TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY
Julkaisu 814 • Publication 814

Raija Lehto

Synthesis Methods for Linear-Phase FIR Filters with a Piecewise-Polynomial Impulse Response

Tampere 2009

Tampereen teknillinen yliopisto. Julkaisu 814
Tampere University of Technology. Publication 814

Raija Lehto

Synthesis Methods for Linear-Phase FIR Filters with a Piecewise-Polynomial Impulse Response

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 2nd of June 2009, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2009

ISBN 978-952-15-2167-6 (printed)
ISBN 978-952-15-4070-7 (PDF)
ISSN 1459-2045

Abstract

This thesis concentrates on synthesis methods for linear-phase finite-impulse response filters with a piecewise-polynomial impulse response. One of the objectives has been to find integer-valued coefficients to efficiently implement filters of the piecewise-polynomial impulse response approach introduced by Saramäki and Mitra. In this method, the impulse response is divided into blocks of equal length and each block is created by a polynomial of a given degree. The arithmetic complexity of these filters depends on the polynomial degree and the number of blocks. By using integer-valued coefficients it is possible to make the implementation of the subfilters, which generates the polynomials, multiplication-free. The main focus has been on finding computationally-efficient synthesis methods by using a piecewise-polynomial and a piecewise-polynomial-sinusoidal impulse responses to make it possible to implement high-speed, low-power, highly integrated digital signal processing systems. The earlier method by Chu and Burrus has been studied. The overall impulse response of the approach proposed in this thesis consists of the sum of several polynomial-form responses. The arithmetic complexity depends on the polynomial degree and the number of polynomial-form responses. The piecewise-polynomial-sinusoidal approach is a modification of the piecewise-polynomial approach. The subresponses are multiplied by a sinusoidal function and an arbitrary number of separate center coefficients is added. Thereby, the arithmetic complexity depends also on the number of complex multipliers and separately generated center coefficients. The filters proposed in this thesis are optimized by using linear programming methods.

Contents

Preface

I would first like to express my gratitude to my thesis supervisor Professor Tapio Saramäki for giving me this great opportunity, and for making it possible and unforgettable. I would also like to thank Professor Olli Vainio for his continuous support, his valuable comments and for co-authoring the articles. My thanks extend to the pre-examiners, Professor Marcello L. R. de Campos and Professor Håkan Johansson for the thorough examination and for their valuable and constructive comments on the manuscript of the thesis.

I am grateful to Dr. Juha Yli-Kaakinen for providing me information and support on practical matters to publish the articles. I sincerely like to thank Prof. Vladimir Katkovnik, Dr. Tuna B. Tarim for their valuable comments regarding my articles and Ph.Lic. Merja Laaksonen, Ph.Lic. Osmo Kaleva and Ph.D. Pertti Koivisto for all their advice.

I also want to thank all my colleagues at the Department of Signal Processing at the Tampere University of Technology, especially Dr. Sari Peltonen for her friendship and for listening to my stories during lunch. The Faculty of Computing and Electrical Engineering and the administration of Department of Signal Processing are gratefully acknowledged.

The financial support of Academy of Finland (Finnish Centre of Excellence program 2006-2011), Ulla Tuominen Foundation, Finnish Concordia Fund, and Finnish Cultural Foundation: Pirkanmaa Regional Fund are gratefully acknowledged.

I wish to thank my friends Dr. Hanspeter Schmid, Dr. Tuna B. Tarim, Dr. Felix Lustenberger and Dr. Wouter Serdjin for their unconditional support, encouragement and making ISCAS a memorable event every year. I wish especially to thank Hanspeter for all the discussions during the years between the heaven and earth and beyond. I am especially grateful for Hanspeter and Tuna for being there for me when mostly needed. I am grateful for their unconditional and continuous friendship.

I also want to acknowledge the members of my theater group, Sampolan Opistoteatteri, for enjoyable 2008-2009 with acting. Special thanks go also to my mother Ph.D. LL.M Leena Lehto for her advice, support and for teaching me English and reviewing the articles and the thesis regarding the language.

Tampere, May 2009

Raija Lehto

List of Publications

The content of this thesis serves as an introduction to the following publications. The publications are referred to as Publication-I, Publication-II and so on in the text.

I R. Lehto, T. Saramäki and O. Vainio, Formulas to Generate Efficient Piecewise-Polynomial Implementations of Narrowband Linear-Phase FIR Filters, *Proceedings of the 2006 IEEE International Symposium on Circuits and Systems*, Kos Island, Greece, pp. 2513-2516, May, 2006.

II R. Lehto, T. Saramäki and O. Vainio, Synthesis of Narrowband Linear-Phase FIR Filters with a Piecewise-Polynomial Impulse Response, *IEEE Transactions on Circuits and Systems - Part 1: Regular Papers*, vol. 54, no. 10, pp. 2262-2276, October, 2007.

III R. Lehto, T. Saramäki and O. Vainio, Synthesis of Wideband Linear-Phase FIR Filters with a Piecewise-Polynomial-Sinusoidal Impulse Response, accepted 1.12.2008 to appear in *Circuits, Systems and Signal Processing*, Birkhäuser Boston.

IV R. Lehto, T. Saramäki, O. Vainio and S. K. Mitra, Synthesis of Narrowband Differentiators with a Piecewise-Polynomial Impulse Response with Parallel-Branch Structures, submitted to *the 6th International Symposium on Image and Signal Processing and Analysis*, Salzburg, Austria, September 16-18, 2009.

The author's contribution to the publications is as follows. The author was the principal author of Publication-I – Publication-IV.

In Publication-I, the filter design methods and the implementation structures were originally proposed by T. Saramäki and S. K. Mitra in [85]. R. Lehto suggested and proposed the principal idea of how to derive the formulas for the integer-valued implementation coefficients. R. Lehto carried out the calculations with Tapio Saramäki and developed, implemented and formulated the algorithms and equations of the integer-valued coefficients.

In Publication-II, the filter synthesis scheme was originally suggested by T. Saramäki and T. G. Campbell in [8]. R. Lehto carried out all the filter designs and simulations. R. Lehto performed extensive analyses, numerical simulations and formulations.

In Publication-III, the basic idea for the filter synthesis scheme was originally suggested by T. Saramäki. R. Lehto carried out all the filter designs and simulations. Likewise, R. Lehto performed extensive analyses, numerical simulations and formulations.

In Publication-IV, the synthesis method of differentiators were based on the methods originally proposed by T. Saramäki and S. K. Mitra in [85]. R. Lehto performed the simulations. Again, R. Lehto performed analyses, numerical simulations and formulations.

Abbreviations

DSP	Digital Signal Processing
FIR	Finite-Impulse Response
IIR	Infinite-Impulse Response
MOS	Metal-Oxide Semiconductor
CMOS	Complementary Metal-Oxide Semiconductor
IC	Integrated circuit
SoC	System-on-Chip
VLSI	Very Large-Scale Integration
FPGA	Field-Programmable Gate Array
IFIR	an interpolated FIR filter
FRM	Frequency-Response Masking
PP	Piecewise-Polynomial
PPS	Piecewise-Polynomial-Sinusoidal
LP	Linear Programming
dB	decibel
NB	nota bene

Chapter 1

Introduction

1.1 Background and Motivation

Digital filters made their first appearance in the 1940s along with the first digital computers. The concept of digital data manipulation has made a dramatic impact on our society due to the increasing efficiency of digital integrated circuits (IC). Today digital filters are essential and often the most important part of digital signal processing (DSP) systems e.g. telecommunication, instrumentation, automation, consumer electronics, image processing, audio, radar, sonar, tachometry, digital TVs, multimedia, biological signal processing and bioinformatics and new emerging products such as portable devices and implantable medical devices. Furthermore, the finite-impulse response (FIR) filters constitute one of the fundamental processing elements in many DSP applications e.g. from video and image processing to telecommunications. In some applications such as video processing, FIR filters must be able to operate at high frequencies and, on the other hand, in applications like cellular telephony, FIR filters must be implemented as low-power circuits. This evolution has been possible due to advances in digital integrated circuits (IC) from metal-oxide semiconductor (MOS) circuits to complementary metal-oxide semiconductor (CMOS) circuits. The advances in CMOS technology have made it possible to integrate numerous functions into a single silicon chip, which has led to system-on-chip (SoC) solutions. Especially advances in very-large-scale integration (VLSI) circuits has made it possible to implement high-speed, low-power and highly integrated DSP systems. [33, 43, 44, 46, 63, 67, 97]

