samples (such as $x(0), x(1), x(2), x(3)$) by such filter must yield zero at $n = 7$. That explains why the memory (delay) elements are set to zero at such time instant.

Based on the above-mentioned example, the effects of the recursive filter on the input noise $x(0) \equiv \Xi_{k,\text{in}}^{(1)}(0)$ (as given by (3.59a)) for the time instants $n = 0 \cdots 7$ is presented in Table A.8. The result is identical to that presented in TableA.2, since $x(0)$ in both cases is subject to the same recursive filter as given by (A.30), and therefore the argument of page 164 applies. The only difference is that for the structure of Fig. 3.12, the cycle takes $4n_k^{(a)} - 1$ samples, while for that of Fig. 3.6, the cycle takes $2n_k^{(a)} - 1$ samples.

n	0	1	2	3	4	5	6	7 (re-setting time)
$\Xi_{k,\text{ou}}^{(1)}$	$x(0)$	$(1/\beta_k)\,x(0)$	$(1/\beta_k^2)\,x(0)$	$(1/\beta_k^3)\,x(0)$	$(1/\beta_k^4)\,x(0)$	$(1/\beta_k^5)\,x(0)$	$(1/\beta_k^6)\,x(0)$	0

Table A.8: The output of the recursive filter of the upper branch for $n = 0 \cdots n = 7$.

A.24 Derivation of (3.69b)

The cyclic nature of the output noise indicates that a "mod" operator can facilitate the noise calculation. In the sequel, only the upper branch is considered.

By (3.67a), $\Xi^{(1)}_{k,\text{out}}(l) \equiv \Xi^{(1)}_{k,\text{out}}(4\rho n^{(a)}_k + \lambda)$, therefore $\lambda = l \bmod (4n^{(a)}_k) \equiv (l-j) \bmod (n)$, with $j \equiv 0$, $n \equiv 4n^{(a)}_k$.

By (3.67b), $\Xi^{(1)}_{k,\text{out}}(l) \equiv \Xi^{(1)}_{k,\text{out}}((4\rho+2)n^{(a)}_k+\lambda)$, therefore $\lambda = (l-2n^{(a)}_k) \bmod (4n^{(a)}_k) \equiv (l-j) \bmod (n)$ with $j \equiv 2n^{(a)}_k$, $n \equiv 4n^{(a)}_k$.

A.25 Derivation of (3.69a)

In words, (3.69a) states that "add the output noises of the first and the second branches, $\Xi^{(1)}_{k,\text{out}}(l) + \Xi^{(2)}_{k,\text{out}}(l)$, using the notation of (3.69b), and convolve that by $\widehat{\Omega}^{(b)}_k(z)/\widehat{\nu}^{(1)}_k\nu^{(2)}_k$, as shown in Fig. 3.12". Note that the coefficients of $\widehat{\Omega}^{(b)}_k(z)$ are called $\eta_k(l)$, as given by (3.51c).

A.26 Derivation of (3.70c)**–**(3.70g) **for Fig. 3.12**

Taking the upper branch into account, two separate noise sources in the structure are distinguished, i.e., the noise generated by the multipliers before the demultiplexers, and those generated by the upper recursive filter coefficients.

The noise generated by the multipliers before the demultiplexers

Since at different time instants, the upper branch is fed by different number of demultiplexers, the noise generated by the multipliers before the demultiplexers has a time dependent variance. For instance as seen from Table A.7, at time instants $n = 0$ or $n = 1$, there is only one noise input, but at time instants $n = 2 \cdots 5$, there are two noise inputs. This time dependency of the input noise can be characterized by a variable $c(n)$, which has been considered in more details later. To make the analysis less complicated, for the moment we assume that $\forall n : c(n) = 1$, in other words, it has been assumed that there is only one input noise at each time.

Denoting the input noise sample to the upper branch at time instant i by e_i, the output

noise $\Xi^{(1)}_{k,\text{out}}(n)$ of the upper branch of Fig. 3.12 is presented in Table A.9.

n	0	1	2	...	$4n_k^{(a)}-1$
$\Xi^{(1)}_{k,\text{out}}(n)$	$E_0 = e_0$	$E_1 = e_1 + (1/\beta_k)\, E_0$ $= e_1 + (1/\beta_k)\, e_0$	$E_2 = e_2 + (1/\beta_k)\, E_1$ $=(1/\beta_k)^2\, e_0 + (1/\beta_k)\, e_1 + e_2$	...	0 (=resetting time)

Table A.9: The output noise of the upper branch of Fig. 3.12, generated by the multipliers before the demultiplexers.

Apart from the resetting time, Table A.9 is identical to Table A.5, and therefore the analysis presented on page 170 applies here too. The only difference is that now e_i is real, and therefore $\text{Im}\{e_i\} = 0$. By this setting, the real and imaginary parts of E_n are simple modifications of (A.39a) and (A.39b), and are given by

$$\text{Re}\left\{E_n\right\} = \sum_{i=0}^{n} \text{Re}\left\{T_{n-i}\right\} \text{Re}\left\{e_i\right\} \tag{A.45a}$$

and

$$\text{Im}\left\{E_n\right\} = \sum_{i=0}^{n} \text{Im}\left\{T_{n-i}\right\} \text{Re}\left\{e_i\right\} \tag{A.45b}$$

respectively. The coefficients T_n are defined according to (A.34), with $\alpha_k \mapsto \beta_k$.

Considering the independency of $\text{Re}\left\{e_i\right\}$ and $\text{Im}\left\{e_i\right\}$, and appreciating the fact that the noise variance of both $\text{Re}\{e_i\}$ and $\text{Im}\{e_i\}$ is σ_e^2, the variances of the real and the imaginary parts of E_n are followed from (A.45a) and (A.45b) to be

$$\text{var}\left(\text{Re}\left\{E_n\right\}\right) = \text{var}\left(\sum_{i=0}^{n} \text{Re}\left\{T_{n-i}\right\} \text{Re}\left\{e_i\right\}\right) = \sum_{i=0}^{n} \text{Re}^2\left\{\frac{1}{\beta_k^i}\right\} \sigma_e^2 \tag{A.46a}$$

and

$$\text{var}\left(\text{Im}\left\{E_n\right\}\right) = \text{var}\left(\sum_{i=0}^{n} \text{Im}\left\{T_{n-i}\right\} \text{Re}\left\{e_i\right\}\right) = \sum_{i=0}^{n} \text{Im}^2\left\{\frac{1}{\beta_k^i}\right\} \sigma_e^2. \tag{A.46b}$$

The correspondence of T_n to $1/\beta_k$ is established in (A.35) with $\alpha_k \mapsto \beta_k$.

The noise generated by the upper recursive filter

The generation of noise originated by the upper recursive filter is illustrated in Table A.6.

n	0	1	2	3	...	$4n_k^{(a)}-1$
$\Xi_{k,\text{out}}^{(1)}(n)$	0 (=zero still in the memory)	$F_1 = f_1$	$F_2 = f_2 + (1/\beta_k)\, F_1 = f_2 + (1/\beta_k)\, f_1$	$F_3 = f_3 + (1/\beta_k)\, F_2 = (1/\beta_k)^2\, f_1 + (1/\beta_k)\, f_2 + f_3$	...	0 (=re-setting time)

Table A.10: Generation of noise originated by the upper recursive filter in Fig. 3.12.

Apart from the resetting time, Table A.10 is identical to Table A.6, and therefore the analysis presented on page 170 applies here too. The only difference is that at any time instance n, f_n is generated by a real input while $F_i, i < n$ are generated by complex inputs. In other words, f_n is generated by one and $F_i, i < n$ by two multiplications. Since each multiplication gives rise to a quantization error input, this error dependency should be included in (A.41). To this end, the coefficient $c_2(i)$ is introduced to modify (A.41) as

$$\text{var}\left(\text{Re}\left\{F_n\right\}\right) = \text{var}\left(\text{Im}\left\{F_n\right\}\right) = \sum_{i=1}^{n} c_2(i) \left|\frac{1}{\beta_k^{i-1}}\right|^2 \sigma_e^2, \tag{A.47}$$

with

$$c_2(i) = \begin{cases} 1, & i = 1, \\ 2, & \text{otherwise.} \end{cases} \tag{A.48}$$

Note that A_n as given by (A.41) is defined in (A.40b).

The total noise

Considering the independencies of the real and imaginary parts of the complex noises f_i and e_i, the total variance of the noise generated at time n in real and imaginary parts of the upper branch follows from (A.46) and (A.47) to be

$$V_{n,\text{re}} = \sum_{i=0}^{n} c_1(i)\text{Re}^2\left\{\frac{1}{\beta_k^i}\right\}\sigma_e^2 + \sum_{i=1}^{n} c_2(i)\left|\frac{1}{\beta_k^{i-1}}\right|^2\sigma_e^2 \tag{A.49a}$$

and

$$V_{n,\text{im}} = \sum_{i=0}^{n} c_1(i)\text{Im}^2\left\{\frac{1}{\beta_k^i}\right\}\sigma_e^2 + \sum_{i=1}^{n} c_2(i)\left|\frac{1}{\beta_k^{i-1}}\right|^2\sigma_e^2 \tag{A.49b}$$

respectively.

To find out the coefficient $c(i)$, the example of page 173 is reconsidered. As illustrated by Table A.7, the time slot in which the upper branch is fed by the demultiplexers Up, Mi and Lo is $0 \le n \le 2n_k^{(a)}-1$, $n_k^{(a)} \le n \le 3n_k^{(a)}-1$, and $2n_k^{(a)} \le n \le 4n_k^{(a)}-1$ respectively. In other words, at time instants $n_k^{(a)} \le n \le 3n_k^{(a)} - 1$, two simultaneous input noise are introduced to the upper branch by the demultiplexers, and the values to be counted twice are in a "window", starting from $n - 3n_k^{(a)} + 1$ stretching up to $n - n_k^{(a)}$ (if possible). For instance as seen from Table A.7, at $n = 6$, the noises produced from $n - 3n_k^{(a)} + 1 = 1$ to $n - n_k^{(a)} = 4$ samples before $n = 6$ should be considered twice, that is, the noise produced at time instants $2 \cdots 5$. Consequently, $c_1(i)$ in (A.49a) and (A.49b) can be expressed as

$$c_1(i) = \begin{cases} 2, & n - 3n_k^{(a)} + 1 < i \le n - n_k^{(a)}, \\ 1, & \text{otherwise.} \end{cases} \tag{A.50}$$

The output prior to the multiplier $\widehat{\nu}_k^{(2)}\widehat{\gamma}_k$

Considering the fact that the variance of the noise at each branch is periodic with the period of $4n_k^{(a)}$, and also the fact that the variance of the noise at the time instant $n + 2n_k^{(a)}$ for

the lower branch is equal to that of the upper branch at the time instant n, the variances of the real and the imaginary part of the noise at any time instant n at the output prior to the multiplier $\widehat{\nu}_k^{(2)}\widehat{\gamma}_k$ are given by

$$V_{n,\text{tot,re}} = V_{n,\text{re}} + V_{n+2n_k^{(a)},\text{re}} \tag{A.51a}$$

and

$$V_{n,\text{tot,im}} = V_{n,\text{im}} + V_{n+2n_k^{(a)},\text{im}} \tag{A.51b}$$

where $V_{n,\text{re}}$ and $V_{n,\text{im}}$ are given by (A.49a) and (A.49b) respectively.

The output after $\widehat{\nu}_k^{(2)}\widehat{\gamma}_k$

The variance of the output noise after the multiplier $\widehat{\nu}_k^{(2)}\widehat{\gamma}_k$ is given by

$$V_{n,\text{tot}} = \text{Re}^2\left\{\widehat{\nu}_k^{(2)}\widehat{\gamma}_k V_{n,\text{tot,re}}\right\} + \text{Im}^2\left\{\widehat{\nu}_k^{(2)}\widehat{\gamma}_k\right\} V_{n,\text{tot,im}} + 2\sigma_e^2. \tag{A.52}$$

The term σ_e^2 in (A.52) signifies the noise generated by the two real multiplier implementing the complex multipliers $\widehat{\nu}_k^{(2)}\widehat{\gamma}_k$.

The output of $\widehat{E}^{(c)}(z)$

As seem from Fig. 3.12, the noise after the multiplier $\widehat{\nu}_k^{(2)}\widehat{\gamma}_k$ is fed to the FIR filter with the transfer function $2\widehat{\Omega}_k^{(b)}(z)/\widehat{\nu}_k^{(1)}\nu_k^{(2)}$. The transfer function of $\widehat{\Omega}_k^{(b)}(z)$ is given by (3.51c), and consequently, the total noise variance at the filter output is given by

$$\left[\sum_{i=0}^{2n_k^{(b)}-2}(b_i)^2\right] V_{n,\text{tot}} + (2n_k^{(b)} - 1)\sigma_e^2, \tag{A.53}$$

where

$$b_i = \text{abs}\left(\frac{2\eta_k(i)}{\widehat{\nu}_k^{(1)}\widehat{\nu}_k^{(2)}}\right).$$

The term $(2n_k^{(b)} - 1)\sigma_e^2$ in (A.53) signifies the noise generated by the $(2n_k^{(b)} - 1)$ real multipliers implementing the filter with the transfer function $2\widehat{\Omega}_k^{(b)}(z)/\widehat{\nu}_k^{(1)}\nu_k^{(2)}$.

Appendix B

Derivation of Some Formulae in Chapter 4

B.27 Derivation of (4.1)

Consider the numerator of (3.10). Now

$$(1-q_k z^{-1})(1-q_k z) = 1 - q_k(z+z^{-1}) + q_k^2 = -q_k\left[(z+z^{-1}) - (q_k + q_k^{-1})\right].$$

The same applies to the each numerator/denominator term of $G(z)G(z^{-1})$.

B.28 Derivation of (4.3b)

By (4.2c) and the definition of (4.4b), (4.3b) follows from the identity:

$$\frac{m}{(z+z^{-1})-(x+x^{-1})} = \frac{-x.m}{(1-xz^{-1})(1-xz)} \tag{B.1}$$

B.29 Derivation of (4.3a)

Consider (4.2b). Now

$$\begin{aligned}
A_k(z) &= \sum_{k=0}^{L-K} A_k \left(z + z^{-1}\right)^k \\
&= A_0 \left(z + z^{-1}\right)^0 + A_1 \left(z + z^{-1}\right)^1 + A_2 \left(z + z^{-1}\right)^2 + A_3 \left(z + z^{-1}\right)^3 + \cdots \\
&= \binom{0}{0} A_0 z^0 + \\
&= \binom{1}{0} A_1 z^{-1} + \binom{1}{0} A_0 z^1 \\
&= \binom{2}{0} A_2 z^{-1} + \binom{2}{1} A_2 z^0 + \binom{2}{2} A_2 z^1 \\
&= \binom{3}{0} A_3 z^{-3} + \binom{3}{1} A_3 z^{-2} + \binom{3}{2} A_3 z^1 + \binom{3}{3} A_3 z^3 + \cdots
\end{aligned}$$

The pattern above is exactly what has been expressed in (4.3a) with a_k defined in (4.4a). (The pattern follows that of the so-called Pascal triangle [45].)