One of the main challenges throughout the years of IC circuits has been power consumption, because that is often the limiting factor when designing CMOS-based circuits. For instance, in applications like wireless communication, the bandwidth and power consumption are critical elements. For digital filters, the major power consumption is due to arithmetic operations such as multiplications and additions and/or subtractions. Multiplications consume a significant portion of the overall power [22]; power consumption is of course also increased with an increase in the sampling rate. Lowering the power consumption means developing computationally efficient filter structures as well as their design techniques. [20, 44, 46, 63, 67]

Linear digital filters are divided into two classes, namely FIR digital filters and infinite impulse response (IIR) filters. In many filtering applications, FIR digital filters are preferred over their IIR counterparts due to their many favorable properties. The main advantages, among others, are the following. An FIR filter can be designed with an exact linear phase, which means that no phase distortion is caused in the input signal during the filtering operation. Both the output noise due to multiplication round-off errors and the sensitivity to variations in filter coefficients are low. Nonrecursive realizations of FIR filters are inherently stable and free of limit cycle oscillations when implemented in a finite-word-length system. [78, 79]

However, the main drawback of conventional narrowband direct-form FIR filter designs is that they require a large number of arithmetic operations and thereby, a high power consumption and a large silicon area. The number of multipliers is the same as the filter length. In the linear-phase case the number of multipliers can be reduced approximately to half by exploiting the coefficient symmetry property. In both cases the number of adders and the number of delays is the same as the filter order. [78, 79]

The design of narrow-transition-band filters is generally regarded as a difficult problem, because they require a very large number of coefficients. This is because the order of the filter is roughly inversely proportional to the transition bandwidth [21, 24, 30, 58], which is not the case with IIR filters. This fact makes the implementation of direct-form FIR filters with a narrow transition bandwidth very costly compared to the corresponding IIR filters in term of their arithmetic complexity, i.e., the required number of adders, multipliers, and delay elements, which strongly correlates with the silicon area and power dissipation [27, 44, 46].

However, the advantages of FIR filters outweigh the difficulties in designing these filters. Computationally efficient FIR filter realizations have thus attracted the attention of several authors also in recent years [4, 5, 19, 27–29, 34–36, 40–42, 44, 51, 53, 57, 73, 75, 83, 83, 88, 94, 95, 98, 102, 103].

Many authors [1, 7, 11, 12, 16, 26, 31, 47, 48, 52, 56, 59, 61, 76, 77, 79, 81, 85–87, 92], have observed that by letting the filter length increase slightly from the minimum, there can be significant savings in the number of multipliers and adders.

Practical frequency-selective FIR filters have an impulse response with a smooth predictable envelope and they do not need the generality provided by standard FIR filter implementations. This means that there is a very strong correlation between neighboring impulse response samples. In the direct-form implementation of an FIR filter, 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. [78, 79, 85] By developing structures which take advantage of this correlation, the number of multipliers required in implementation can be drastically reduced [1, 7, 8, 11, 12, 16, 26, 31, 40–42, 44, 47, 48, 51–53, 56, 59, 61, 62, 76, 77, 79, 81, 85–88, 92]. Due to this correlation in direct-form FIR filters, if some of the samples are removed, they can be easily found with good accuracy by some interpolation scheme. There are several methods that utilize this observation to obtain computationally efficient FIR filter realizations. Thinning of the impulse response by removing some of the coefficients has been proposed by Smith and Farden [92]. The method gives some improvement, but the design of the filters is complicated and the resulting filter has non-uniform coefficient spacing and is therefore irregular in structure. Thinned FIR filters are designed by direct optimization to obtain the filter coefficients.

An efficient approach to utilize the redundancy of the impulse response samples and to overcome problems in synthesizing FIR filters is to create them with a piecewise- polynomial or a piecewise-polynomial-sinusoidal impulse response and to implement them by using recursive structures [8,11,12].

Boudreaux and Parks [7] were among the first to propose a recursive piecewise-polynomial approximation of an impulse response of FIR filters. The approach uses a low-order IIR filter in cascade with a uniformly or non-uniformly thinned numerator. Dynamic programming is used to optimize the filter coefficients of this cascade. The IIR section performs the interpolation. Even though there is a recursive IIR section, in the proposed implementation the filter has a finite- length impulse response due to pole-zero cancellation. This approach is most suitable for narrow-transition-band FIR filters.

Chu and Burrus [11, 12] proposed piecewise-polynomial filters by generalizing the filters proposed by Boudreaux and Parks. Chu and Burrus extended their approach to include wideband filters with narrow transition band by using a piecewise-polynomial-sinusoidal impulse response approximation. However, the approach proposed by Chu and Burrus suffers from drawbacks. First, the derivation of the overall filter structure is very complicated and does not arrive at the best available implementation form. Secondly, the polynomial coefficients are obtained by using nonlinear optimization with some of the coefficients fixed. In order to overcome these problems Campbell and Saramäki [8] presented a new preliminary filter structure based on and modifying the structure of Chu and Burrus for narrowband FIR filters. Saramäki and Vainio [87] proposed structures to synthesize narrowband linear-phase FIR filters with a piecewise-polynomial impulse response by using polynomials of increasing degrees and accumulators. Saramäki and Mitra [84, 85] presented a more straightforward approach to synthesize piecewise-polynomial impulse responses for narrowband FIR filters with the restriction that all the blocks where the impulse response follows a piecewise-polynomial, are of equal length. Also other approaches to reduce arithmetic complexity in narrow-transition-band applications have been developed. One of the most efficient techniques is interpolated FIR (IFIR) filters [61, 86]. This approach is based on implementing two FIR subfilters as a cascade, where the first section generates the sparse impulse response with every Lth sample to be nonzero and the other section "fills in" the missing samples. Another approach, which is also based on using periodic subfilters to generate the overall impulse response, is the frequency-response masking (FRM) technique originally developed by Lim [52]. This technique is also suitable for applications which require a narrow transition band.

Furthermore, approaches to reduce the arithmetic complexity by altering the sampling rate in FIR subfilters have been developed. One such a method is multirate and complementary filtering. The internal data rate is altered by using decimation and interpolation. The redundancy of the impulse response samples is reduced by making the actual frequency shaping at a low rate and with a somewhat wide passband. This approach is efficient in both narrowband and wideband applications. The major drawback is that, due to the s-ampling rate alteration in subfilters, there is a danger of aliasing, which has to be taken into account in the design process. [18, 70]

The cost of the reduction of arithmetic complexity in all the above-mentioned approaches is a somewhat increased filter order compared to the equivalent optimum direct-form linear-phase FIR filter. Methods to estimate the required FIR filter order have been developed by several authors [21, 24, 30, 58].

Additionally, efficient synthesis methods for linear-phase FIR filters have been developed by using a switching and resetting principle of IIR filters. The filters are constructed by using a causal and anti-causal version of an IIR filter so that their impulse response is a truncated and shifted version of the overall impulse response of the resulting filter. The resulting filter is implemented by using two identical copies of the same IIR filter by switching and resetting. The switching and resetting principle works so that e.g. half of the input samples are fed into the first filter and the second half of the input samples to the other filter. The overall structure is reset before the new set of samples is fed to the structure to stabilize the inexact pole-zero cancellation due to finite word length used in the implementation. [4, 5, 81]

These efficient methods result in filters with reduced arithmetic operations, such as fewer multipliers and adders in applications regarding low power consumption and less silicon area. But the number of storage elements remains the same as the filter order. [27, 44, 46]

1.2 Scope of the Thesis

This thesis presents methods to synthesize linear-phase FIR filters for computationally efficient realizations especially for applications requiring a narrow transition band. The main focus is as follows. First, formulas for integer-valued coefficients to generate efficient implementations are proposed in Publication-I for linear-phase FIR filters with a piecewise-polynomial impulse response developed by Saramäki and Mitra. Second, computationally efficient methods for the design and implementation of linear-phase FIR filters with a piecewise-polynomial and a piecewise-polynomial-sinusoidal impulse response are proposed in Publication-II and Publication-III. The arithmetic complexity of the proposed filters is based on the number of subresponses and the polynomial order and, in the sinusoidal case, also on the number of complex multipliers in case of Type-1 linear-phase FIR filter. Third, a computationally efficient synthesis method based on the approach proposed by Saramäki and Mitra in [85] is proposed for differentiators in Publication-IV.

1.3 Outline of the Thesis

The thesis is organized as follows. Chapter 2 gives briefly the background information of design and implementation considerations of linear-phase FIR filters. Chapter 3 reviews some of the most important previous approaches closely related to Publication-I – Publication-IV to synthesize linear-phase FIR filters efficiently for narrow-transition-band applications. The methods described are multirate and complementary filtering, interpolated FIR (IFIR) filters, the frequency-response masking approach (FRM), and methods based on switching and resetting IIR filters. Chapters 4 summarizes the original contributions of the thesis. Chapter 5 concludes with results and discussion. A list of symbols is not given because the same variables are used in different contexts in the publications and references. The aim has, where possible, been to use the symbols in the same meanings throughout the thesis.

1.4 Definitions

This section presents some essential definitions used in the thesis. In this thesis $\lfloor\cdot\rfloor$ means rounding downwards to the nearest integer.

Definition 1.4.1. Suppose N is the order of the filter. The transfer function of an FIR filter is given by [55, 78]

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

and its frequency response by [55, 78]

$$H(e^{j\omega}) = \sum_{n=0}^{N} h(n) e^{-jn\omega},$$

where $h(n)$ is the impulse response of the filter.

Definition 1.4.2. Suppose N is the order of the filter. The four types of linear-phase FIR filters are given as [55, 78]
Type 1: N is even and $h(N-n) = h(n)$ for all n.
Type 2: N is odd and $h(N-n) = h(n)$ for all n.
Type 3: N is even and $h(N-n) = -h(n)$ for all n.
Type 4: N is odd and $h(N-n) = -h(n)$ for all n.
Additionally, Type 3 attains also the value zero at the center of symmetry.

Definition 1.4.3. Suppose N is the order of the filter. The zero-phase frequency responses of linear-phase FIR filters are of the form [55, 78]:

$$H(\omega) = \begin{cases} h(N/2) + \sum_{n=1}^{N/2} h(N/2-n) 2\cos(n\omega), & \text{Type 1} \\ \sum_{n=0}^{(N-1)/2} h((N-1)/2-n) 2\cos((n+1/2)\omega), & \text{Type 2} \\ \sum_{n=0}^{N/2-1} h(N/2-1-n) 2\sin((n+1)\omega), & \text{Type 3} \\ \sum_{n=0}^{(N-1)/2} h((N-1)/2-n) 2\sin((n+1/2)\omega), & \text{Type 4.} \end{cases} \tag{1.1}$$

Definition 1.4.4. Two linear-phase Type 1 FIR filters are said to be complementary if $H_a(\omega) + H_c(\omega) = 1$.

Fig. 1.1 shows the scheme of a complementary filter pair.

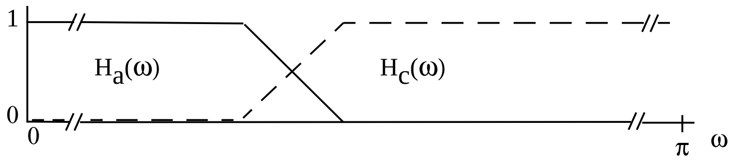


Figure 1.1: Complementary filter pair. [18]

Definition 1.4.5. Suppose N is the order of the linear-phase FIR filter, and $N \in \mathbb{N}$. Let $M \in \mathbb{Z}_+$. Let $N_j \in \mathbb{N}$ for $j = 1, 2, \ldots, M$ and $N_{M+1} \in \mathbb{R}_+$ be such that $N_1 = 0$, $N_j < N_{j+1}$ when $j = 1, 2, \ldots, M$ and $N_{M+1} = N/2$. Then, the block b_j is defined to be the interval (see Fig. 1.2)

$$b_j = \begin{cases} [N_j, N_{j+1}[, & \text{if } j = 1, 2, \ldots, M-1, \\ [N_M, N_{M+1}], & \text{if } j = M, \\ ]N - N_{2M-j+2}, N - N_{2M-j+1}], & \text{if } j = M+1, M+2, \ldots, 2M. \end{cases}$$

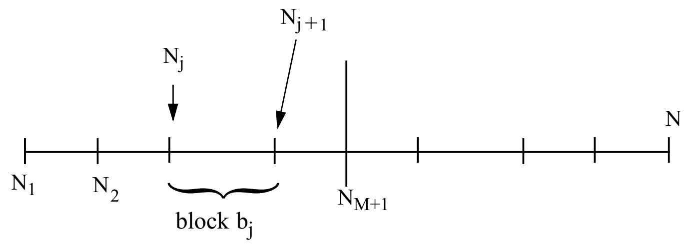


Figure 1.2: A graphical presentation of the definition of a block.

Definition 1.4.6. Let $n \in \mathbb{N}$. If p_j^L, for $j = 1, 2, \ldots, M$, is a jth polynomial of a degree L, then a piecewise polynomial is

$$\begin{cases} p_1^L(n-N_1), & n \in b_1 \\ p_1^L(n-N_1) + p_2^L(n-N_2), & n \in b_2 \\ p_1^L(n-N_1) + p_2^L(n-N_2) + p_3^L(n-N_3), & n \in b_3 \\ \vdots \quad \vdots \quad \vdots & \vdots \\ \sum_{j=1}^{M-1} p_j^L(n-N_j) & n \in b_{M-1} \\ \sum_{j=1}^{M} p_j^L(n-N_j) & n \in b_M, \end{cases}$$

which applies up to the center of symmetry for linear-phase FIR filters. After the center of symmetry it is either symmetrical or anti-symmetrical depending on the linear-phase type.

Definition 1.4.7. Suppose N is the order of the linear-phase FIR filter type, and $N \in \mathbb{N}$. Let $N_j \in \mathbb{N}$, $j = 1, 2, \ldots, M$. A slice s_j is an interval $s_j = [N_j, N - N_j]$ for $j = 2, 3, \ldots, M$. If $j = 1$, a slice s_1 is an interval $s_1 = [N_1, N]$ (see Fig. 1.3).

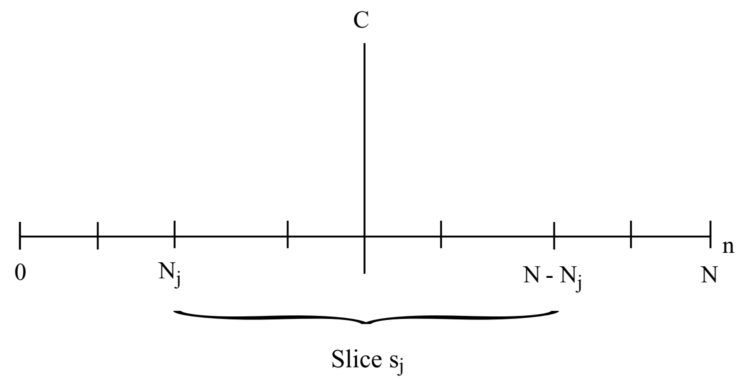


Figure 1.3: A graphical presentation of the definition of a slice, where C marks the center of symmetry.