B.30 Derivation of (4.8) **and** (4.10b)

Based on the identity

$$\begin{aligned}
(1 - \alpha z)\left(1 - \alpha z^{-1}\right) \sum_{n=-M+1}^{M-1} \alpha^{|n|} z^n =& 1 - \alpha^2 + \alpha^{M+1}\left[z^{M-1} + z^{-(M-1)}\right] \\
& - \alpha^M \left[z^M + z^{-M}\right],
\end{aligned}$$

(4.8) simplifies to

$$\frac{\kappa_k}{1 - \alpha_k^2} \sum_{n=-M_k+1}^{M_k-1} \alpha_k^{|n|} z^n,$$

whose impulse response is a truncation of (4.5c), as represented by (4.7). Derivation of (4.10b) follows by the same argument.

B.31 Derivation of (4.15)

By (4.13), $\widehat{R}(z)$ is an FIR filter, whose impulse response should be defined for $|n| \leq M_k - 1$. Using (4.13) and ignoring the scaling factor κ_k, the transfer function of $\widehat{R}(z)$ can be written as:

$$\frac{1+\alpha_k^{2M_k}}{(1-\alpha_k z^{-1})(1-\alpha_k z)} - \alpha_k^{M_k} z^{M_k} \frac{1}{(1-\alpha_k z^{-1})(1-\alpha_k z)} - \alpha_k^{M_k} z^{-M_k} \frac{1}{(1-\alpha_k z^{-1})(1-\alpha_k z)}. \tag{B.2}$$

By (4.3b) and (4.5c), the impulse response of $\frac{1}{(1-\alpha_k(z))(1-\alpha_k(z^{-1}))}$ is given by $\frac{\alpha_k^{|n|}}{1-\alpha_k^2}$. Hence (using the time shift property of z-transform [72]), the impulse response of (B.2) can be expressed as

$$\frac{\left(1+\alpha_k^{2M_k}\right)\alpha_k^{|n|}}{1-\alpha_k^2} - \alpha_k^{M_k}\frac{\alpha_k^{|n+M_k|}}{1-\alpha_k^2} - \alpha_k^{M_k}\frac{\alpha_k^{|n-M_k|}}{1-\alpha_k^2}, \quad |n| < M_k - 1. \tag{B.3}$$

Since $|n| < M_k - 1$, $\alpha_k^{|n+M_k|} = \alpha_k^{M_k}\alpha_k^n$ and $\alpha_k^{|n-M_k|} = \alpha_k^{M_k}\alpha_k^{-n}$, and consequently the numerator of (B.3) becomes:

$$\begin{aligned}\left(1-\alpha_k^{2M_k}\right)\alpha_k^{|n|} - \alpha_k^{2M_k}\left(\alpha_k^n + \alpha_k^{-n}\right) =& \alpha_k^{|n|} + \underbrace{\alpha_k^{2M_k}\alpha_k^{|n|}}_{T_1} - \underbrace{\alpha_k^{2M_k}\alpha_k^n}_{T_2} - \underbrace{\alpha_k^{2M_k}\alpha_k^{-n}}_{T_3} \\ =& \alpha_k^{|n|} - \alpha_k^{2M_k}\alpha_k^{-|n|},\end{aligned}$$

since for $n \geq 0, T_1 = T_2$, and for $n < 0, T_1 = T_3$.

B.32 Derivation of (4.16)

Obviously

$$\left(\sum_{l=0}^{N-1} x^l z^{-l}\right)\left(\sum_{l=0}^{N-1} x^l z^l\right) = t(0) + \sum_{l=1}^{N-1} t(l)\left(z^l + z^{-l}\right), \tag{B.4a}$$

where

$$t(l) = x^l\left(1 + x^2 + \cdots + \left(x^2\right)^{(N-1-l)}\right) = \frac{x^l\left(1-\left(x^2\right)^{(N-l)}\right)}{1-x^2}. \tag{B.4b}$$

Expressing (4.14b) as

$$\widehat{\Gamma}_k(z) = \gamma_k \left(\sum_{l=0}^{N_k-1} \beta_k^l z^{-l} \right) \left(\sum_{l=0}^{N_k-1} \beta_k^l z^{l} \right) \tag{B.5}$$

and applying (B.4) to (B.5) yields

$$\widehat{\Gamma}_k(z) = t_k(0) + \sum_{l=1}^{N_k-1} t_k(l) \left(z^l + z^{-l} \right), \tag{B.6a}$$

where

$$t_k(l) = \frac{\gamma_k \beta_k^l \left(1 - (\beta_k^2)^{N_k-l} \right)}{1 - \beta_k^2}. \tag{B.6b}$$

Now

$$\begin{aligned} \widehat{C}_k(z) =& 2\Re \left\{ \widehat{\Gamma}_k(z) \right\} \text{ (By (4.14a))} \\ =& 2\text{Re}\{t_k(0)\} + \sum_{l=1}^{N_k-1} 2\text{Re}\{t_k(l)\} \left(z^l + z^{-l} \right) \text{ (By (B.6a))} \end{aligned} \tag{B.7}$$

By (B.6b), the transfer function of (B.7) gives the impulse response of (4.16).

B.33 Derivation of (4.17a)**–**(4.19)

Starting from (4.3c):

$$\begin{aligned} C_k(z) =& 2\Re \left\{ \frac{\gamma_k}{(1 - \beta_k z)(1 - \beta_k z^{-1})} \right\} \\ =& \frac{\gamma_k}{(1 - \beta_k z)(1 - \beta_k z^{-1})} + \frac{\gamma_k^*}{(1 - \beta_k^* z)(1 - \beta_k^* z^{-1})} \\ =& \frac{\overbrace{\gamma_k \left(1 + \beta_k^{*2} \right) + \gamma_k^* \left(1 + \beta_k^2 \right)}^{\eta_0} + \overbrace{\left(-2\gamma_k \beta_k^* - 2\gamma_k^* \beta_k \right)}^{\eta_1} (z + z^{-1})}{(1 - \beta_k z)(1 - \beta_k z^{-1})(1 - \beta_k^* z)(1 - \beta_k^* z^{-1})} \end{aligned}$$

$\widehat{\Omega}_k(z)$ implements the truncation by providing zeros to cancel the poles of $C_k(z)$, in other words $\widehat{C}_k(z) = \widehat{\Omega}_k(z) C_k(z)$, which by (4.3c) yields (4.19).

B.34 Derivation of (4.20)

By (4.3d), (4.14b) and (4.18b):

$$\begin{aligned}\widetilde{\Gamma}_k(z) \equiv&\widehat{\Omega}_k(z)\Gamma_k(z)\\ =&\gamma_k\frac{\left(1-\beta_k^{N_k}z^{-N_k}\right)\left(1-\beta_k^{N_k}z^{N_k}\right)}{\left(1-\beta_k z^{-1}\right)\left(1-\beta_k z\right)}\left(1-\beta_k^{*N_k}z^{-N_k}\right)\left(1-\beta_k^{*N_k}z^{N_k}\right)\\ =&\widehat{\Gamma}_k(z)\widetilde{\Omega}_k(z),\end{aligned} \tag{B.8a}$$

where

$$\begin{aligned}\widetilde{\Omega}_k(z) =&\left(1-\beta_k^{*N_k}z^{-N_k}\right)\left(1-\beta_k^{*N_k}z^{N_k}\right)\\ =&1+\beta_k^{*2N_k}-\beta_k^{*N_k}\left(z^{N_k}+z^{-N_k}\right).\end{aligned} \tag{B.8b}$$

Equation (B.8) implies that the transfer function of $\widetilde{\Gamma}_k(z)$ is a scaled and shifted version of that of $\widehat{\Gamma}_k(z)$, where the transfer function $\widetilde{\Omega}_k(z)$ determines the shift and the scaling. Using (B.6), the resulting transfer function is given by

$$\begin{aligned}\widetilde{\Gamma}_k(z) =&\left(1+\beta_k^{*2N_k}\right)t_k(0)+\sum_{l=1}^{N_k-1}\left(1+\beta_k^{*2N_k}\right)t_k(l)\left(z^l+z^{-l}\right)\\ &-\sum_{l=1}^{N_k-1}\beta_k^{*N_k}t_k(l)\left[z^{l-N_k}+z^{-(l-N_k)}+z^{l+N_k}+z^{-(l+N_k)}\right].\end{aligned} \tag{B.9}$$

Hence for $0\le l\le N_k-1$, the impulse response of $\widetilde{\Gamma}_k(z)$, denoted by $g(l)$ is given by

$$\begin{aligned}g(l) =&\left(1+\beta_k^{*2N_k}\right)t_k(l)-\beta_k^{*N_k}t_k(N_k-l)\\ =&\frac{\gamma_k}{1-\beta_k^2}\left[\left(1+\beta_k^{*2N_k}\right)\left(\beta_k^l-\beta_k^{2N_k-l}\right)+\beta_k^{*N_k}\left(\beta_k^{N_k+l}-\beta_k^{N_k-l}\right)\right].\end{aligned} \tag{B.10}$$

For $N_k\le l\le 2N_k-1$, $g(l)$ is given by

$$-\beta_k^{*N_k}t_k(l-N_k)=\frac{\gamma_k\left(\beta_k\beta_k^*\right)^{N_k}\left(\beta_k^{2N_k-l}-\beta_k^{l-2N_k}\right)}{1-\beta_k^2}. \tag{B.11}$$

B.35 Derivation of (4.25)**–**(4.27)

Obviously

$$1-\left(\alpha_k^{M_k}+\frac{1}{\alpha_k^{M_k}}\right)z^{-M_k}+z^{-2M_k}=\left(1-\left(\alpha_k z^{-1}\right)^{M_k}\right)\left(1-\left(\alpha_k^{-1}z^{-1}\right)^{M_k}\right). \quad \text{(B.12)}$$

Ignoring the term $\kappa_k(\alpha_k)^{(M_k-1)}$, (B.12) presents a factorization for the numerator of (4.22c). If $M_k = m_k^{(a)}m_k^{(a)}$, then (4.23) can be applied to the right hand side (RHS) of (B.12), yielding

$$\begin{aligned}
&\left(1-\left(\alpha_k z^{-1}\right)^{M_k}\right)\left(1-\left(\alpha_k^{-1}z^{-1}\right)^{M_k}\right)\\
&=\left(1-\alpha_k^{m_k^{(a)}}z^{-m_k^{(a)}}\right)\sum_{i=0}^{m_k^{(b)}-1}\left(\alpha_k^{m_k^{(a)}}z^{-m_k^{(a)}}\right)^i\left(1-\alpha_k^{-m_k^{(a)}}z^{-m_k^{(a)}}\right)\sum_{i=0}^{m_k^{(b)}-1}\left(\alpha_k^{-m_k^{(a)}}z^{-m_k^{(a)}}\right)^i\\
&=\underbrace{\left(1-\left(\alpha_k^{m_k^{(a)}}+\frac{1}{\alpha_k^{m_k^{(a)}}}\right)z^{-m_k^{(a)}}+z^{-2m_k^{(a)}}\right)}_{T}\times\\
&\underbrace{\sum_{i=0}^{m_k^{(b)}-1}\left(\alpha_k^{m_k^{(a)}}z^{-m_k^{(a)}}\right)^i\cdot\sum_{i=0}^{m_k^{(b)}-1}\left(\alpha_k^{-m_k^{(a)}}z^{-m_k^{(a)}}\right)^i}_{S}
\end{aligned}$$

Ignoring the term $\kappa_k(\alpha_k)^{(M_k-1)}$, T defines the numerator of (4.25b).

The transfer function of S can be alternatively expressed as

$$z^{-m_k^{(a)}(m_k^{(b)}-1)}\left(s_k(0)+\sum_{l=1}^{m_k^{(b)}-1}s_k(l)\left(z^l+z^{-l}\right)\right), \quad \text{(B.13)}$$

where

$$\begin{aligned}
s_k(l)=&\alpha_k^{-m_k^{(a)}(m_k^{(b)}-1-l)}\sum_{i=0}^{m_k^{(b)}-l-1}\left(\alpha_k^{2m_k^{(a)}}\right)^i\\
=&\alpha_k^{-m_k^{(a)}(m_k^{(b)}-1-l)}\frac{1-\alpha_k^{2m_k^{(a)}(m_k^{(b)}-l)}}{1-\alpha_k^{2m_k^{(a)}}}\\
=&\frac{\left(\alpha^{m_k^{(a)}}\right)^{(m_k^{(b)}-l)}-\left(\alpha^{-m_k^{(a)}}\right)^{(m_k^{(b)}-l)}}{\alpha^{m_k^{(a)}}-\alpha^{-m_k^{(a)}}}.
\end{aligned} \quad \text{(B.14)}$$

Equation (B.14) defines the $\zeta_k(2m_k^{(b)} - 2 - l)$ part of (4.25d). The first part of (4.25d) follows directly from the symmetry of the transfer function defined in (B.13). Through changing the variables $\alpha_k, M_k, m_k^{(a)}$ and $m_k^{(b)}$ with $\beta_k, N_k, n_k^{(a)}$ and $n_k^{(b)}$ respectively, the argument presented above proves (4.27).

B.36 Derivation of (4.29) **and** (4.31)

To prove (4.31), consider (4.24) with $m_k^{(a)} = 1$ and $m_k^{(b)} = m_k^{(a)}$. Now $\Upsilon_k^{(a)}$ as given by (4.25b) is $\kappa_k \alpha_k^{M_k - 1}$ and $\Upsilon_k^{(b)} = \sum_{l=0}^{2m_k^{(b)}-2} \zeta_k(l) z^{-l}$, where ζ_k is given by (4.25d) with $m_k^{(a)} \to 1$. By the same argument, (4.29a) and (4.29b) can be derived from (4.27d) with $n_k^{(a)} = 1$ and $n_k^{(b)} = n_k^{(a)}$.

B.37 Derivation of (4.32)**–**(4.34)

$$\begin{aligned}
\Upsilon_k^{(b)}(z) &= \sum_{l=0}^{m_k^{(b)}-1} \left(\alpha_k z^{-1}\right)^{l m_k^{(a)}} \sum_{l=0}^{m_k^{(b)}-1} \left(\alpha_k^{-1} z^{-1}\right)^{l m_k^{(a)}} \text{ (Derivation of (4.25)–(4.27))} \\
&= \prod_{i=1}^{I} \sum_{l=0}^{m_k^{(i)}-1} \left(\alpha_k z^{-1}\right)^{l M_k{}^{(i)}} \prod_{k=1}^{I} \sum_{l=0}^{m_k^{(i)}-1} \left(\alpha_k^{-1} z^{-1}\right)^{l M_k{}^{(k)}} \text{ (By (4.32))} \\
&= \prod_{k=1}^{I} \underbrace{\sum_{l=0}^{m_k^{(i)}-1} \left(\alpha_k z^{-1}\right)^{l M_k{}^{(i)}} \sum_{l=0}^{m_k^{(i)}-1} \left(\alpha_k^{-1} z^{-1}\right)^{l M_k{}^{(i)}}}_{\Upsilon_k^{(i)}(z)}.
\end{aligned}$$

By what has been shown in the Derivation of (4.25a)–(4.25d) on page 186, the impulse response of $\Upsilon_k^{(i)}(z)$ is given by (4.33d). (4.34d) is derived by a similar argument.

B.38 Derivation of (4.37)

Ignoring the pre-multiplier $\kappa_k \alpha_k^{M_k - 1}$ in (4.25b), $\Upsilon_k^{(a)}(z)$ can be expressed as:

$$\Upsilon_k^{(a)}(z) = \frac{1 - \left(\alpha_k^{m_k^{(a)}} + \alpha_k^{-m_k^{(a)}}\right) z^{-m_k^{(a)}} + z^{-2m_k^{(a)}}}{1 - \left(\alpha_k + \alpha_k^{-1}\right) z^{-1} + z^{-2}}$$

$$
\begin{aligned}
&= \frac{\left(1-\alpha_k^{m_k^{(a)}} z^{-m_k^{(a)}}\right)\left(1-\alpha_k^{-m_k^{(a)}} z^{-m_k^{(a)}}\right)}{\left(1-\alpha_k z^{-1}\right)\left(1-\alpha_k^{-1} z^{-1}\right)} \\
&= \sum_{l=0}^{m_k^{(a)}-1} \left(\alpha_k z^{-1}\right)^l \sum_{l=0}^{m_k^{(a)}-1} \left(\alpha_k^{-1} z^{-1}\right)^l,
\end{aligned}
$$

but according to Derivation of (4.25a)–(4.25d), the impulse response of the last transfer function, denoted by $\zeta_k^{(a)}(l)$, is given by:

$$
\zeta_k^{(a)}(l) = \begin{cases} \dfrac{\alpha_k^{1(l+1)} - \alpha_k^{-1(l+1)}}{\alpha_k - \alpha_k^{-1}}, & 0 \le l \le m_k^{(a)} - 1, \\ \zeta(2m_k^{(a)} - 2 - l), & m_k^{(a)} \le l \le 2m_k^{(a)} - 2, \end{cases}
$$

proving the validity of (4.37c). Derivation of (4.37d) follows by a similar argument.

B.39 Derivation of (4.40)

It should be proven that $H_k^{(c)}(z)$ as given by (4.40b) is a delayed version of $\widehat{C}_k(z)$, as given by (4.19). To this end, (4.18b) is expanded:

$$
\begin{aligned}
\widehat{\Omega}_k(z) =& \left(1-\beta_k^{N_k} z^{-N_k}\right)\left(1-\beta_k^{*N_k} z^{-N_k}\right)\left(1-\beta_k^{N_k} z^{N_k}\right)\left(1-\beta_k^{*N_k} z^{N_k}\right) \\
=& \beta_k^{N_k}\beta_k^{*N_k} z^{2N_k}\Bigg[1 - \underbrace{\frac{\left(\beta_k^{N_k}\beta_k^{*N_k}+1\right)\left(\beta_k^{N_k}+\beta_k^{*N_k}\right)}{\beta_k^{N_k}\beta_k^{*N_k}}}_{=\chi(0)} z^{-N_k} \\
&+ \underbrace{\frac{1+\left(\beta_k^{N_k}+\beta_k^{*N_k}\right)^2+\beta_k^{2N_k}\beta_k^{*2N_k}}{\beta_k^{N_k}\beta_k^{*N_k}}}_{=\chi(1)} z^{-2N_k} \\
&+ \frac{\left(\beta_k^{N_k}\beta_k^{*N_k}+1\right)\left(\beta_k^{N_k}+\beta_k^{*N_k}\right)}{\beta_k^{N_k}\beta_k^{*N_k}} z^{-3N_k} + z^{-4N_k}\Bigg] \\
=& \beta_k^{N_k}\beta_k^{*N_k} z^{2N_k}\Omega_k(z),
\end{aligned} \tag{B.15}
$$

where $\Omega_k(z)$ is given in (4.40d).

The denominator term of (4.19) can also be expanded as

$$1-\beta_k z-\beta_k z^{-1}+\beta_k^2=-\beta_k z\left(1-\left(\beta_k+\beta_k^{-1}\right)z^{-1}+z^{-2}\right). \tag{B.16}$$

By (B.15) and (B.16), (4.19) can alternatively be expressed as:

$$\begin{aligned}\widehat{C}_k(z)=&\frac{z^{2N_k}}{z}2\Omega_k(z)\Re\left\{\frac{-\beta_k^{N_k}\beta_k^{*N_k}/\beta_k}{1-\left(\beta_k+\beta_k^{-1}\right)z^{-1}+z^{-2}}\right\}\\ =&\frac{z^{2N_k}}{z}2\Omega_k(z)\Re\left\{\frac{\widehat{\gamma}_k}{1-\left(\beta_k+\beta_k^{-1}\right)z^{-1}+z^{-2}}\right\}\text{(By (4.40c))},\end{aligned} \tag{B.17}$$

proving that $H_k^{(c)}(z)$ as given by (4.40b) is a delayed version of $\widehat{C}_k(z)$.

B.40 Derivation of (4.42)

By (B.15), $\Omega_k(z)$ can alternatively be expressed as

$$\Omega_k(z)=\left(1-\beta_k^{N_k}z^{-N_k}\right)\left(1-\beta_k^{*N_k}z^{-N_k}\right)\left(1-\beta_k^{-N_k}z^{-N_k}\right)\left(1-\beta_k^{*-N_k}z^{-N_k}\right), \tag{B.18}$$

and by the same argument

$$\Omega_k^{(a)}(z)=\left(1-\beta_k^{n_k^{(a)}}z^{-n_k^{(a)}}\right)\left(1-\beta_k^{*n_k^{(a)}}z^{-n_k^{(a)}}\right)\left(1-\beta_k^{-n_k^{(a)}}z^{-n_k^{(a)}}\right)\left(1-\beta_k^{*-n_k^{(a)}}z^{-n_k^{(a)}}\right). \tag{B.19}$$

Now by (B.18) and (B.19)

$$\frac{\Omega_k(z)}{\Omega_k^{(a)}(z)}=\frac{\left(1-\beta_k^{N_k}z^{-N_k}\right)}{\left(1-\beta_k^{n_k^{(a)}}z^{-n_k^{(a)}}\right)}\frac{\left(1-\beta_k^{*N_k}z^{-N_k}\right)}{\left(1-\beta_k^{*n_k^{(a)}}z^{-n_k^{(a)}}\right)}\frac{\left(1-\beta_k^{-N_k}z^{-N_k}\right)}{\left(1-\beta_k^{-n_k^{(a)}}z^{-n_k^{(a)}}\right)}\frac{\left(1-\beta_k^{*-N_k}z^{-N_k}\right)}{\left(1-\beta_k^{*-n_k^{(a)}}z^{-n_k^{(a)}}\right)}. \tag{B.20}$$

If $N=n_k^{(a)}n_k^{(b)}$, then (B.20) can be expressed according to (4.23) as

$$\sum_{l=0}^{n_k^{(b)}-1}\left(\beta_k^{n_k^{(a)}}z^{-n_k^{(a)}}\right)^l\sum_{l=0}^{n_k^{(b)}-1}\left(\beta_k^{-n_k^{(a)}}z^{-n_k^{(a)}}\right)^l\sum_{l=0}^{n_k^{(b)}-1}\left(\beta_k^{*n_k^{(a)}}z^{-n_k^{(a)}}\right)^l\sum_{l=0}^{n_k^{(b)}-1}\left(\beta_k^{*-n_k^{(a)}}z^{-n_k^{(a)}}\right)^l. \tag{B.21}$$

By what has been shown in the Derivation of (4.32)–(4.34), (B.21) equals

$$\Gamma_k^{(b)}(z)\Gamma^{*(b)}_k(z), \tag{B.22}$$

where $\Gamma_k^{(b)}(z)$ is defined by (4.34c) and (4.34d), and $\Gamma^{*(b)}_k(z)$ is derived from $\Gamma_k^{(b)}(z)$ by $\beta_k \to \beta_k^*$. This proves (4.42g).

In principle, (4.42f) states that the polynomial multiplication defined by (B.22) can be performed by the convoluting the coefficients of $\Gamma_k^{(b)}(z)$ and $\Gamma^{*(b)}_k(z)$. Moreover, since both $\Gamma_k^{(b)}(z)$ and $\Gamma^{*(b)}_k(z)$ are symmetric, $\Omega_k^{(j)}(z)$ as given by (4.42f) is symmetric too.

To appreciate (4.42f), an example is worked out. Define

$$g_k^{(b)}[n] = [a_0 \quad a_1 \quad a_2 \quad a_3 \quad a_2 \quad a_1 \quad a_0]$$

and

$$g^{*(b)}_k[n] = [a_0^* \quad a_1^* \quad a_2^* \quad a_3^* \quad a_2^* \quad a_1^* \quad a_0^*]$$

to represent the impulse responses of $\Gamma_k^{(b)}(z)$ and $\Gamma^{*(b)}_k(z)$ respectively. Denoting the convolution operation by $\otimes$, the first, the second and the third samples of $g_k^{(b)}[n] \otimes g^{*(b)}_k[n]$ are given by $a_0a_0^*$, $\underbrace{a_0a_1^* + a_1a_0^*}_{2\mathrm{Re}\{a_0a_1^*\}}$, and $\underbrace{a_0a_2^* + a_2a_0^*}_{2\mathrm{Re}\{a_0a_2^*\}} + a_1a_1^*$ respectively, which are exactly those given in (4.42f).

Finally by (B.18)–(B.20), decomposibility of N_k as given by (4.41b) and (4.41a) leads to a corresponding decomposibility of $\Omega_k^{(b)}(z)$, as given by (4.41d). For instance, assume

$$\begin{aligned} N_k =& n_k^{(a)} n_k^{(1)} n_k^{(2)} \\ =& N_k^{(2)} n_k^{(2)} \quad \text{By (4.42e)} \end{aligned} \tag{B.23}$$

Since $N_k^{(2)}$ divides N_k, then by (B.20) and (B.21), $\Omega_k^{(a,1)}(z)$ divides $\Omega_k(z)$, where

$$\Omega_k^{(a,1)}(z) = \left(1 - \beta_k^{N_k^{(2)}} z^{-N_k^{(2)}}\right)\left(1 - \beta_k^{*N_k^{(2)}} z^{-N_k^{(2)}}\right) \times \left(1 - \beta_k^{-N_k^{(2)}} z^{-N_k^{(2)}}\right)\left(1 - \beta_k^{*-N_k^{(2)}} z^{-N_k^{(2)}}\right).$$

As proven above, the transfer function of the quotient, denoted by $\Omega_k^{(2)}(z)$, is in the form of (4.42b). But by the same argument, $\Omega_k^{(a,1)}(z)$ is divisible by $\Omega_k^{(a)}(z)$ as given by (B.19), and the quotient, denoted by $\Omega_k^{(1)}(z)$, is in the form of (4.42b). In other words, it has been shown that

$$\Omega_k(z) = \Omega_k^{(a)}(z)\Omega_k^{(1)}(z)\Omega_k^{(2)}(z)$$

as required. So basically the argument presented above boils down to an inductive proof, where the inductive base is $\Omega_k^{(a)}(z)$ and all other $\Omega_k^{(i)}(z)$'s are of the form of (4.42b).

B.41 Derivation of (4.43)

$$\begin{aligned}\Omega_k^{(1)}(z) &= \sum_{l=0}^{4n_k^{(1)}-4} \eta_k^{(1)}(l) z^{-lN_k^{(1)}} \text{ (By (4.42b))} \\ &= \sum_{l=0}^{4n_k^{(1)}-4} \eta_k^{(1)}(l) z^{-ln_k^{(a)}} \text{ (By (4.42e))}. \end{aligned} \tag{B.24}$$

But for J_k=1, $n_k^{(b)} = n_k^{(1)}$ (by(4.41b)) and $\Omega_k^{(b)}(z) = \Omega_k^{(1)}(z)$ (by(4.41d)), and consequently (B.24) can be expressed by

$$\Omega_k^{(b)}(z) = \sum_{l=0}^{4n_k^{(b)}-4} \eta_k^{(b)}(l) z^{-ln_k^{(a)}}$$

as required.

B.42 Derivation of (4.45) **and** (4.46)

Before presenting the discussion, the following proposition is presented and proved.

Proposition 2. *Assuming the realness of the impulse response of $A_k(z)$, if the impulse response of $A_k(z)B(z)$ is $h(n)$, then the impulse response of $A_k(z)\Re\{B(z)\}$ is Re $\{h(n)\}$.*

Proof. Assume that $A_k(z) = \sum_{i=0}^{\infty} a_i z^{-i}$ and $B(z) = \sum_{i=0}^{\infty} b_i z^{-i}$. Then $B^*(z) = \sum_{i=0}^{\infty} b_i^* z^{-i}$.

$B^*(z)$ is the transfer function corresponding to $B(z)$, with changing all the coefficients of $B(z)$ into their complex conjugate counterparts, and $a_i, b_i,$ and b_i^* are the impulse response values of $A_k(z), B(z)$ and $B^*(z)$ respectively. Now if the impulse response values of the filter $A_k(z)\Re\{B(z)\}$are called c_i, then

$$\sum_{i=0}^{\infty} c_i z^{-i} = A_k(z)\Re\{B(z)\} = \frac{1}{2}A_k(z)\left(B(z) + B^*(z)\right)$$

$$= \frac{1}{2}\sum_{i=0}^{\infty} a_i z^{-i}.\sum_{i=0}^{\infty} 2\text{Re}\{b_i\} z^{-i} = \text{Re}\left\{\sum_{i=0}^{\infty} a_i z^{-i}\sum_{i=0}^{\infty} b_i z^{-i}\right\}$$

The last term in brackets gives the impulse response values of $A_k(z)B(z)$. □

On page 128, it reads that the impulse response of $\nu_k^{(1)}\nu_k^{(2)}\Gamma_k^{(a)}(z)\prod_{i=1}^{j}\Omega_k^{(j)}(z)$ is given by Re $\left\{b(l, N_k^{(j+1)}, \nu_k^{(1)}\nu_k^{(2)}\widehat{\gamma}_k\prod_{i=1}^{j}\varphi_k^{(i)})\right\}$. The validity of the claim is investigated below. In what follows, the coefficients $\nu_k^{(1)}\nu_k^{(2)}$ have been ignored.