NB: In this work, a polynomial p_j^L in Definition 1.4.6 is on the slice s_j. A slice s_j is also a union of blocks in a closed interval, i.e., $s_j = \bigcup_{k=j}^{2M-j+1} b_k$.

Definition 1.4.4 is used in Chapter 3 and Definitions 1.4.6 and 1.4.7 are used in Publication-II and Publication-III. Publication-II gives also a definition of a slice with a slightly different formulation, i.e., Eqs. (16a)–(16c) in Publication-II.

Chapter 2

Background of FIR Filter Synthesis

This chapter gives a brief background of digital filter design process and implementation issues related to the FIR filters proposed in Publication-I – Publication-IV according to literature [2, 18, 55, 78–80, 100]. Digital filters are the most important part in DSP systems and they are defined either by the difference equation, by the transfer function or the frequency response or by the impulse response of the system or by pole and zero locations and a gain factor. Each of these descriptions completely characterizes the digital filter. The difference equation gives an input-output relation of the filter in the time domain. The frequency response shows the frequency properties, and it is the transfer function evaluated on the unit circle. Digital filters are designed to modify an input signal in the frequency domain to meet the given specifications. The purpose is to extract the properties of the desired signal from other types of signals such as noise, interference or some other unwanted signals. [2, 18, 55, 78]

2.1 Filter Design Process

This section gives a brief overview of the basic steps to design digital filters. These steps are described in relation to the publications in this thesis. The digital filter design process usually involves the following three steps:

1. Approximation is a process to design a digital filter to meet the given specifications such as magnitude, phase and, sometimes, even time-domain conditions. Certain error measures are used to characterize the "goodness" of the filter. Approximation problems are usually solved by using some optimization methods to obtain filter coefficient

values.

2. Implementation (realization) is a process to obtain the filter structure as a block-diagram or signal-flow-diagram from the transfer function of the filter. The filter structure is the "high-level" description of how to implement the filter. It is derived from the design method at hand. All of the transfer functions can be implemented by using different kinds of structures.
3. Practical implementation can be performed in various ways based on the application at hand and on the filter structure. The filter structures can be implemented in various ways e.g. by using software or hardware. A software-based implementation is usually a program running on a computer or on a signal processor. In a hardware-based implementation, the filter structure is transformed into a dedicated circuitry.

2.2 Filter Design Methods

The filter design methods are basically divided into two categories, optimization-based and non-optimization-based methods.

2.2.1 Optimization-Based Methods

When designing linear-phase FIR filters by optimization, the required filter order to meet the specified design criteria cannot be analytically determined, only estimated. There are several estimation methods developed during the years [21, 23–25, 30, 58].

The Remez Algorithm

One efficient algorithm to design optimum-magnitude-response FIR filters with arbitrary specifications is the Remez multiple-exchange algorithm. Originally, the algorithm was implemented by Parks and McClellan and later on improved by McClellan, Parks and Rabiner [54]. This program is directly applicable to obtain optimal designs for most FIR filters. The algorithm is based on minimizing the L_∞ norm in an iterative way. It can be used directly to design filters with several passbands and stopbands. The main drawback is that it cannot be used if filters have several constraints specified in the time and/or frequency domains. In that case linear programming is the most suitable method to obtain the filter coefficients. Filters designed by the Remez algorithm are realized by using direct-form structures as such or by using direct-form structures as building blocks.

Linear Programming

One of the most flexible method to design FIR filters is linear programming. It can be used when constraints in the time and/or frequency domains are specified. Linear-programming algorithms solve a minimax problem, i.e., minimize the L_∞ norm. Publication-I – Publication-IV in this thesis use linear programming to obtain the filter coefficients.

2.2.2 Non-Optimization-Based Methods: The Windowing Method

The most straightforward method to design FIR filters is to determine the infinite-duration impulse response by expanding the frequency response of an ideal filter in a Fourier series and by truncating it and smoothing it with a window function. The main advantage of the window method is that the filter coefficients can be obtained very fast and easily in the closed form. The main drawback is that the passband and stopband ripples are restricted to be the same. Additionally, this method is not optimal, i.e, the required filter order is not the lowest one to satisfy the given specifications.

2.3 FIR Filter Implementation

When the required specifications are formulated and the suitable transfer functions are obtained, a realization phase takes place. Some transfer functions can be realized using many different structures. These structures differ from each other with respect to complexity, coefficient sensitivity, and influence of round-off errors in arithmetic operations. The most basic structures are derived directly from the basic convolution sum. Usually, these structures require as many multiplications as the filter length; as many additions as the filter order. Therefore, these structures are not computationally efficient realizations. In case of linear-phase FIR filters the number of multiplications is approximately half of the filter length. Especially, when narrow-transition-band filters are considered, the direct-form structures require too many coefficients to be implemented and too many additions and thus the filter requires more power and more silicon area. In Publication-I – Publication-III computationally efficient structures are proposed.

2.4 Practical FIR Filter Implementation

2.4.1 Coefficient Quantization

For practical implementations, when implementing in a finite word-length system, the filter coefficients have to be quantized to a fixed number of bits. This means that the frequency response deviates from the one which would have been obtained with infinite word length. Quantization can be performed e.g. by using rounding or truncation. The implementation cost, arithmetic complexity and the speed depend on the filter coefficient word length; therefore, quantization is usually performed in such a way that the word length is the minimum one still satisfying the given filter specification due to cost, complexity and the speed of the implementation.

2.4.2 Coefficient Representation

Before the quantization of the filter coefficients, the number representation has to be decided. The main numerical representations are fixed-point, floating-point and a combination of these two called a block-floating-point representation. The most common fixed-point representations are the sign and magnitude, one's complement and two's complement. The choice of the representation is usually determined by hardware or programming considerations. Two's complement arithmetic has some favorable properties. When several numbers are added, the result is correct even if there are overflows in the intermediate additions if the final result is in the desired range. When numbers are added, the sign bit is treated like other digits, and thereby the operation of subtraction may be performed by using appropriate hardware components like an adder and a two's complementer. In Publication-I – Publication-IV two's complement arithmetic is recommended for practical implementation due to its above-mentioned properties.

2.4.3 Finite Word-Length Effects

When implementing digital filters in digital hardware with finite word length to represent the filter parameters, the finite word-length representation of the filter coefficients causes errors in magnitude and phase of the filtered signal and also in internal calculations. In general FIR filters are scaled so that the filter input is divided by the scaling constant and the output is multiplied with the same value. Additionally, if filter coefficients can be represented as integer-valued numbers, there is no need for multiplications in implementation, only additions are needed. In this thesis the filter structures are developed to be used in combination with scaling and modulo arithmetic, e.g. two's complement arithmetic for efficient implementation.

The following main errors are caused by the use of finite word length in filters.

1. Analog-digital conversion generates noise when representing samples of an input data with few bits.
2. Coefficient quantization errors are caused by representing the filter coefficients with a finite number of bits. Quantization of the coefficients results in an error in the magnitude and phase responses.
3. Various kinds of oscillations such as granular oscillations and overflow oscillations may be present in filters. These limit cycles are caused by overflows and rounding or truncation in recursive structures [100]. Overflow oscillations occur due to arithmetic operations inside the filter when the signal value is out of the dynamic range of a number representation used in the filter.
4. Output noise due to multiplication round-off errors occurs when rounding or truncating the multiplication products within the filter.

2.4.4 Scaling Methods

Filter scaling is used mainly because of two reasons. First, to avoid overflows. Secondly, to reduce output noise due to multiplication round-off errors. In order to accomplish that, the signal levels inside the filter, should be kept as high as possible. Therefore, these two requirements are "mutually disjoint". There exist, however, several scaling methods which make a compromise between the probability of overflows and the values of output noise. The purpose of scaling is to ensure that, at the node variable, $w(n)$, which needs to be scaled, satisfies

$$|w(n)| \leq 1 \text{ for all values of } n. \tag{2.1}$$

Also, we assume that input signal of the filter is bounded by unity, i.e.,

$$|x(n)| \leq 1 \text{ for all values of } n. \tag{2.2}$$

Next, the most common scaling methods are briefly reviewed.

Worst-Case Scaling

Worst-case scaling is defined generally, in case of FIR filters, as follows. $w(n)$ can be expressed as a linear convolution of $h(n)$ and the input $x(n)$, i.e.,

$$w(n) = \sum_{k=0}^{N} h(k)x(n-k).$$

From which it follows that

$$|w(n)| = |\sum_{k=0}^{N} h(k)x(n-k)| \leq \sum_{k=0}^{N} |h(k)|.$$

Thus, Eg. (2.1) is satisfied if $\sum_{k=0}^{N} |h(k)| \leq 1$, where $h(k)$ is the impulse response up to the node variable, $w(n)$, N is the overall filter order. Thus, the condition (2.1) is satisfied. The above condition is both necessary and sufficient to quarantee no overflow [55].

In Publication-I – Publication-IV worst-case scaling is recommended in practical implementations.

The L_∞ norm, denoted by $\| L \|_\infty$

First, the Fourier transform of (2.1) is

$$W(e^{j\omega}) = H(e^{j\omega})X(e^{j\omega}).$$

The inverse Fourier transform is

$$|w(n)| \leq \frac{1}{2\pi}\int_{-\pi}^{\pi} |H(e^{j\omega})|X(e^{j\omega})|d\omega \quad (2.3)$$

$$\leq \| H(e^{j\omega}) \|_\infty \frac{1}{2\pi}\int_{-\pi}^{\pi} |X(e^{j\omega})|d\omega \quad (2.4)$$

$$\leq \| H \|_\infty \| X \|_1 \ . \quad (2.5)$$

$$(2.6)$$

If $\| X \|_1 \leq 1$, then (2.1) is satisfied if

$$\| H(e^{j\omega}) \|_\infty \leq 1. \quad (2.7)$$

If the mean absolute values of the input spectrum are bounded by unity, there will be no adder overflow if the peak gains from the filter input to all adder outputs are scaled satisfying (2.7) [55]. The L_∞ norm is also called the peak-scaling norm.

The L_2 norm, denoted by $\| L \|_2$

Applying the Schwartz inequality in (2.3) we get

$$|w(n)|^2 \leq \left(\frac{1}{2\pi}\int_{-\pi}^{\pi} |H(e^{j\omega})|^2 d\omega\right)\left(\frac{1}{2\pi}\int_{-\pi}^{\pi} |X(e^{j\omega})|^2 d\omega\right.$$

If the input to the filter has a finite energy bounded by unity, then the adder overflow will be prevented by scaling the filter satisfying $\| H \|_2 \leq 1$ [55]. This norm is most suitable when filtering random input signals.

Chapter 3

Design and Implementation Methods for Linear-Phase FIR Filters

This chapter reviews basic ideas behind some of the most efficient and common methods used to design linear-phase FIR filters. In the literature, several methods have been proposed to reduce the arithmetic complexity of sharp FIR filters [1, 3, 6, 7, 11, 12, 14, 16, 20, 26, 27, 29, 35, 36, 44, 46–48, 51, 59, 68–70, 72, 77, 79, 81, 90, 99, 101, 102].

In this chapter, efficient FIR filter synthesis approaches are reviewed belonging to three common FIR filter classes, such as filters based on multirate filtering, the use of periodic subfilters and recursive implementation structures. The first section very briefly gives some basic ideas of FIR filters using multirate and complementary filtering [18, 70, 71]. The two following sections describe filter approaches such as the interpolated FIR filters (IFIR) and the frequency-response masking (FRM) approach. These two filters use periodic subfilters as building blocks. The figures of the filter responses in this chapter show zero-phase frequency responses and magnitude responses. The fourth section gives the idea of FIR filters based on switching and resetting IIR filters. These filters are implemented by using recursive structures.

3.1 FIR filters Using Multirate and Complementary Filtering

This section gives a very brief description of multirate and complementary filtering technique.

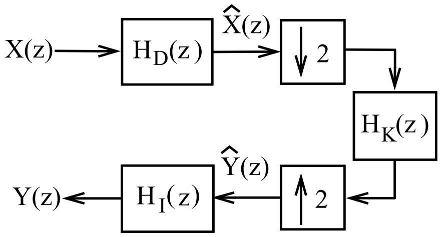


Figure 3.1: A single-stage multirate implementation structure [18].

The first step of multirate filtering [18, 70] is that the bandwidth of a signal is first reduced by using a narrowband FIR filter with a moderate transition bandwidth. The second step is to reduce the sampling rate. The output signal is further filtered at the reduced sampling rate and, finally, the sampling rate is restored by interpolation. The inner part of the filter stage in Fig. 3.1 consists of the downsampler, the kernel filter transfer function $H_K(z)$ and the upsampler. Additionally, at the input of the filter stage there is a decimator transfer function $H_D(z)$ and at the filter output there is an interpolator with the transfer function $H_I(z)$. The overall system is periodically time invariant. The multirate complementary technique uses a basic multirate filter stage shown in Fig. 3.2 and a complement formation as a building block resulting in the structure in Fig. 3.3.

Wideband filters with a cutoff frequency in the range $0 \leq \omega \leq \pi$ and with a steep slope in the transition band can be realised if the basic multirate filter stage is complemented, i.e., from the input, a delay line, z^{-N}, Fig. 3.2, is added to the output. The sign of the output of the interpolator is complemented. This delay between input and output compensates the propagation delay through the decimation, $H_D(z)$, kernel, $H_K(z)$, and interpolator, $H_I(z)$, filter transfer functions, which are assumed to be linear

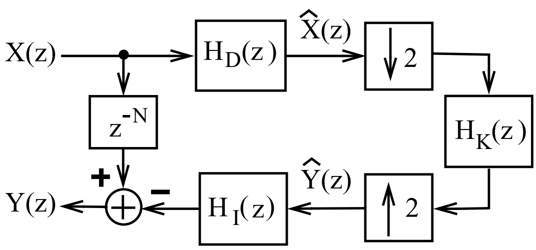


Figure 3.2: A single-stage multirate implementation structure with a complement formation [18].

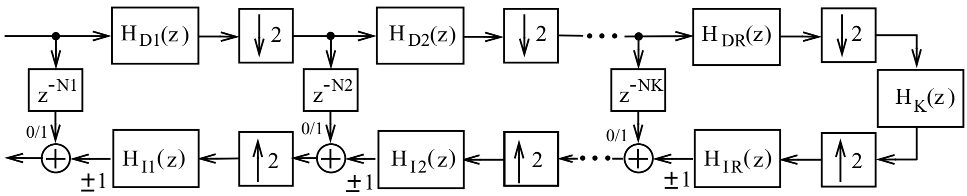


Figure 3.3: A general structure of multirate implementation with complement formations [18].

phase. By cascading multirate filter stages with a complement formation, as shown in Fig. 3.3, it is possible to realize filters with arbitrarily narrow-transition bands. At each filter stage, the following decisions have to be made. First, whether a complement should be taken (weightings +1 or −1; see Fig. 3.3) or not (weightings 0 or 1; see Fig. 3.3). Second, should the decimation and interpolator filters be implemented as a lowpass or a highpass filter. The first stage determines if the overall filter is a lowpass or a highpass filter. E.g. for a lowpass filter, the kernel filter is to be realized as a lowpass one and, if the cutoff frequency is above $\pi/2$, it is necessary to take the complement and to implement $H_D(z)$ and $H_I(z)$ as highpass filters. If the cutoff frequency is below $\pi/2$, no complements should be taken and $H_D(z)$ and $H_I(z)$ should be implemented as lowpass filters. The structure can be viewed as a nesting of multirate filter stages. [18] The advantage of multirate complementary filters is that arbitrarily narrow-transition-band filters can be implemented. [18, 70, 79]