As seen from Fig. 4.8 as using (4.44b), the transfer function from the input to the output of $\Omega_k^{(j)}(z)$ is:

$$H(z) = \prod_{i=1}^{j}\varphi_k^{(i)}\left[G(z)\Re\left\{\frac{\widehat{\gamma}_k}{1-\left(\beta_k+\beta_k^{-1}\right)z^{-1}+z^{-2}}\right\}\right], \tag{B.25}$$

where

$$G(z) = \Omega_k^{(a)}(z)\prod_{i=1}^{j}\Omega_k^{(i)}(z).$$

By the proof presented in the Derivation of (4.42), the transfer function $G(z)$ is of the

form:

$$\begin{aligned} G(z) =& \left(1-\beta_k^{N_k^{(j+1)}} z^{-N_k^{(j+1)}}\right)\left(1-\beta_k^{*N_k^{(j+1)}} z^{-N_k^{(j+1)}}\right)\left(1-\beta_k^{-N_k^{(j+1)}} z^{-N_k^{(j+1)}}\right) \\ & \left(1-\beta_k^{*-N_k^{(j+1)}} z^{-N_k^{(j+1)}}\right) \\ =& 1-\varepsilon(0) z^{-N_k^{(j+1)}}+\varepsilon(1) z^{-2N_k^{(j+1)}}-\varepsilon(0) z^{-3N_k^{(j+1)}}+z^{-4N_k^{(j+1)}}, \end{aligned} \tag{B.26}$$

where $N_k^{(j)}$ is given by (4.42e) and

$$\begin{aligned} \varepsilon(0) =& \beta_k^{N_k^{(j+1)}}+\beta_k^{*N_k^{(j+1)}}+\beta_k^{-N_k^{(j+1)}}+\beta_k^{*-N_k^{(j+1)}}, \\ \varepsilon(1) =& 2+\left(\beta_k^{N_k^{(j+1)}}+\beta_k^{-N_k^{(j+1)}}\right)\left(\beta_k^{*N_k^{(j+1)}}+\beta_k^{*-N_k^{(j+1)}}\right). \end{aligned}$$

As an example for $J=1$:

$$n_k^{(b)}=n_k^{(1)}, \text{ so } N_k=n_k^{(a)}n_k^{(b)}=n_k^{(b)}n_k^{(1)}, \text{ and } N_k^{(1)}=n_k^{(a)}, N_k^{(2)}=n_k^{(a)}n_k^{(1)}=n_k^{(a)}n_k^{(b)}.$$

In other words, the index J indicates the number of filters coming after $\Omega_k^{(a)}(z)$. If there is only one filter to come, it is $\Omega_k^{(b)}(z)$. If there are more, they are $\Omega_k^{(1)}(z)$, $\Omega_k^{(2)}(z)$, etc. The index "j" in $\Omega_k^{(j)}(z)$ corresponds to the aforementioned $\Omega_k^{(1)}(z)$, $\Omega_k^{(2)}(z), \cdots$,. The index j in $N_k^{(j)}$ is a number indicating the total number of filters in the compound cascade structure, so $N_k^{(1)}$ corresponds to $\Omega_k^{(a)}(z)$, $N_k^{(2)}$ corresponds to the cascade of $\Omega_k^{(a)}(z)$ with $\Omega_k^{(1)}(z)$, etc. For the simplest (two filter) structure, $J=1$, $N_k^{(1)}=n_k^{(a)}$, and so the whole structure is: $\underbrace{\Omega_k^{(a)}(z)}_{n_k^{(a)}\times}\underbrace{\Omega_k^{(1)}(z)}_{n_k^{(1)}=N_k^{(2)}}$ and for J=2 we have: $\underbrace{\Omega_k^{(a)}(z)}_{n_k^{(a)}\times}\underbrace{\Omega_k^{(1)}(z)}_{n_k^{(1)}\times}\underbrace{\Omega_k^{(2)}(z)}_{n_k^{(2)}=N_k^{(3)}}$. The examples above should clarify the significance of $N_k^{(j+1)}$ appearing in the expression for $G(z)$ as given by (B.26).

Finally we prove that $b(l, N_k^{(j+1)}, \nu_k^{(1)}\nu_k^{(2)}\widehat{\gamma}_k\prod_{l=1}^{j}\varphi_k^{(l)})$ is the impulse response of $H(z)$ as given by (B.25). To this end, the impulse response of

$$K(z)=G(z)\frac{1}{1-\left(\beta_k+\beta_k^{-1}\right)z^{-1}+z^{-2}}$$

is considered, where $G(z)$ is defined in (B.26).

1. The roots of the denominator of $K(z)$, β_k and β_k^{-1}, are the roots of $G(z)$, and consequently, $K(z)$ is an FIR filter of order $4N_k^{(j+1)} - 2$. This proves the last part of (4.45a).

2. The second last part of (4.45a) states that the impulse response of $K(z)$ is symmetric. This stems from the fact that if the symmetric polynomial $q(x)$ divides the symmetric polynomial $p(x)$, then $p(x)/q(x)$ is a symmetric polynomial. Evidently both the numerator and the denominator of $K(z)$ are symmetric polynomials in z, implying that the impulse response samples of $K(z)$ for $0 \le n \le 2N_k^{(j+1)} - 2$ are also those of $2N_k^{(j+1)} \le n \le 4N_k^{(j+1)} - 2$.

3. The impulse response of $K(z)$ is a shifted and scaled version of that of $F(z)$ (by $G(z)$), with
$$F(z) = \frac{1}{(1-\beta_k z^{-1})} \frac{1}{\left(1-\beta_k^{-1} z^{-1}\right)}.$$
In particular, the first $N_k^{(j+1)} - 1$ impulse response samples of $K(z)$ are equal to those of $F(z)$. By what has been shown in the Derivation of (4.25)–(4.27) on page 186, the impulse response of $F(z)$ for $0 \le n \le N_k^{(j+1)} - 1$ can be found by:
$$\sum \beta_k^l z^{-l} \sum \beta_k^{-l} z^{-l} = \frac{\beta_k^{l+1} - \beta_k^{-l-1}}{\beta_k - \beta_k^{-1}} z^{-l}, \quad 0 \le l \le N_k^{(j+1)} - 1.$$
This confirms the first part of (4.45a).

4. For $N_k^{(j+1)} \le n \le 2N_k^{j+1} - 1$, a scaled-shifted version of the impulse response of $F(z)$ (scaled by $\varepsilon(0)$ as given by (4.40f) and shifted by $N_k^{(j+1)}$), should be added to the impulse response of $F(z)$. This means that the z-transform of the impulse

response is:

$$\frac{\beta_k^{l+1}-\beta_k^{-l-1}}{\beta_k-\beta_k^{-1}}z^{-l}-$$
$$\left(\beta_k^{N_k^{(j+1)}}+\beta_k^{*N_k^{(j+1)}}+\beta_k^{-N_k^{(j+1)}}+\beta_k^{*-N_k^{(j+1)}}\right)\frac{\beta_k^{l+1-N_k^{(j+1)}}-\beta_k^{-l-1+N_k^{(j+1)}}}{\beta_k-\beta_k^{-1}}z^{-l}$$
$$=\frac{1}{\beta_k-\beta_k^{-1}}\times$$
$$\Bigg[-\beta_k^{l-N_k^{(j+1)}+1}\left(\beta_k^{*N_k^{(j+1)}}+\beta_k^{*-N_k^{(j+1)}}+\beta_k^{-N_k^{(j+1)}}\right)$$
$$+\beta_k^{-l+N_k^{(j+1)}-1}\left(\beta_k^{*N_k^{(j+1)}}+\beta_k^{*-N_k^{(j+1)}}+\beta_k^{N_k^{(j+1)}}\right)\Bigg],$$

which is exactly of the form given by (4.45b) and (4.45c). This proves the second line of (4.45a).

5. By Items 1–4 above, the impulse response of $K(z)$ is $b(l, N_k^{(j+1)}, \nu_k^{(1)}\nu_k^{(2)}\widehat{\gamma}_k \prod_{i=1}^{j}\varphi_k^{(i)})$. Since the impulse response of $G(z)$ is real (it consists of cascade of $\Omega_k^{(j)}(z)$'s, all having real impulse responses according to (4.42f)), by Proposition 2, the impulse response of $H(z)$ is Re $\left\{b(l, N_k^{(j+1)}, \nu_k^{(1)}\nu_k^{(2)}\widehat{\gamma}_k \prod_{i=1}^{j}\varphi_k^{(i)}\right\}$, as required.

B.43 Derivation of (4.50b) **and** (4.51b)

To appreciate these equations, an example is in order. Assuming $m_k^{(a)}$=2 and $m_k^{(b)}$=3, by (4.25b), the order of $\Upsilon_k^{(a)}(z)$ is found to be $2m_k^{(a)}-2=2$. This implies that the input signal $x(n)$ is parsed into blocks of length $2m_k^{(a)}-1=3$.

First, we consider the upper branch of the structure. The upper demultiplexer has been denoted by Up, the middle one by Mi and the lower one by Lo. The following table indicates the samples fed to the upper branch at each time instant n. The table also signifies which demultiplexer feeds the sample in question.

Time n	0	1	2	3	4	5	6	7	8	9	10	11	12	...
Up	$x(0)$	$x(1)$	$x(2)$	–	–	–	$x(6)$	$x(7)$	$x(8)$	–	–	–	$x(12)$	$x(13)$
Mi	–	–	$x(0)$	$x(1)$	$x(2)$	–	–	–	$x(6)$	$x(7)$	$x(8)$	–	–	–
Lo	–	–	–	–	$x(0)$	$x(1)$	–	–	–	–	$x(6)$	$x(7)$	–	–

Table B.1: The flow of the input samples in the upper branch.

Obviously the upper branch is in charge of filtering the samples $x(0), x(1), x(2), x(6), x(7), x(8), x(12), x(13), x(14), \ldots$, and the lower branch is in charge of filtering the samples $x(3), x(4), x(5), x(9), x(10), x(11), x(15), x(16), x(17), \ldots$. The time instance $n = 5$ is very important, since then, the output will be computed, and at the same time, the memory elements will be set to zero before the beginning of a new cycle, that is, the time instance $n = 6$. Note that ideally, the output at time $n = 5$ should be zero, since the order of the FIR filter $\Upsilon_k^{(a)}(z)$ is 2. Filtering of 3 samples (such as $x(0), x(1), x(2)$) by such filter must yield zero at $n = 5$. That explains why the memory (delay) elements are set to zero at such time instant.

Based on the above-mentioned example, the effects of the recursive filter on the input noise $x(0) \equiv \Xi_{k,\text{in}}^{(1)}(0)$ (as given by (3.59a)) is considered next. $x(0)$ is filtered by the recursive IIR filter

$$H(z) = \frac{1}{1 - (\beta_k + 1/\beta_k)z^{-1} + z^{-2}}, \tag{B.27}$$

which defines the difference equation

$$y[n] = x[n] + \beta y[n-1] - y[n-2], \tag{B.28}$$

where

$$\beta \equiv \beta_k + 1/\beta_k. \tag{B.29}$$

Based on (B.28), for this noise input sample ($x(0)$), the output of the recursive filter of the

upper branch $\Xi^{(1)}_{k,\text{out}}$ between the time instants $n = 0$ and $n = 5$ is given in Table B.2.

n	0	1	2	3	4	5 (resetting time)
$\Xi^{(1)}_{k,\text{out}}$	$x(0)$	$\beta x(0)$	$\beta^2 x(0) - x(0)$	$\beta^3 x(0) - 2\beta x(0)$	$\beta^4 x(0) - 3\beta^2 x(0) + x(0)$	0

Table B.2: The output of the recursive filter of the upper branch for $n = 0 \cdots n = 5$.

As noted from Table B.2, at time $n = 5$, the filter is reset, implying that the cumulative (recursive) effect of the filter on the noise sample $\Xi^{(1)}_{k,\text{in}}(0)$ has been removed (in fact this cancellation applies to other noises in the filter, that is, $\Xi^{(1)}_{k,\text{in}}(1)$ and $\Xi^{(1)}_{k,\text{in}}(2)$. Note that $\Xi^{(1)}_{k,\text{in}}(3)$, $\Xi^{(1)}_{k,\text{in}}(4)$ or $\Xi^{(1)}_{k,\text{in}}(5)$ are not considered here, since the upper demultiplexer is feeding the lower recursive filter at those time instances).

The next cycle starts at $n = 6$ and ends at $n = 11$, following exactly the same pattern as that for $n = 0$ up to $n = 5$. The output noise is zero at $n = 5$, $n = 11$, etc, justifying (4.50a). Compared to the input noise, the output noise is equally amplified at $n = 0$ and $n = 6$, $n = 1$ and $n = 7$, etc. This justifies the periodicity of (4.50b), and also explains why only the samples $\lambda \equiv n$, $0 \leq \lambda \leq 4m^{(a)}_k - 4$ are considered. (In the above table, this corresponds to the time instances $0 \ldots 4$).

Finally the validity of (4.50b) for each cycle defined above is illustrated. The impulse response of the $H(z)$ as given by (B.27) is [76]

$$h(n) = \frac{\beta_k^{n+1} - 1/\beta_k^{n+1}}{\beta_k - 1/\beta_k}, \quad n \geq 0. \tag{B.30}$$

The output noise is the convolution sum of the input noise $\Xi^{(1)}_{k,\text{in}}(2\rho(2n^{(a)}_k - 1) + n)$ with (B.30), which is exactly (4.50b) for the cycle $0 \leq i \leq n$ (note that in (4.50b), $\lambda \equiv n$). In other words, (4.50b) can be interpreted as multiplying the latest input noise $\Xi^{(1)}_{k,\text{in}}(2\rho(2n^{(a)}_k -$

$1) + \lambda)$ by

$$\frac{\beta_k^{\lambda+1-\lambda} - 1/\beta_k^{\lambda+1-\lambda}}{\beta_k - 1/\beta_k} = 1,$$

the previous one $i = \lambda - 1 \rightarrow \Xi_{k,\text{in}}^{(1)}(2\rho(2n_k^{(a)} - 1) + \lambda - 1)$ by

$$\frac{\beta_k^{\lambda+1-\lambda+1} - 1/\beta_k^{\lambda+1-\lambda+1}}{\beta_k - 1/\beta_k} = \beta_k + 1/\beta_k,$$

still the previous one $i = \lambda - 2 \rightarrow \Xi_{k,\text{in}}^{(1)}(2\rho(2n_k^{(a)} - 1) + \lambda - 2)$ by

$$\frac{\beta_k^3 - 1/\beta_k^3}{\beta_k - 1/\beta_k} = \beta_k^2 + 1/\beta_k^2 - 1$$

etc.

By considering the fact that the lower branch lags the upper branch by $2n_k^{(a)} - 1$, (4.51a) and (4.51b) follow from (4.50a) and (4.50b).

B.44 Derivation of (4.52)

The cyclic nature of the output noise indicates that a "mod" operator can facilitate the noise calculation. In the sequel, only the upper branch is considered.

By (4.50a), $\Xi_{k,\text{out}}^{(1)}(l) \equiv \Xi_{k,\text{out}}^{(1)}(2\rho(2n_k^{(a)} - 1) + \lambda)$, therefore $\lambda = l \bmod (4n_k^{(a)} - 2) \equiv (l - j) \bmod (n)$ and $2\rho(2n_k^{(a)} - 1) = l - \lambda$.

By (4.51a), $\Xi_{k,\text{out}}^{(1)}(l) \equiv \Xi_{k,\text{out}}^{(1)}((2\rho - 1)(2n_k^{(a)} - 1) + \lambda)$, therefore $\lambda = (l - 2n_k^{(a)} - 1) \bmod (4n_k^{(a)} - 2) \equiv (l - j) \bmod (n)$ and $(2\rho - 1)(2n_k^{(a)} - 1) = l - \lambda$. In (4.52), the variable "j" denotes the lag between the upper and lower branches. Using j, a single symbol $\Xi_{k,\text{out}}^{(m)}(l, j, n)$ encompasses the noise cycle in both the upper and the lower branches.

B.45 Derivation of (4.53)

In words, (4.53) states that "add the output noises of the first and the second branches, $\Xi_{k,\text{out}}^{(1)}(l) + \Xi_{k,\text{out}}^{(2)}(l)$, using the notation of (4.52), and convolve that by $\Upsilon_k^{(b)}(z)/\mu$, as shown in Fig. 4.3". For instance note that the output at time $l - m_k^{(a)}$ is multiplied by $\zeta(1)/\mu$ (actually

is first multiplied and then delayed). Also note that based on (4.25d), the coefficients $\zeta(i)$ are symmetric.

B.46 Derivation of (4.54a)**–**(4.57) for Fig. 4.3

Taking the upper branch into account, two separate noise sources in the structure are distinguished, i.e., the noise generated by the multipliers before the demultiplexers, and those generated by the upper recursive filter coefficients.

The noise generated by the multipliers before the demultiplexers

Since at different time instants, the upper branch is fed by different number of demultiplexers, the noise generated by the multipliers before the demultiplexers has a time dependent variance. For instance as seen from Table B.1, at time instants $n = 0$ or $n = 1$, there is only one noise input, but at time instants $n = 2$ or $n = 4$, there are two noise inputs. This time dependency of the input noise can be characterized by a variable $c(n)$, which has been considered in more details later. To make the analysis less complicated, for the moment we assume that $\forall n : c(n) = 1$, in other words, it has been assumed that there is only one input noise at each time.

Defining

$$\delta \equiv \alpha_k + \frac{1}{\alpha_k},$$

and denoting the input noise sample to the upper branch at time instant i by e_i, the output noise $\Xi^{(1)}_{k,\text{out}}(n)$ of the upper branch of Fig. 4.3 is presented in Table B.3.

Table B.3 is in fact a generalization of Table B.1, where all the input samples (instead of only e_0) are considered, and is therefore governed by the same difference equation ((B.28)). In other words,

$$E_n = \sum_{i=0}^{n} T_{n-i} e_i, \quad 0 \le n \le 4m_k^{(a)} - 4,$$

n	0	1	2	3	...	$4m_k^{(a)}-3$
$\Xi_{k,\text{out}}^{(1)}(n)$	$E_0=e_0$	$E_1=e_1+\delta E_0$ $=e_1+\delta e_0$	$E_2=e_2+\delta E_1-E_0$ $=(\delta^2-1)e_0+\delta e_1+e_2$	$E_3=e_3+\delta E_2-E_1$	...	0 (=resetting time)

Table B.3: The output noise of the upper branch of Fig. 4.3, generated by the multipliers before the demultiplexers.

where the coefficients T_n are defined recursively as:

$$T_{-1}=0, T_0=1, \text{ and } T_n=\delta T_{n-1}-T_{n-2}, \tag{B.31}$$

as indicated by the transfer function of the recursive filter $H(z)$, defined in (B.27).

The noise generated by the upper recursive filter

The noise generated by the upper recursive filter lags that generated by the multipliers before the demultiplexers by one sample (since it takes one multiplication operation before this noise is generated in the recursive part of the filter), but otherwise, it is subject to the same recursion. The generation of noise originated by the upper recursive filter is illustrated in Table B.4.

n	0	1	2	3	4	...	$4m_k^{(a)}-3$
$\Xi_{k,\text{out}}^{(1)}(n)$	0 (=zero still in the memory)	$F_1=f_1$	$F_2=f_2+\delta F_1$	$F_3=f_3+\delta F_2-F_1$	$F_4=f_4+\delta F_3-F_2$	...	0 (=resetting time)

Table B.4: Generation of noise originated by the upper recursive filter in Fig 4.3.

Comparing Table A.4 to Table A.3 reveals that $F_n=\sum_{i=1}^{n} T_{n-i}f_i$, where T_i is defined in (B.31). As seen above, the lower bound of index i lags that of E_n by 1.

The total noise

Considering the independency of f_i and e_i, and appreciating the fact that the noise variances of both f_i and e_i is σ_e^2, the total variance of the noise generated at time n in the upper branch is:

$$V_n = \left[\underbrace{\sum_{i=0}^{n} c(i) T_{n-i}^2}_{\text{The part related to } E_i} + \underbrace{\sum_{i=1}^{n} T_{n-i}^2}_{\text{The part related to } F_i} \right] \sigma_e^2$$

$$\overset{k=n-i}{=} \left[\underbrace{\sum_{k=0}^{n} c(k) T_k^2}_{\text{The part related to } E_i} + \underbrace{\sum_{k=0}^{n-1} T_k^2}_{\text{The part related to } F_i} \right] \sigma_e^2, \tag{B.32}$$

where $\sigma_e^2 = \frac{2^{-2b}}{12}$ is the variance of a single noise source. c is no longer a function of i, but rather a function of k.

To find out the coefficient $c(k)$, the example of page 195 is reconsidered. As illustrated by Table B.1, the upper and the lower demultiplexers never feed the upper branch simultaneously. The middle and the upper, and the middle and the lower demultiplexers however overlap. The upper demultiplexer feeds the upper branch for the time slot $0 \leq n \leq 2m_k^{(a)} - 2$, the middle demultiplexer feeds the upper branch for the time slot $m_k^{(a)} \leq n \leq 3m_k^{(a)} - 2$, and the lower does so for the time slot $2m_k^{(a)} \leq n \leq 4m_k^{(a)} - 3$. It means that the simultaneous feeding of the upper branch takes place in the time slots

$$m_k^{(a)} \leq n \leq 2m_k^{(a)} - 2 \tag{B.33a}$$

and

$$2m_k^{(a)} \leq n \leq 3m_k^{(a)} - 2. \tag{B.33b}$$

In the context of Table B.1 and without loss of generality, consider the time instant $\delta(n) \equiv n = 0$, where $\delta(n)$ is given by (4.55), i.e., assume that we are at the beginning of the noise cycle of the upper branch. At $n = 0$, only $x(0)$ is fed to the upper branch and so $c(0) = 1$. Note that $c(0)$ corresponds to the current time and the coefficient T_0 of (B.31). At $n = 1$, only $x(1)$ is fed to the upper branch and so $c(0) = c(1) = 1$. Here, $c(1)$ corresponds to the previous time instant $(n = 0)$ and the coefficient T_1 of (B.31). For $n = 2$, $c(0) = 2$, since two samples ($x(2)$ and $x(0)$) are fed to the upper branch simultaneously, but for instance $c(1) = 1$, since there was only one sample $x(1)$ which was fed to the upper branch at the previous time instant ($n = 1$). By a similar argument, at $n \equiv \delta(n) = 3$, $c(1) = 2$, and at $n \equiv \delta(n) = 4$, $c(0) = c(2) = 2$, and other $c(i)$'s are 1. This means that if $n \equiv \delta(n)$ is larger than the overlap time as given by (B.33), n should be subtracted from the overlap time, yielding the indices that should be regarded twice. In other words,

$$c(n) = \begin{cases} 2, & \delta(n) - 2m_k^{(a)} + 2 \leq n \leq \delta(n) - m_k^{(a)} \\ & \text{or } \delta(n) - 3m_k^{(a)} + 2 \leq n \leq \delta(n) - 2m_k^{(a)}, \\ 1, & \text{otherwise,} \end{cases} \tag{B.34}$$

The output of $\Upsilon_k^{(a)}(z)$

Considering the fact that the variance of the noise at each branch is periodic with the period of $4m_k^{(a)} - 2$, and also the fact that the variance of the noise at time "$n + 2m_k^{(a)} - 1$" for the lower branch is equal to that of the upper branch at time n, the variance of the noise at time "n" at the output of $\Upsilon_k^{(a)}(z)$ is: $V_{\text{tot}} = V_n + V_{n+2m_k^{(a)}-1}$, where V_n is given by (B.32).

The output of $H_k^{(r)}(z)$

The total noise variance at the filter output is given by

$$N_{tot} = \left[\sum_{i=0}^{2m_k^{(b)}-2} {b_i}^2 \right] V_{\text{tot}} + (2m_k^{(b)} - 1)\sigma_e^2,$$

where $b_i = \frac{\zeta(i)}{\mu}$. The term $(2m_k^{(b)} - 1)\sigma_e^2$ is the noise generated by the coefficients of $\Upsilon_k^{(b)}(z)$.

B.47 Derivation of (4.54a)**–**(4.57) for Fig. 4.2

The derivaiton of (4.54a)–(4.57) for Fig. 4.2 follows the same principle as that stated for Fig. 4.3. The only difference is the complex nature of the noise and the coefficients. In what follows, the upper branch is considered first.

The noise generated by the multipliers before the demultiplexers

Ignoring the coefficient $c(n)$ (as for the case of Fig. 4.3), the output noise $\Xi_{k,\text{out}}^{(1)}(n)$ of the upper branch of Fig. 4.2 is presented in Table B.5. The main difference compared to the case of Fig. 4.3 is that now

$$\delta \equiv \beta_k + \frac{1}{\beta_k}. \tag{B.35}$$

n	0	1	2	3	...	$4n_k^{(a)} - 3$
$\Xi_{k,\text{out}}^{(1)}(n)$	$E_0 = e_0$	$E_1 = e_1 + \delta$ $= e_1 + \delta e_0$	$E_2 = e_2 + \delta E_1 - E_0$ $=(\delta^2 - 1)e_0 + \delta e_1 + e_2$	$E_3 = e_3 + \delta E_2 - E_1$	...	0 (=resetting time)

Table B.5: The output noise of the upper branch of Fig. 4.2, generated by the multipliers before the demultiplexers.

As for the case of Fig. 4.3, in general:

$$E_n = \sum_{i=0}^{n} T_{n-i} e_i, \quad 0 \le n \le 4n_k^{(a)} - 4,$$

where the coefficients T_n are defined according to (B.31), with δ as given by (B.35).

With $c(i)$ as defined by (B.34), the real and the imaginary parts of the complex noise E_n are given by

$$\begin{aligned}\text{Re}\left\{E_n\right\} =& \text{Re}\left\{\sum_{i=0}^{n} c(i)T_{n-i}e_i\right\} = \sum_{i=0}^{n} \text{Re}\left\{c(i)T_{n-i}e_i\right\} \\ =& \sum_{i=0}^{n} c(i)\left(\text{Re}\left\{T_{n-i}\right\}\text{Re}\left\{e_i\right\} - \Im\left\{T_{n-i}\right\}\Im\left\{e_i\right\}\right)\end{aligned}$$

and

$$\begin{aligned}\text{Im}\left\{E_n\right\} =& \text{Im}\left\{\sum_{i=0}^{n} c(i)T_{n-i}e_i\right\} = \sum_{i=0}^{n} \text{Im}\left\{c(i)T_{n-i}e_i\right\} \\ =& \sum_{i=0}^{n} c(i)\left(\text{Re}\left\{T_{n-i}\right\}\text{Im}\left\{e_i\right\} + \text{Im}\left\{T_{n-i}\right\}\text{Re}\left\{e_i\right\}\right)\end{aligned}$$

respectively. $\text{Re}\left\{e_i\right\}$ is independent of $\text{Im}\left\{e_i\right\}$, and consequently

$$\text{var}\left(\text{Re}\left\{E_n\right\}\right) = \text{var}\left(\text{Im}\left\{E_n\right\}\right) = \left(\sum_{i=0}^{n} c(i)\left(\text{Re}\left\{T_{n-i}\right\}^2 + \text{Im}\left\{T_{n-i}\right\}^2\right)\right)\sigma_e^2.$$

The noise generated by the upper recursive filter

As shown for the case of Fig. 4.3, the noise generated by the upper recursive filter at any time instant n, denoted by F_n, is basically a delayed version of the noise generated by the multipliers before the demultiplexers. Moreover, contrary to the noise introduced by the demultiplexers, there are no multiple simultaneous noise inputs in this case, that is, $\forall i : c(i) = 1$ and hence

$$\text{var}\left(\text{Re}\left\{F_n\right\}\right) = \text{var}\left(\text{Im}\left\{F_n\right\}\right) = \left(\sum_{i=1}^{n} \text{Re}\left\{T_{n-i}\right\}^2 + \text{Im}\left\{T_{n-i}\right\}^2\right)\sigma_e^2.$$

The total noise

As in the case of Fig. 4.3, considering the independency of the real and the imaginary parts of $f_i\left(n\right)$ and $e_i\left(n\right)$, and following the steps taken in (B.32), the total variance of the noise

generated at time n of the upper branch can be expressed as:

$$V_n = \left(\sum_{k=0}^{n} c(k) \left(\text{Re}\left\{T_k\right\}^2 + \text{Im}\left\{T_k\right\}^2 \right) + \sum_{k=0}^{n-1} \left(\text{Re}\left\{T_k\right\}^2 + \text{Im}\left\{T_k\right\}^2 \right) \right) \sigma_e^2,$$

where the coefficient $c\,(k)$ is given by (B.34).

The output of $\Gamma_k^{(a)}(z)$

By the same argument discussed for $\Upsilon_k^{(a)}(z)$ of Fig. 4.3 (with $m_k^{(a)} \rightarrow n_k^{(a)}$), the variance of the noise at time "n" at the output of $\Gamma_k^{(a)}(z)$ is given by

$$V_{tot,n} = V_n + V_{n+2n_k^{(a)}-1}. \tag{B.36}$$

The output of $H_k^{(c)}(z)$

For calculating the total variance, first note that each complex coefficient $k = 2\eta_k(k)/\nu_k$ is realized as two multipliers, one as the real part of k and the other as the imaginary part of k.