3.2 Interpolated FIR Filters

This section reviews the basic idea behind interpolated FIR (IFIR) filters [52,56,61,79,86,88]. This method considers lowpass filters when the stopband edge, ω_s, is less than $\pi/2$ as a narrowband filter. This section also describes the basic idea of IFIR filters according to [18] and the above-mentioned publications. The main idea is to implement the filter as a cascade of two FIR filter sections as

$$H(z) = F(z^L)G(z), \tag{3.1}$$

where the first subfilter with the transfer function, $F(z^L)$, generates a sparse impulse response with every Lth impulse response value as non-zero, which makes this subfilter periodic. The second subfilter, the non-periodic $G(z)$, performs the interpolation. Both subfilters are processed at the same sampling rate which is the same as the rate at the input and output of the filter.

Hereby, IFIR has a constant internal data rate and has never internal aliasing problems and it belongs to a class of FIR filters using periodic subfilters. An example with periodic and non-periodic responses of an IFIR design is shown in Fig. 3.4. The zero-phase frequency response of this filter is given by

$$H(\omega) = F(L\omega)G(\omega), \tag{3.2}$$

where the periodical response $F(L\omega)$ is obtained by adding zero-valued impulse response samples between every non-zero-valued sample of the prototype filter response, $F(\omega)$, shown in Figs. 3.4(a) and 3.5(a). This compresses the response by a factor of L and makes it periodical as shown in Figs. 3.4(b) and 3.5(b). This results in an L-times narrower transition band than the transition band of the prototype $F(\omega)$ is. Therefore, the resulting response is a frequency-axis-compressed version of the prototype filter so that the interval $[0, L\pi]$ is shrunk onto $[0, \pi]$, i.e., the passband and stopband edges of $F(L\omega)$ are given as $\omega_p = \theta/L$ and $\omega_s = \phi/L$ (see Fig. 3.5). The periodicity of the $F(L\omega)$ is $2\pi/L$ and it has unwanted passband regions and does not provide attenuations in the following regions.

$$\Omega_s = \bigcup_{k=1}^{\lfloor L/2 \rfloor} [k\frac{2\pi}{L} - \omega_s, \min(k\frac{2\pi}{L} + \omega_s, \pi)]. \tag{3.3}$$

The non-periodic $G(z)$ is used to eliminate these unwanted replicas of the passband given by (3.3). The non-periodic response $G(\omega)$ shown in Fig. 3.5(c), does not need to have a very steep transitionband if the value of L is properly chosen. Therefore, it has a lower arithmetic complexity than the periodic filter. When the periodic filter with every Lth value as non-zero, is cascaded with the non-periodic filter, the non-periodic one "fills in" the missing samples of the periodic one. The periodic $F(L\omega)$ is also referred to as a shaping filter due to its shaping characteristics of the desired baseband. It is important to note that the passband and the transition band of the overall filter are $1/L$th of the corresponding widths of the prototype filter. The periodic $F(z^L)$ in (3.1), increases the order of $H(z)$ given in (3.1) to L times the order of $F(z)$. Since, in the periodic filter, only every Lth value is non-zero, the arithmetic complexity, i.e., the number of adders and multipliers, remains the same. The use of $G(z)$ increases the arithmetic complexity of the overall filter to be slightly more than $1/L$th of that of the direct-form design. If the factor L is properly chosen, it adds very little to the filter complexity. For the periodic, $F(z^L)$ in (3.1), the implementation structure is obtained by substituting the single delays, z^{-1}, with multiple delays, z^{-L}, as shown in Fig. 3.6. This structure cannot be used with a sampling rate reduced by a factor L, instead L different input samples have to be stored, one

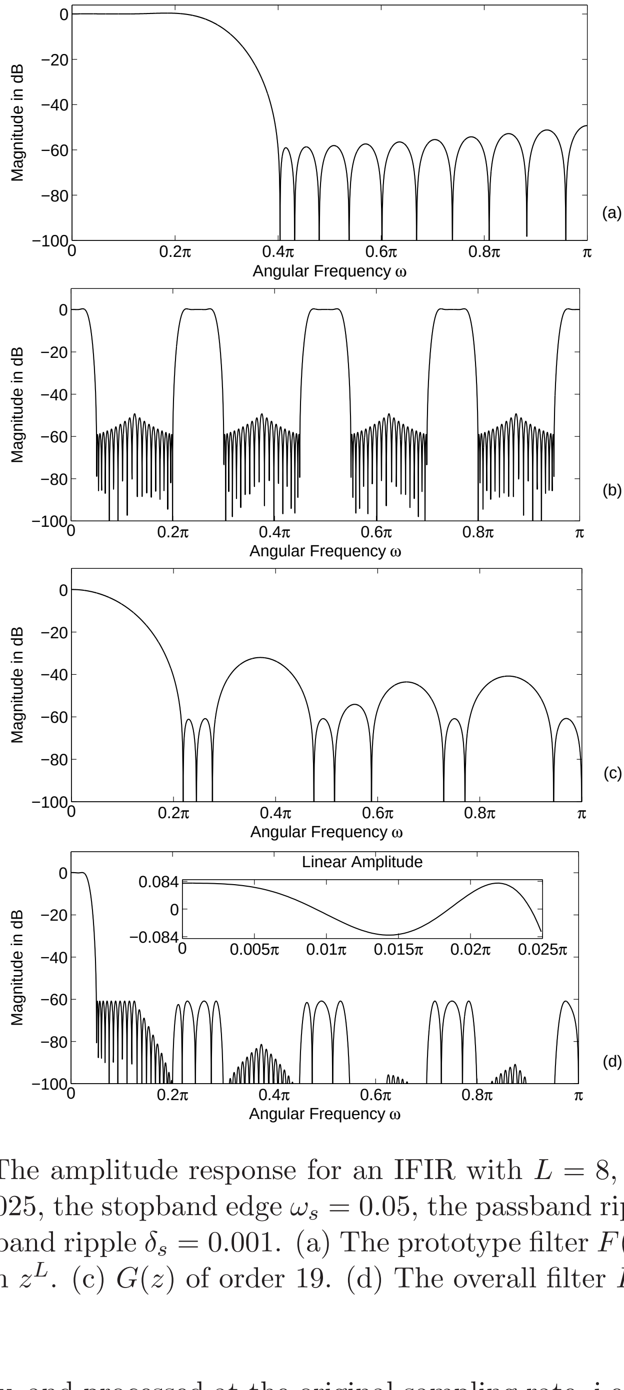


Figure 3.4: The amplitude response for an IFIR with $L = 8$, the passband edge $\omega_p = 0.025$, the stopband edge $\omega_s = 0.05$, the passband ripple $\delta_p = 0.01$ and the stopband ripple $\delta_s = 0.001$. (a) The prototype filter $F(z)$, (b) $F(z^L)$ of order 26 in z^L. (c) $G(z)$ of order 19. (d) The overall filter $H(z)$.

for each delay, and processed at the original sampling rate, i.e., all these input samples contribute to the final output. This means that the arithmetic complexity of $H(z)$ is reduced by a factor of L compared to the conven-

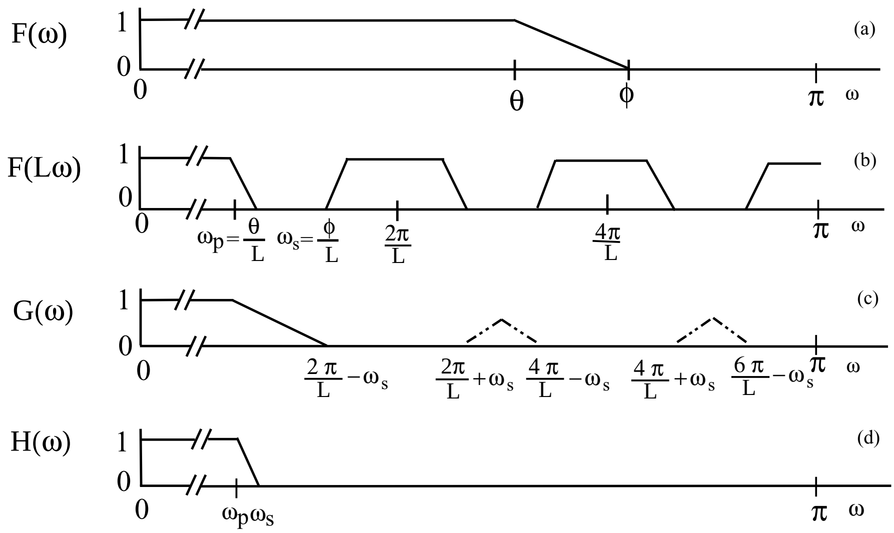


Figure 3.5: Synthesis of a narrowband filter as a cascade of a periodic and a non-periodic zero-phase frequency response. a) A prototype, $F(\omega)$, b) a periodic response $F(L\omega)$, c) a non-periodic response $G(\omega)$. d) The obtained response, $H(\omega)$ [77].

tional direct-form design, i.e., totally approximately to half of that which is required in direct-form FIR filter synthesis. Thereby, the benefit of this technique is that the number of coefficients is reduced. By exploiting the coefficient symmetry for linear-phase FIR filters, the number of multipliers in the implementation can be further reduced. The example shown in Fig. 3.4 requires 24 multipliers, 45 adders when the coefficient symmetry is exploited. The filter order is increased by approximately 5 per cent compared to the equivalent direct-form filter. The reduction of the multipliers also results in reduced coefficient sensitivity and round-off noise levels.

In the cases where the factor L can be factorized and expressed as a product of the form

$$L = \prod_{r=1}^{R} L_r, \tag{3.4}$$

where L_rs are integers, further savings in filter complexity can be achieved if the $G(z)$ is designed in multistage form as follows:

$$G(z) = G_1(z)G_2(z^{L_1})G_3(z^{L_1L_2})\cdots G_R(z^{L_1L_2\cdots L_R}). \tag{3.5}$$

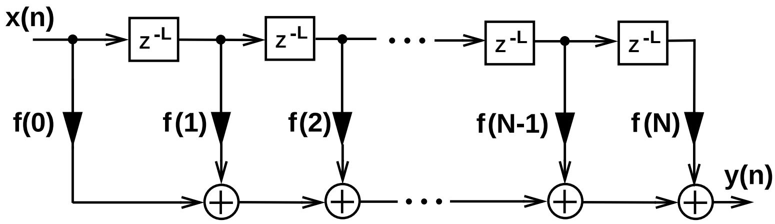


Figure 3.6: A direct-form implementation structure for $F(z^L)$ [18].

The design of IFIR can also be done iteratively in order to further reduce the arithmetic complexity. Also the optimum value of L can be found, at which the required number of coefficients is the least, which makes it [86] suitable in very narrow passband cases, where the interpolator factor L becomes very high [78, 79]. Fig. 3.4 shows the design of a narrow passband filter. For most very narrowband cases, filters synthesized in the simplified form $H(z) = F(z^L)G(z)$, give the best result. The estimate of the order of the periodic $F(z^L)$ is given by [80]

$$N_F = \frac{N}{L} \tag{3.6}$$

and the estimate of the order of $G(z)$ is given by [80]

$$\begin{aligned} N_g = \cosh^{-1}(\frac{1}{\delta_s})\Big[& \frac{1}{\cosh^{-1} X\left(\frac{L\omega_p}{2}, \pi - \frac{L(\omega_p + 2\omega_s)}{3}\right)} \\ & + \frac{L/2}{\cosh^{-1} X\left(\frac{L\omega_p}{2}, \pi - \frac{L(\omega_p + 2\omega_s)}{6}\right)}\Big], \end{aligned} \tag{3.7}$$

where

$$X(\omega_p, \omega_s) = \frac{2\cos(\omega_p) - \cos(\omega_s) + 1}{1 + \cos(\omega_s)}. \tag{3.8}$$

The design results of narrowband filters with a piecewise-polynomial impulse response in Publication-II are compared to IFIR filters to show the effectiveness of the proposed filters in terms of their arithmetic complexity.

3.3 Frequency-Response Masking Approach

The frequency-response masking approach (FRM) is eminently suitable to implement narrow transition-band FIR filters [20, 27, 29, 44, 45, 48, 60, 72, 79, 88, 101, 103] and to implement linear-phase filter banks for applications such as tone control and digital audio systems [49]. Originally, the method was proposed by Lim [47] and further developed by other authors [20, 27, 29, 44, 45, 48, 49, 60, 72, 83, 88, 101, 103]. This section briefly gives the basic idea of the FRM method for arbitrary bandwidth design discussed in the above-mentioned publications.

3.3.1 One-Stage Approach

The approach is based on a pair of filter transfer functions, the transfer function of a prototype filter,

$$F(z) = \sum_{n=0}^{N_F} f(n) z^{-n}, \tag{3.9}$$

and its complementary transfer function, $z^{-N_F/2} - F(z)$, and on two non-periodical masking filters with the transfer functions, $G_1(z)$ and $G_2(z)$ as follows:

$$G_1(z) = z^{-M_1} \sum_{n=0}^{N_1} g_1(n) z^{-n}, g_1(N_1 - n) = g_1(n), n = 0, 1, \ldots, N_1 \tag{3.10}$$

and

$$G_2(z) = z^{-M_2} \sum_{n=0}^{N_2} g_2(n) z^{-n}, g_2(N_2 - n) = g_2(n), n = 0, 1, \ldots, N_2. \tag{3.11}$$

N_F is the order of $F(z)$ and $N_1 + 2M_1$ and $N_2 + 2M_2$ are the orders of $G_1(z)$ and $G_2(z)$ in the above equations, respectively. The filter transfer function pair $F(z)$ and $z^{-N_F/2} - F(z)$ forms a complementary pair because their zero-phase frequency responses, $F(\omega)$ and $1 - F(\omega)$, add up to unity according to the Definition 1.4.4 in Section 1.4 and as seen in Fig. 3.7(a). The first step is to create a periodical filter pair of the prototype filter pair by substituting each delay term, z^{-1}, in (3.9) by L multiple delays resulting in z^{-L} delays and giving the following periodical transfer function:

$$F(z^L) = \sum_{n=0}^{N_F} f(n) z^{-nL}, f(N_F - n) = f(n), n = 0, 1, \ldots, N_F \tag{3.12}$$

and its complementary pair: $z^{-LN_F/2} - F(z^L)$. The substitution of z^{-1} with z^{-L} preserves the complementary property and results in the periodic