Consider the complex coefficient $k = 2\eta_k(k)/\nu_k$. As seen from Fig. 4.2, in the realization of $H_k^{(c)}(z)$, only $n_k^{(b)}$ of these k's are needed, but the generated noise by these k's is fed into $2n_k^{(b)} - 1$ different adders and hence $2n_k^{(b)} - 1$ complex noise sources should be considered. Since each complex coefficient k is realized as two multipliers (one as the real part of k and the other as the imaginary part of k), the noise variance generated by these $2n_k^{(b)} - 1$ number of k's is equal to $2(2n_k^{(b)} - 1)\sigma_e^2 = (4n_k^{(b)} - 2)\sigma_e^2$.

On the other hand, the complex input noise from $\Gamma_k^{(a)}(z)$ is filtered by the aforementioned k's. This gives rise to the noise $\left[\sum_{i=0}^{2n_k^{(b)}-2} c_i \right] V_{tot,n}$, where $V_{tot,n}$ is given in (B.36) and

$$c_i = \text{Re}^2 \left\{ \frac{2\eta_k(i)}{\nu_k} \right\} + \text{Im}^2 \left\{ \frac{2\eta_k(i)}{\nu_k} \right\} = \left(\text{abs} \left\{ \frac{2\eta_k(i)}{\nu_k} \right\} \right)^2.$$

The total noise variance at the filter output is then

$$T = \left[\sum_{i=0}^{2n_k^{(b)}-2} c_i \right] V_{tot,n} + (4n_k^{(b)} - 2)\sigma_e^2.$$

B.48 Derivation of (4.59)

To appreciate (4.59), an example is first in order. Assuming $n_k^{(a)}$=2 and $n_k^{(b)} = 3$, by (4.42a) the order of $\Omega_k^{(a)}$ will be $4n_k^{(a)} = 8$, and consequently, the order of the FIR filter $\Gamma_k^{(a)}(z)$ as given by (4.44b) is found to be $4n_k^{(a)} - 2 = 6$. This implies that the input signal $x(n)$ is parsed into blocks of length $4n_k^{(a)} - 1 = 7$.

First, the upper branch of $\nu_k^{(1)}\nu_k^{(2)}\Gamma_k^{(a)}(z)$ as depicted in Fig. 4.8 is considered. The demultiplexers have been denoted by $M1$,...,$M5$, with $M1$ being the uppermost and $M5$ being the lowermost demultiplexer. The following table indicates the input samples ($x(n)$) fed to the upper branch at each time instant n. The table also signifies which demultiplexer feeds the sample in question.

Time n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	...
$M1$	$x(0)$	$x(1)$	$x(2)$	$x(3)$	$x(4)$	$x(5)$	$x(6)$	-	-	-	-	-	-	-	$x(14)$	...
$M2$	-	-	$x(0)$	$x(1)$	$x(2)$	$x(3)$	$x(4)$	$x(5)$	$x(6)$	-	-	-	-	-	-	...
$M3$	-	-	-	-	$x(0)$	$x(1)$	$x(2)$	$x(3)$	$x(4)$	$x(5)$	$x(6)$	-	-	-	-	...
$M4$	-	-	-	-	-	-	$x(0)$	$x(1)$	$x(2)$	$x(3)$	$x(4)$	$x(5)$	$x(6)$	-	-	...
$M5$	-	-	-	-	-	-	-	-	$x(0)$	$x(1)$	$x(2)$	$x(3)$	$x(4)$	$x(5)$	-	...

Table B.6: The flow of the input samples in the upper branch of Fig. 4.8.

Obviously the upper branch is in charge of filtering the samples $x(0), x(1), \cdots, x(6)$, $x(14), x(15), \cdots$, and the lower branch is in charge of filtering the samples $x(7), x(8), \cdots$, $x(13), x(21), x(22), \cdots$. The time instance $n = 13$ is very important, since then, the output will be computed, and at the same time, the memory elements will be set to zero before the

beginning of a new cycle, that is, the time instance $n = 14$. Note that ideally, the output at time $n = 13$ should be zero, since as mentioned above, the order of the FIR filter $\Gamma_k^{(a)}(z)$ is 6. Filtering of 7 samples (such as $x(0), x(1), \cdots, x(6)$) by such filter must yield zero at $n = 13$. That explains why the memory (delay) elements are set to zero at such time instant.

For both structures of Fig. 4.2 and Fig. 4.8, the noise introduced by the demultiplexers ($\Xi_{k,\text{in}}^{(1)}(n)$) is subject to the same recursive filter. This implies that (4.52a) and (4.52b) apply as such to the structure of Fig. 4.8, with $m = 1, j = 0, n = 8n_k^{(a)} - 2$ for the upper branch and $m = 2, j = 4n_k^{(a)} - 1, n = 8n_k^{(a)} - 2$ for the lower branch[1].

Using the notation of (4.52a), (4.59) states that "add the output noises at the output of the upper and the lower recursive filters, multiply this sum by $\nu_k^{(2)}\widehat{\gamma}_k$ and take the real part of it, and finally convolve that by the filter $2\Omega_k^{(b)}(z)$, as given by (4.43). In other words, the noise at the output of $\Gamma_k^{(a)}(z)$ as given by (4.44b) and depicted in Fig. 4.8 should be convolved by $\Omega_k^{(b)}(z)$ to yield the overall noise. This is another manifestation of the fact that by (4.40b), (4.41c), and (4.44b), $H_k^{(c)}(z) = \Gamma_k^{(a)}(z)\Omega_k^{(b)}(z)$.

B.49 Derivation of (4.60)**–**(4.61) for Fig. 4.8

Taking the upper branch into account, two separate noise sources in the structure are distinguished, i.e., the noise generated by the multipliers before the demultiplexers, and those generated by the upper recursive filter coefficients.

[1]As already mentioned on page 198, in $\Xi_{k,\text{out}}^{(1)}(l, j, n)$, the variable l denotes the time instant, the variable $m = 1 (m = 2)$ denotes the upper (lower) branch, the variable n denotes the lag between successive resetting of the recursive filters, and the variable j denotes the lag between the resetting time of the upper- and lower recursive filters.

The noise generated by the multipliers before the demultiplexers

Since at different time instants, the upper branch is fed by different number of demultiplexers, the noise generated by the multipliers before the demultiplexers has a time dependent variance. For instance as seen from Table B.6, at time instants $n = 0$ or $n = 1$, there is only one noise input, but at time instants $n = 2$, $n = 4$ and $n = 6$, there are two, three and four noise inputs respectively. This time dependency of the input noise can be characterized by a variable $c(n)$, which has been considered in more details later. To make the analysis less complicated, for the moment we assume that $\forall n : c(n) = 1$, in other words, it has been assumed that there is only one input noise at each time. Denoting the input noise sample to the upper branch at time instant i by e_i and defining δ as given by (B.35), the output noise $\Xi^{(1)}_{k,\text{out}}(n)$ of the upper branch of Fig. 4.8 is presented in Table B.7.

n	0	1	2	3	...	$8n_k^{(a)} - 3$
$\Xi^{(1)}_{k,\text{out}}(n)$	$E_0 = e_0$	$E_1 = e_1 + \delta E_0$ $= e_1 + \delta e_0$	$E_2 = e_2 + \delta E_1 - E_0$ $=(\delta^2 - 1)e_0 + \delta e_1 + e_2$	$E_3 = e_3 + \delta E_2 - E_1$	...	0 (=resetting time)

Table B.7: The output noise of the upper branch of Fig. 4.8, generated by the multipliers before the demultiplexers.

The results presented in Table B.7 are identical to those presented in Table B.5 on page 203 and therefore in Table B.7,

$$E_n = \sum_{i=0}^{n} T_{n-i} e_i, \quad 0 \leq n \leq 8n_k^{(a)} - 4, \tag{B.37}$$

where the coefficients T_n are defined according to (B.31).

Since the noise E_n is complex, and appreciating the fact that e_i is real, the real and the

imaginary parts of E_n are

$$\begin{aligned}\text{Re}\left\{E_n\right\} =& \text{Re}\left\{\sum_{i=0}^{n} c(i)T_{n-i}e_i\right\} = \sum_{i=0}^{n} \text{Re}\left\{c(i)T_{n-i}e_i\right\} \\ &= \sum_{i=0}^{n} c(i)\left(\text{Re}\left\{T_{n-i}\right\}\text{Re}\left\{e_i\right\} - \text{Im}\left\{T_{n-i}\right\}\underbrace{\text{Im}\left\{e_i\right\}}_{=0}\right) \\ &= \sum_{i=0}^{n} c(i)\left(\text{Re}\left\{T_{n-i}\right\}e_i\right)\end{aligned}$$

and

$$\begin{aligned}\text{Im}\left\{E_n\right\} =& \text{Im}\left\{\sum_{i=0}^{n} c(i)T_{n-i}e_i\right\} = \sum_{i=0}^{n} \text{Im}\left\{c(i)T_{n-i}e_i\right\} \\ &= \sum_{i=0}^{n} c(i)\left(\text{Re}\left\{T_{n-i}\right\}\underbrace{\text{Im}\left\{e_i\right\}}_{=0} + \text{Im}\left\{T_{n-i}\right\}\text{Re}\left\{e_i\right\}\right) = \sum_{i=0}^{n} c(i)\left(\text{Im}\left\{T_{n-i}\right\}e_i\right) \\ &= \sum_{i=0}^{n-1} c(i)\left(\text{Im}\left\{T_{n-i}\right\}e_i\right). \qquad \text{(B.38)}\end{aligned}$$

respectively. The last equality of (B.38) stems from the fact that $\text{Im}\left\{T_0\right\} = 0$.

All the samples of $\text{Re}\left\{e_i\right\}$ are independent of those of $\text{Im}\left\{e_i\right\}$, therefore

$$\text{var}\left(\text{Re}\left\{E_n\right\}\right) = \left(\sum_{i=0}^{n} c(i)\text{Re}\left\{T_{n-i}\right\}^2\right)\sigma_e^2,$$

and

$$\text{var}\left(\text{Im}\left\{E_n\right\}\right) = \left(\sum_{i=0}^{n-1} c(i)\text{Im}\left\{T_{n-i}\right\}^2\right)\sigma_e^2.$$

The noise generated by the upper recursive filter

Since the recursive filter of Fig. 4.8 is identical to that of Fig. 4.2, the analysis presented on page 204 applies here as such, and consequently

$$\text{var}\left(\text{Re}\left\{F_n\right\}\right) = \text{var}\left(\text{Im}\left\{F_n\right\}\right) = \left(\sum_{i=1}^{n} \text{Re}\left\{T_{n-i}\right\}^2 + \text{Im}\left\{T_{n-i}\right\}^2\right)\sigma_e^2.$$

The total noise

Considering the independency of the real and the imaginary parts of the complex noise $f_i(n)$ and the real noise $e_i(n)$, and appreciating the fact that the noise variance of Re$\{f_i\}$, Im$\{f_i\}$ and e_i is σ_e^2, the total variance of the real and imaginary noise generated at the time instant n in the upper branch are given by

$$\begin{aligned} V_{\text{real},n} &= \left(\sum_{i=1}^{n} \text{Re}\left\{T_{n-i}\right\}^2 + \text{Im}\left\{T_{n-i}\right\}^2 + \sum_{i=0}^{n} c(i)\text{Re}\left\{T_{n-i}\right\}^2 \right) \sigma_e^2 \\ &\overset{k=n-i}{=} \left(\sum_{k=0}^{n-1} \text{Re}\left\{T_k\right\}^2 + \text{Im}\left\{T_k\right\}^2 + \sum_{k=0}^{n} c(k)\text{Re}\left\{T_k\right\}^2 \right) \sigma_e^2 \end{aligned} \tag{B.39}$$

and

$$\begin{aligned} V_{\text{imag},n} &= \left(\sum_{i=1}^{n} \text{Re}\left\{T_{n-i}\right\}^2 + \text{Im}\left\{T_{n-i}\right\}^2 + \sum_{i=0}^{n-1} c(i)\text{Im}\left\{T_{n-i}\right\}^2 \right) \sigma_e^2 \\ &\overset{k=n-i}{=} \left(\sum_{k=0}^{n-1} \text{Re}\left\{T_k\right\}^2 + \text{Im}\left\{T_k\right\}^2 + \sum_{k=1}^{n} c(k)\text{Im}\left\{T_k\right\}^2 \right) \sigma_e^2 \end{aligned} \tag{B.40}$$

respectively.

To find out the coefficient $c(k)$, the example of page 173 is reconsidered. As illustrated by Table B.6, the time slot in which the upper branch is fed by the demultiplexers $M1, M2, M3, M4$ and $M5$ is $0 \leq n \leq 4n_k^{(a)} - 2$, $n_k^{(a)} \leq n \leq 5n_k^{(a)} - 2$, $2n_k^{(a)} \leq n \leq 6n_k^{(a)} - 2$, $3n_k^{(a)} \leq n \leq 7n_k^{(a)} - 2$ and $4n_k^{(a)} \leq n \leq 8n_k^{(a)} - 2$ respectively. Consequently the number of overlap at different time slots can be expressed according to the Table B.8.

In the context of Table B.6 and without loss of generality, consider the time instant $\delta(n) \equiv n = 0$, where $\delta(n)$ is given by (4.60c), i.e., assume that we are at the beginning of the noise cycle of the upper branch. At $n = 0$, only $x(0)$ is fed to the upper branch and so $c(0) = 1$. Note that $c(0)$ corresponds to the current time and the coefficient T_0 of

Time n	Number of overlaps	Number of input noise samples
$0 \leq n \leq n_k^{(a)} - 1$	0	1
$n_k^{(a)} \leq n \leq 2n_k^{(a)} - 1$	1	2
$2n_k^{(a)} \leq n \leq 3n_k^{(a)} - 1$	2	3
$3n_k^{(a)} \leq n \leq 4n_k^{(a)} - 2$	3	4
$4n_k^{(a)} - 1$	2	3
$4n_k^{(a)} \leq n \leq 5n_k^{(a)} - 2$	3	4
$5n_k^{(a)} - 1 \leq n \leq 6n_k^{(a)} - 2$	2	3
$6n_k^{(a)} - 1 \leq n \leq 7n_k^{(a)} - 2$	1	2
$7n_k^{(a)} - 1 \leq n \leq 8n_k^{(a)} - 3$	0	1

Table B.8: The number of overlapping noise samples at different time slots at the output of the recursive filter of Fig. 4.8.

(B.37). At $n = 1$, only $x(1)$ is fed to the upper branch and so $c(0) = c(1) = 1$. Here, $c(1)$ corresponds to the previous time instant $(n = 0)$ and the coefficient T_1 of (B.37). For $n = 2$, $c(0) = 2$, since two samples ($x(2)$ and $x(0)$) are fed to the upper branch simultaneously, but for instance $c(1) = 1$, since there was only one sample $x(1)$ which was fed to the upper branch at the previous time instant ($n = 1$). By a similar argument, at $n \equiv \delta(n) = 3$, $c(0) = c(1) = 2$, and at $n \equiv \delta(n) = 6$, $c(0) = 4, c(1) = c(2) = 3, c(3) = c(4) = 2$, and $c(1) = c(0) = 1$. This means that if $n \equiv \delta(n)$ is larger than the overlap time as given by Table B.8, n should be subtracted from the overlap time, yielding the indices that are regarded repeatedly. In other words,

$$c(n) = \begin{cases} 2, & n_k^{(a)} \leq \delta(n) - k \leq 2n_k^{(a)} - 1 \text{ or } 6n_k^{(a)} - 1 \leq \delta(n) - k \leq 7n_k^{(a)} - 2, \\ 3, & 2n_k^{(a)} \leq \delta(n) - k \leq 3n_k^{(a)} - 1 \text{ or } \delta(n) - k = 4n_k^{(a)} - 1, \\ 4, & 3n_k^{(a)} \leq \delta(n) - k \leq 4n_k^{(a)} - 2 \text{ or } 4n_k^{(a)} \leq \delta(n) - k \leq 5n_k^{(a)} - 2, \\ 1, & \text{otherwise.} \end{cases} \tag{B.41}$$

The output of $\Gamma_k^{(a)}(z)$ before the multiplier $\nu_k^{(2)}\widehat{\gamma}_k$

The complex multiplier $\nu_k^{(2)}\widehat{\gamma}_k$ consists of two real multipliers, Re $\left\{\nu_k^{(2)}\widehat{\gamma}_k\right\}$ and Im $\left\{\nu_k^{(2)}\widehat{\gamma}_k\right\}$. If the real and imaginary inputs to this multiplier are denoted by R_{in} and I_{in} respectively, the output after this multiplier will be given by

$$R_{\text{in}}\text{Re}\left\{\nu_k^{(2)}\widehat{\gamma}_k\right\} - I_{\text{in}}\text{Im}\left\{\nu_k^{(2)}\widehat{\gamma}_k\right\}.$$

If the variances of R_{in} and I_{in} at any arbitrary time n are denoted by $V_{R,n}$ and $V_{I,n}$ respectively, the variance of the output after $\nu_k^{(2)}\widehat{\gamma}_k$ at any arbitrary time n denoted by $V_{\text{out},n}$ will be:

$$V_{\text{out},n} = V_{R,n}\left(\text{Re}\left\{\nu_k^{(2)}\widehat{\gamma}_k\right\}\right)^2 + V_{I,n}\left(\text{Im}\left\{\nu_k^{(2)}\widehat{\gamma}_k\right\}\right)^2 + 2\sigma_e^2. \tag{B.42}$$

The term $2\sigma_e^2$ in (B.42) is generated by the two real multipliers of $\nu_k^{(2)}\widehat{\gamma}_k$.

Since the variance of the noise at each branch is periodic with the period of $8n_k^{(a)} - 2$, and also considering the fact that the variance of the noise at the time instant $n + 4n_k^{(a)} - 1$ for the lower branch is equal to that of the upper branch at time n (and also the fact that the noises in the upper and the lower branches are independent), $V_{R,n}$ and $V_{I,n}$ in (B.42) are found to be

$$V_{R,n} = V_{\text{real},n} + V_{\text{real},n+4n_k^{(a)}-1}$$

and

$$V_{I,n} = V_{\text{imag},n} + V_{\text{imag},n+4n_k^{(a)}-1},$$

where $V_{\text{imag},n}$ and $V_{\text{real},n}$ are defined in (B.39) and (B.40) respectively.

The output of $H_k^{(c)}(z)$

Once the transient time of $H_k^{(c)}(z)$ has elapsed, the total noise variance at its output using (4.43) can be calculated to be:

$$V_{\text{tot}} = \left[\sum_{i=0}^{4n_k^{(b)}-4} {C_k}^2 \right] V_{\text{out},n} + (4n_k^{(b)} - 5)\sigma_e^2,$$

where

$$C_k = \frac{2\eta_k^{(b)}}{\nu_k^{(1)}\nu_k^{(2)}}. \tag{B.43}$$

The reason for the second term $(4n_k^{(b)} - 5)\sigma_e^2$ is that there are $4n_k^{(b)} - 3$ coefficients in the filter $\Omega_k^{(b)}(z)$, as indicated by (4.43). But according to (4.42f) and (4.42g), two of these coefficients are equal to 1, and therefore they generate no quantization noise. Hence there are $4n_k^{(b)} - 5$ noise generating coefficients left.

The pre-multiplier "2" in (B.43) comes from (4.44a), and the denominator in (B.43) accounts for the scaling coefficients introduced in $\Gamma_k^{(a)}(z)$.

B.50 Derivation of (4.48)

$$\begin{aligned}
&\Re\left\{ \frac{\widehat{\gamma}_k}{1 - \left(\beta_k + \beta_k^{-1}\right) z^{-1} + z^{-2}} \right\} \\
=&\frac{1}{2}\left[\frac{\widehat{\gamma}_k}{\left(1 - \beta_k z^{-1}\right)\left(1 - \beta_k^{-1} z^{-1}\right)} + \frac{\widehat{\gamma}_k^*}{\left(1 - \beta_k^* z^{-1}\right)\left(1 - \beta_k^{*-1} z^{-1}\right)} \right] \\
=&\frac{\Re\left\{ \widehat{\gamma}_k \left(1 - \left(\beta_k + \beta_k^{*-1}\right) z^{-1} + z^{-2}\right) \right\}}{\left(1 - \beta_k z^{-1}\right)\left(1 - \beta_k^{-1} z^{-1}\right)\left(1 - \beta_k^* z^{-1}\right)\left(1 - \beta_k^{*-1} z^{-1}\right)}
\end{aligned} \tag{B.44}$$

According to (4.41c) (with $n_k^{(a)} = 1.n_k^{(a)}$), $\Omega_k^{(a)}(z)$ can be decomposed as

$$\Omega_k^{(a)}(z) = \Omega_k^{(m)}(z)\Omega_k^{(n)}(z), \tag{B.45}$$

where

$$\begin{aligned}
\Omega_k^{(m)}(z) = &\left(1 + z^{-4}\right) - \left(\beta_k + \beta_k^* + \beta_k^{-1} + \beta_k^{*-1}\right)\left(z^{-1} + z^{-3}\right) \\
&+ \left(2 + \left(\beta_k + \beta_k^{-1}\right)\left(\beta_k^* + \beta_k^{*-1}\right)\right) z^{-2}
\end{aligned} \tag{B.46}$$

and

$$\begin{aligned}\Omega_k^{(n)}(z) &= \sum_{l=0}^{4n_k^{(1)}-4} \eta_k^{(1)}(l) z^{-lN_k^{(1)}} \\ &= \sum_{l=0}^{4n_k^{(a)}-4} \eta_k^{(a)}(l) z^{-l},\end{aligned}$$

The last equality follows from the fact that by (4.42e), for $n_k^{(a)} = 1$ and $n_k^{(1)} = n_k^{(a)}$, $N_k^{(1)} = 1$. The notation $\eta_k^{(1)} \equiv \eta_k^{(a)}$ has been used to emphasize that $n_k^{(1)} \equiv n_k^{(a)}$. ($n_k^{(1)}$ is used in (4.42g) to calculate $\eta_k^{(1)}$).

$\Omega_k^{(m)}(z)$ is equal to the denominator of (B.44), so

$$\Omega_k^{(m)}(z)\Re\left\{\frac{\widehat{\gamma}_k}{1-\left(\beta_k+\beta_k^{-1}\right)z^{-1}+z^{-2}}\right\} = \Re\left\{\widehat{\gamma}_k\left(1-\left(\beta_k+\beta_k^{*-1}\right)z^{-1}+z^{-2}\right)\right\}. \tag{B.47}$$

Plugging (B.45) in (4.44b) and using (B.47) yields

$$\begin{aligned}\Gamma_k^{(a)}(z) =& \Omega_k^{(n)}\Omega_k^{(m)}(z)\Re\left\{\frac{\widehat{\gamma}_k}{1-\left(\beta_k+\beta_k^{-1}\right)z^{-1}+z^{-2}}\right\} \\ =& \Omega_k^{(n)}(z)\Re\left\{\widehat{\gamma}_k\left(1-\left(\beta_k^{*}+\beta_k^{*-1}\right)z^{-1}+z^{-2}\right)\right\}.\end{aligned}$$

Ignoring the scaling coefficients $\nu_k^{(1)}$ and $\nu_k^{(2)}$, the above equation is exactly what is given in (4.48).

B.51 Derivation of (4.49)

If $n_k^{(j)} = 2$, then by (4.42b)

$$\Omega_k^{(j)}(z) = \sum_{l=0}^{4} \eta_k^{(j)}(l) z^{-lN_k^{(j)}}.$$

By (4.42f), the impulse response defined by $\Omega_k^{(j)}(z)$ is symmetric, and therefore, $\eta_k^{(j)}(0) = \eta_k^{(j)}(4)$ and $\eta_k^{(j)}(1) = \eta_k^{(j)}(3)$. Hence

$$\Omega_k^{(j)}(z) = \eta_k^{(j)}(0)\left[1+z^{-4N_k^{(j)}}\right] + \eta_k^{(j)}(1)\left[z^{-N_k^{(j)}}+z^{-3N_k^{(j)}}\right] + \eta_k^{(j)}(2)z^{-2N_k^{(j)}},$$

but by (4.42f) and (4.42g), $v_k(0) = \eta_k^{(j)}(0) = 1$, and (4.49) follows.

Bibliography

[1] J. W. Adams and A. N. Willson, Jr., "A new approach to FIR digital filters with fewer multipliers and reduced sensitivity," *IEEE Trans. Circuits Syst.*, vol. 30, no. 5, pp. 277–283, May 1983.

[2] ——, "A novel approach to the design of efficient FIR digital bandpass filters," in *Proc. IEEE Int. Symp. Circuits Syst.*, Montral, Canada, May 1984, pp. 28–32.

[3] ——, "Some efficient digital prefilter structures," *IEEE Trans. Circuits Syst.*, vol. 31, no. 3, pp. 260–266, Mar 1984.

[4] A. N. Akansu and R. A. Haddad, *Multiresolution signal decomposition : transforms, subbands and wavelets*. Academic Press, 1992.

[5] V. R. Algazi and M. Suk, "On the frequency weighted least squares design of finite duration filters," *IEEE Trans. Circuits Syst.*, vol. 22, no. 12, pp. 943–953, Dec. 1975.

[6] V. R. Algazi, M. Suk, and C. S. Rim, "Design of almost minimax FIR filters in one and two dimensions by WLS techniques," *IEEE Trans. Circuits Syst.*, vol. 33, no. 6, pp. 590–596, June 1986.

[7] P. Arian and T. Saramäki, "A systematic technique for optimizing one-stage two-filter linear-phase FIR filters for sampling rate conversion," in *Proc. IEEE Int. Symp. on Circuits and Syst.*, vol. 4, Bangkok, Thailand, May 2003, pp. 197–200.

[8] P. Arian, T. Saramäki, and A. T. Fam, "Design of IIR-like filter blocks in cascade for generating narrow transition-band linear-phase FIR filters," in *Proc. IEEE TENCON' 2004*, vol. A, Chiang Mai, Thailand, November 2004, pp. 60–63.

[9] ——, "Generalized linear-phase FIR filters based on switching and resetting of IIR filters," in *Proc. Int. Conf. on Signals and Elec. Syst. (ICSES'04)*, Poznan, Poland, Sept. 2004, pp. 39–42.

[10] ——, "Implementation of IIR-like filter blocks in cascade for generating narrow transition-band linear-phase FIR filters," in *Proc. IEEE TENCON' 2004*, vol. A, Chiang Mai, Thailand, November 2004, pp. 64–67.

[11] ——, "A new class of linear-phase FIR filters based on switching and resetting of IIR filters," in *Proc. IEEE NORCHIP 2004*, Oslo, Norway, Nov. 2004, pp. 167–170.

[12] ——, "An alternative approach for generating linear-phase FIR filters from IIR-like filter blocks in cascade," in *3rd International IEEE Northeast Workshop on Circuits and Systems (IEEE-NEWCAS'05)*, Qubec City, Canada, June 2005, pp. 349–352.

[13] ——, "A decomposition technique for cascaded IIR-like filter blocks generating linear-phase FIR filters," in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 6, Kobe, Japan, May 2005, pp. 2008–2011.

[14] ——, "Unstable IIR-like filter blocks for generating linear-phase FIR filters: A noise analysis," in *Proc. IEEE TENCON' 2005*, Melbourne, Australia, November 2005.

[15] P. Arian, T. Saramäki, and S. K. Mitra, "A systematic technique for optimizing multiple branch FIR filters for sampling rate conversion," in *Proc. IEEE Int. Symp. on Circuits and Syst.*, vol. 4, Scotsdale, USA, May 2002, pp. 1–4.

[16] ——, "Improving multiple-branch FIR filter structures through constrained optimization," in *Proc. IEEE NORCHIP 2004*, Oslo, Norway, Nov. 2004, pp. 163–166.

[17] M. G. Bellanger, G. Bonnerot, and M. Coudreuse, "Digital filtering by polyphase network: Application to sample-rate alteration and filter banks," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 24, pp. 109–114, Apr. 1976.

[18] M. G. Bellanger, J. L. Daguet, and G. P. Lepagno, "Interpolation, extrapolation, and reduction of computation speed in digital filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 22, pp. 231–235, Aug. 1974.

[19] B. P. Bogert, "Demonstration of delay destortion correction by time-reversal techniques," *IRE Trans. Commun. Syst.*, vol. CS-5, pp. 2–7, Dec. 1957.

[20] G. F. Boudreaux and T. W. Parks, "Thinning digital filters: A piecewise exponential approach," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 31, no. 1, pp. 105–112, Feb. 1983.

[21] C. S. Burrus, "Multiband least squares FIR filter design," *IEEE Trans. Signal Processing*, vol. 43, no. 2, pp. 412–421, Feb. 1995.

[22] C. S. Burrus, A. W. Soewito, and R. A. Gopinath, "Least squares error FIR filter design with transition bands," *IEEE Trans. Signal Processing*, vol. 40, no. 6, pp. 1327–1340, June 1992.

[23] T. G. Campbell and Y. Neuvo, "A family of predictive FIR filters with low computational complexity," in *Proc. 30th Midwest Symp. Circuits Syst.*, Syracuse, NY, Aug. 1987, pp. 60–63.

[24] S. Chu and C. S. Burrus, "Optimum FIR and IIR multistage multirate filter design," *Circuits, Systems and Signal Processing*, vol. 2, no. 3, pp. 361–386, 1983.

[25] ——, "Efficient recursive realization of FIR filters: Part 1," *Circuits, Syst., Signal Processing*, vol. 3, no. 1, pp. 3–20, 1984.

[26] ——, "Efficient recursive realization of FIR filters: Part 2," *Circuits, Syst., Signal Processing*, vol. 3, no. 1, pp. 21–57, 1984.

[27] ——, "Multirate filter designs using comb filters," *IEEE Trans. Circuits Syst.*, vol. 31, no. 11, pp. 913–924, Nov. 1984.

[28] T. Coleman, M. A. Branch, and A. Grace, *Optimization Toolbox User's Guide*. The MathWorks Inc., 1999.

[29] R. E. Crochiere and L. R. Rabiner, "Optimum FIR digital filter implementations for decimation, interpolation, and narrow-band filtering," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 23, no. 5, pp. 444–456, Oct. 1975.

[30] ——, "Further considerations in the design of decimators and interpolators," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 24, no. 4, pp. 296–311, Aug. 1976.

[31] ——, *Multirate Digital Signal Processing*. Prentice-Hall, Inc., 1983.

[32] A. W. Crooke and J. W. Craig, "Digital filters for sample rate reduction,," *IEEE Trans. Audio Electroacoust.*, vol. 20, 1972.

[33] R. Czarnach, "Recursive processing by noncausal digital filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 30, no. 3, pp. 363–370, June 1982.

[34] R. Czarnach, H. W. Schüssler, and G. Röhrlein, "Linear phase recursive digital filters for special applications," in *Proc. Int. Conf. Acoust., Speech, and Signal Processing*, vol. 7, May 1982, pp. 1825–1828.

[35] B. Djokić, M. Popović, and M. Lutovac, "A new improvement to the Powell and Chau linear phase IIR filters," *IEEE Trans. Signal Processing*, vol. 46, no. 6, pp. 1685–1688, June 1998.

[36] A. T. Fam, "MFIR filters: Properties and applications," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 29, no. 6, pp. 1128–1136, Dec. 1981.

[37] ——, "Digital interpolation based on functional interation," in *Proc. Eighteenth Annu. Asilomar Conf. Circuits, Syst. Comput.*, Pacific Grove, CA, Nov. 1984, pp. 97–101.

[38] ——, "FIR filters that approach IIR filters in their computational efficiency," in *Proc. Twenty-first Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, Nov. 1987, pp. 28–30.

[39] N. J. Fliege, *Multirate Signal Processing*. Chicester, John Wiley and Sons, 1994.

[40] J. Földvári-Orosz, T. Henk, and E. Simonyi, "Simultaneous amplitude and phase approximation for lumped and sampled filters," *Int. J. Circuit Theory Applicat.*, vol. 19, pp. 77–100, 1991.

[41] S. L. Freeney, R. B. Kieburtz, K. V. Mina, and S. K. Tewksbury, "Design of digital filters for an all digital frequency division multiplex-time division multiplex translator," *IEEE Trans. circuit Theory (Special Issue on Active and Digital Networks)*, vol. CT-18, pp. 702–711, Nov. 1971.

[42] A. J. Gibbs, "On the frequency domain responses of causal digital filters," Ph.D. thesis, Univ. of Wisconsin, Madison, 1969.

[43] ——, "The design of digital filters," *Aust. Telecommunication Res. J.*, vol. 4, pp. 29–34, 1970.

[44] B. Gold and C. M. Rader, *Digital Processing of Signals*. NewYork: McGraw-Hill, 1969.

[45] S. I. Grossman, *Multivariable Calculus, Linear Algebra and Differential Equations*. Saunders college publishing, 1995.

[46] O. Herrmann, L. R. Rabiner, and D. S. K. Chan, "Practical design rules for optimum finite impulse response lowpass digital filters," *Bell Syst. Tech. J.*, vol. 52, no. 6, pp. 769–799, July-Aug. 1973.

[47] E. Ifeachor and B. Jervis, *Digital signal processing*, 2nd ed. Prentice-Hall Inc., 2001.

[48] P. Järvilehto and K.-P. Estola, "A new modular VLSI filter architecture using computationally efficient recursive digital filter topology," in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 2, Espoo, Finland, June 1988, pp. 1301–1304.

[49] Z. Jing and A. T. Fam, "A new structure for narrow transition band, lowpass digital filter design," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 32, no. 2, pp. 362–370, Apr. 1984.

[50] A. Joworski, S. S. Lawson, and T. Wicks, "Desing of cascaded allpass structures with magnitude and delay constraints using simulated annealing and quasi-newton methods," in *Proc. IEEE Int. Symp. Circuits Syst.*, Singapore, June 1991, pp. 2439–2442.

[51] B. Joworski and T. Saramäki, "Linear phase FIR filters composed of two parallel allpass sections," in *Proc. IEEE Int. Symp. Circuits Syst.*, London, UK, May 1994, pp. 537–540.

[52] J. F. Kaiser, *System Analysis by Digital Computer*. NewYork: Wiley, 1966, f. F. Kuo and J. F. Kaiser, Edts.

[53] ——, "Nonrecursive digital filter design using I_0-sinh window function," in *IEEE Int. Symp. Circuits Syst.*, Apr. 1974, pp. 20–23.

[54] J. F. Kaiser and R. W. Hamming, "Sharpening the response of a symmetric nonrecursive filter by multiple use of the same filter," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 25, no. 5, pp. 415–422, Oct. 1977.

[55] H. Kikuchi, Y. Abe, H. Watanabe, and T. Yanagisawa, "Efficient prefiltering for FIR digital filters," *Trans. IEICE*, vol. E70, pp. 918–927, Oct. 1987.

[56] ——, "Interpolated FIR filters based on the cyclotomic polynomials," *Trans. IEICE*, vol. E70, pp. 928–937, Oct. 1987.

[57] J. J. Kormylo and V. K. Jain, "Two-pass recursive digital filter with zero phase shift," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 22, no. 5, pp. 384–387, Oct. 1974.

[58] A. Kurosu, S. Miyase, S. Tomiyama, and T. Takebe, "A technique to truncate IIR filter impulse response and its application to real-time implementation of linear-phase IIR filters," *IEEE Trans. Signal Processing*, vol. 51, no. 5, pp. 1284–1292, May 2003.

[59] S. S. Lawson and T. Wicks, "Design of efficient digital filters satisfying arbitrary loss and delay specifications," *Proc. Inst. Elect. Eng., Part G, Circuits, Devices, and Systems*, vol. 139, pp. 611–620, Oct. 1992.

[60] F. Leeb, "Lattice wave digital filters with simultaneous conditions on amplitude and phase," in *Proc. Int. Conf. Acoust., Speech, and Signal Processing*, Toronto, Canada, May 1991, pp. 1645–1648.

[61] J. K. Liang and R. J. P. de Figueiredo, "A new class of nonlinear-phase FIR digital filters and its application to efficient design of multirate digital filters," *IEEE Trans. Circuits Syst.*, vol. 32, no. 9, pp. 944–948, Sept 1985.

[62] Y. C. Lim, "Frequency-response masking approach for the synthesis of sharp linear phase digital filters," *IEEE Trans. Circuits Syst.*, vol. 33, no. 4, pp. 357–364, Apr. 1986.

[63] Y. C. Lim and Y. Lian, "Frequency-response masking approach for digital filter design: Complexity reduction via masking filter factorization," *IEEE Trans. Circuits Syst.*, vol. 41, no. 8, pp. 518–525, Aug. 1994.

[64] R. N. Madan, "Maximum entropy method and the design of linear phase FIR filters," *IEEE Trans. Circuits Syst. II*, vol. 39, no. 9, pp. 611–618, Sept. 1992.

[65] H. G. Martinez and T. W. Parks, "A class of infinite-duration impulse response digital filters for sampling rate reduction," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 27, no. 2, pp. 154–162, Apr. 1979.

[66] J. H. McClellan and T. W. Parks, "A unified approach to the design of optimum FIR linear-phase digital filters," *IEEE Trans. Circuit Theory*, vol. 20, no. 6, pp. 697–701, Nov. 1973.

[67] S. Nakamura and S. K. Mitra, "Design of FIR digital filters using tapped cascaded FIR subfilters," *Circuits, Syst. Signal Processing*, vol. 1, no. I, pp. 43–56, March 1982.

[68] Y. Neuvo, C. Y. Dong, and S. K. Mitra, "Interpolated finite impulse response filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 563–570, June 1984.

[69] Y. Neuvo, G. Rajan, and S. K. Mitra, "Design of narrow-band FIR bandpass digital filters with reduced arithmetic complexity," *IEEE Trans. Circuits Syst.*, vol. 34, no. 4, pp. 409–419, Apr. 1987.

[70] D. J. Nowak, "Linear phase digital filter design," *Proceedings of the IEEE*, vol. 57, pp. 850–851, May 1969.

[71] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*. Prentice-Hall Inc., 1989.

[72] A. V. Oppenheim, A. S. Willsky, and S. H. Nawab, *Signals and systems*, 2nd ed. Prentice-Hall Inc., 1996.

[73] T. W. Parks and J. H. McClellan, "Chebyshev approximation for nonrecursive digital filters with linear phase," *IEEE Trans. Circuit Theory*, vol. CT-19, no. 2, pp. 189–194, Mar. 1972.

[74] T. W. Parks and J. H. Mcclellan, "A program for the design of linear phase finite impulse response digital filters," *IEEE Trans. Audio Electroacoust.*, vol. 20, no. 3, pp. 195–199, Aug. 1972.

[75] R. K. Patney and S. C. D. Roy, "Design of linear-phase FIR filters using pseudo-boolean methods," *IEEE Trans. Circuits Syst.*, vol. 26, no. 4, pp. 255–260, Apr. 1979.

[76] A. D. Poularikas, *The Handbook Of Formulas and Tables for Signal Processing*. CRC Press, 1999.

[77] S. R. Powell and P. M. Chau, "A technique for realizing linear phase IIR filters," vol. SP-39, no. 11, pp. 2425–2435, Nov. 1991.

[78] L. R. Rabiner and R. E. Crochiere, "A novel implementation for narrow-band FIR digital filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-23, no. 5, pp. 457–464, Oct. 1975.

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

[80] L. R. Rabiner, J. F. Kaiser, . Herrmann, and M. I. Dolan, "Some comparisions between FIR and IIR digital filters," *Bell Syst. Tech. J.*, vol. 53, pp. 303–331, Feb. 1974.

[81] T. A. Ramstad, "Digital two-rate IIR :and hybrid IIR/FIR filters for sampling rate conversion," *IEEE Trans. Comm.*, vol. 30, pp. 1466–1476, July 1982.

[82] E. Y. Remez, "General computational methods of tchebycheff approximation," *Atomic Energy Transl. 4491*, pp. 1–85, 1957.

[83] I. Santamaría, "Design of linear-phase FIR filters using support vector regression approach," *Electronics letters*, vol. 39, no. 19, pp. 1422–1423, Sept. 2003.

[84] T. Saramäki, "Computationally efficient narrowband linear phase FIR filters," *IEEE Proc. G, Electron Circuits and Syst.*, vol. 130, pp. 20–24, Feb. 1983.

[85] ——, "Efficient recursive digital filters for sampling rate conversion," in *Proc. IEEE Int. Symp. Circuits Syst.*, Newport Beach, CA, May 1983, pp. 1322–1326.

[86] ——, "A class of linear-phase FIR filters for decimation, interpolation, and narrowband filtering," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 32, pp. 1023–1036, Oct. 1984.

[87] ——, "Design of optimal multistage IIR and FIR filters for sampling rate alteration," in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 2, San Jose, CA, May 1986, pp. 227–230.

[88] ——, "Design of FIR filters as a tapped cascaded interconnection of identical subfilters," *IEEE Trans. Circuits Syst.*, vol. 34, no. 9, pp. 1011–1029, Sept. 1987.

[89] T. Saramäki and A. T. Fam, "Properties and structures of linear-phase FIR filters based on switching and resetting of IIR filters," in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 4, New Orleans, USA, May 1990, pp. 3271–3274.

[90] T. Saramäki and S. K. Mitra, "Multiple branch FIR filters for sampling rate conversion," in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 2, San Diego, CA, May 1992, pp. 1007–1010.

[91] T. Saramäki, Y. Neuvo, and S. K. Mitra, "Design of computationally efficient interpolated FIR filters," *IEEE Trans. Circuits Syst.*, vol. 35, pp. 70–87, Jan. 1988.

[92] T. Saramäki, J. Yli-Kaakinen, and H. Johansson, "Optimization of frequency-response masking based FIR filters," *Journal of Circuits and Systems and Computers*, vol. 12, no. 5, pp. 563–590, Oct. 2003.

[93] R. W. Schafer and L. R. Rabiner, "A digital signal processing approach to interpolation," *Proc. IEEE*, vol. 61, pp. 692–702, June 1973.

[94] R. R. Shively, "On multistage FIR filters with decimation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 23, pp. 353–357, Aug. 1975.

[95] V. Sreeram and P. Agathoklis, "Design of linear phase IIR filters via impulse-response gramians," *IEEE Trans. Signal Processing*, vol. 40, no. 2, pp. 389–394, Feb. 1992.

[96] K. Surma-aho and T. Saramäki, "A systematic technique for designing approximately linear phase recursive digital filters," *IEEE Trans. Circuits Syst. II*, vol. 46, no. 7, pp. 956–963, July 1999.

[97] C. Tsai and A. T. Fam, "Efficient linear phase filters based on switching and time reversal," in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 3, New Orleans, LA, May 1990, pp. 2161–2164.

[98] P. P. Vaidyanathan, "Efficient and multiplierless design of FIR filters with very sharp cutoff via maximally flat building blocks," *IEEE Trans. Circuits Syst.*, vol. 32, no. 3, pp. 236–244, Mar. 1985.

[99] P. P. Vaidyanathan and G. Beitman, "On prefilters for digital FIR filter design," *IEEE Trans. Circuits Syst.*, vol. 32, no. 5, pp. 494–499, May 1985.

[100] P. P. Vaidyanathan and T. Q. Nguyen, "Eigenfilters: A new approach to least squares FIR filter design and applications including nyquist filters," *IEEE Trans. Circuits Syst.*, vol. 34, no. 1, pp. 11–23, Jan. 1987.

[101] P. P. Vaidyatnahan, S. K. Mitra, and Y. Neuvo, "A new approach to the realization of low-sensitivity IIR digital filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 34, no. 2, pp. 350–361, Apr. 1986.

[102] A. Wang and J. O. Smith, "On fast FIR filters implemented as tail-canceling IIR filters," *IEEE Trans. Signal Processing*, vol. 45, no. 6, pp. 1415–1427, June 1997.

[103] A. N. Willson, Jr. and H. J. Orchard, "An improvement to the Powell and Chau linear phase IIR filters," *IEEE Trans. Signal Processing*, vol. 42, no. 10, pp. 2842–2848, Oct. 1994.

Tampereen teknillinen yliopisto
PL 527
33101 Tampere

Tampere University of Technology
P.O. Box 527
FIN-33101 Tampere, Finland