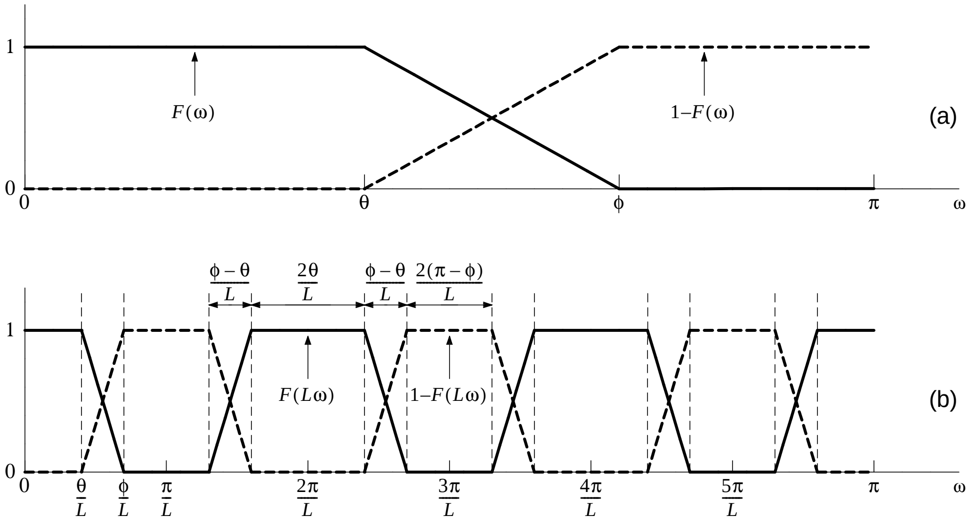


Figure 3.7: Generation of a periodic complementary filter pair. a) Prototype filter responses, $F(\omega)$ and $1 - F(\omega)$. b) Periodic filter responses, $F(L\omega)$ and $1 - F(L\omega)$ [77].

responses, $F(L\omega)$ and $1 - F(L\omega)$, shown in Fig. 3.7, with a filter order increased to LN_F. The periodic filters are frequency-axis-compressed versions of the prototype responses so that the interval of $[0, \pi]$ is compressed into the interval $[0, \pi/L]$. Because the periodicity of the prototype responses is 2π, the periodicity of the resulting responses is $2\pi/L$. Thereby, the periodic filters contain several passbands and stopbands in the baseband interval $[0, \pi]$ as seen in Fig. 3.7(b).

Because of the passband and transition-band replicas on the stopband of the periodic prototype filters, two masking filters with transfer functions, $G_1(z)$ and $G_2(z)$ given by (3.10), and (3.11), are needed to attenuate the unwanted replicas of the passbands and the transition bands and to form the passband of the desired filter as follows. The first masking filter masks the response of the periodic $F(z^L)$ and the second masking filter masks the response of $z^{-LN_F/2} - F(z^L)$ [47]. The overall filter is obtained by summing these two resulting filters as follows:

$$H(z) = H_1(z) + H_2(z) = F(z^L)G_1(z) + (z^{-LN_F/2} - F(z^L))G_2(z). \quad (3.13)$$

Next, the selections, which assure that the obtained transfer function $H(z)$ has a linear phase, are explained. The order N_F is even and the orders N_1 and N_2 can be either even or odd.

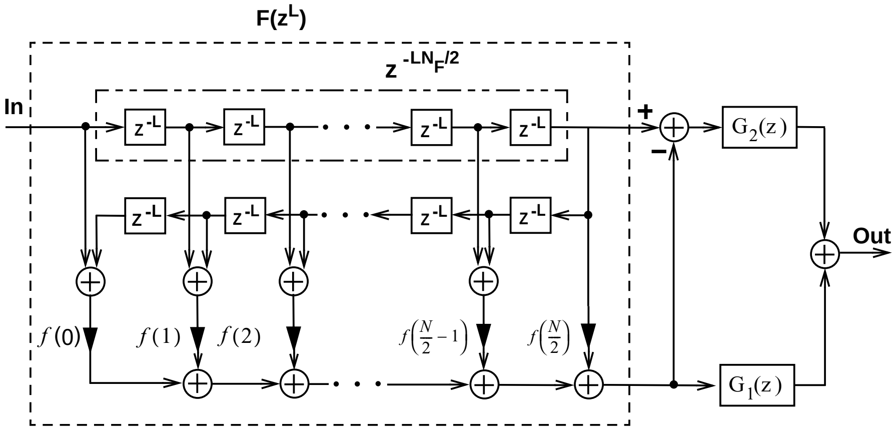


Figure 3.8: A single-stage implementation structure [77].

- orders $N_1 = N_2$ and the number of delays terms $M_1 = M_2 = 0$ given in (3.10) and (3.11).
- if $N_1 > N_2$, $M_1 = 0$ and $M_2 = (N_1 - N_2)/2$
- if $N_1 < N_2$, $M_1 = (N_1 - N_2)/2$, $M_2 = 0$

Hereby, delays for both terms of $H(z)$ become equal and ensure that the phase of the filter of $H(z)$ is linear. Due to the linear-phase condition, the overall frequency response can also be written as

$$H(e^{j\omega}) = H(\omega)e^{-j(LN_F + max(N_1,N_2))\omega/2}, \tag{3.14}$$

where $H(\omega)$ is a zero-phase frequency response and can be expressed as

$$H(\omega) = F(L\omega)G_1(\omega) + (1 - F(L\omega))G_2(\omega), \tag{3.15}$$

where

$$F(\omega) = f(\frac{N_F}{2}) + 2\sum_{n=1}^{N_F/2} f(\frac{N_F}{2} - n)\cos(n\omega) \tag{3.16}$$

and

$$G_k(\omega) = \begin{cases} g(\frac{N_k}{2}) + 2\sum_{n=1}^{N_k} g_k(\frac{N_k}{2} - n))\cos(n\omega), \; N_k \text{ even} \\ \sum_{n=1}^{(N_k-1)/2} g_k(\frac{N_k - 1}{2} - n))\cos((n + 1/2)\omega), \; N_k \text{ odd} \end{cases} \tag{3.17}$$

for $k = 1, 2$. The transfer function in (3.13) can be efficiently implemented as shown in Fig. 3.8, if the coefficient symmetry is exploited.The effectiveness

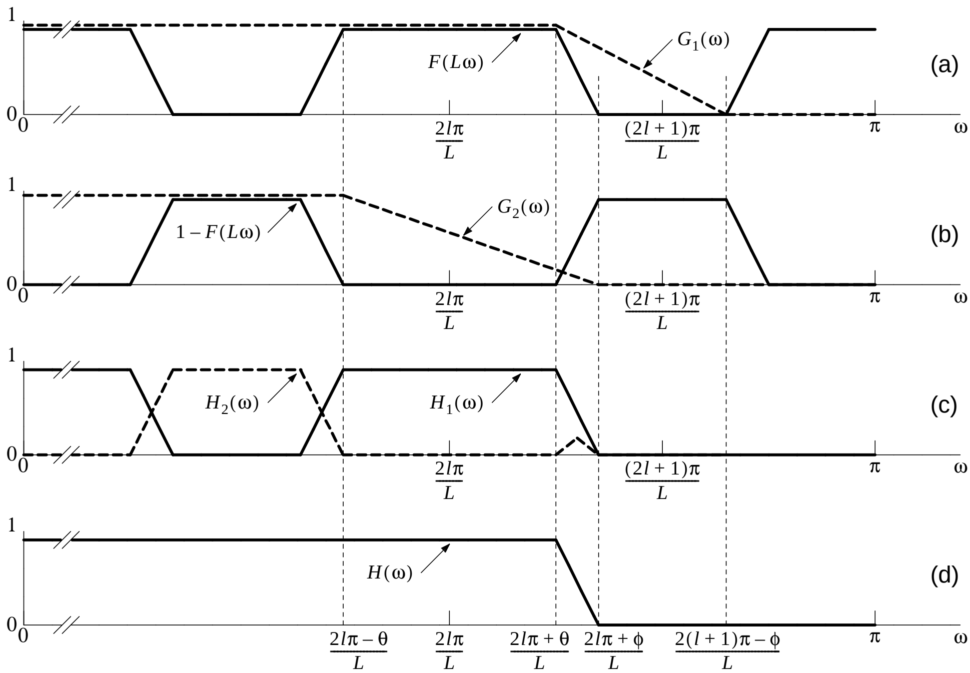


Figure 3.9: Case A design of a lowpass filter using the frequency-response masking technique [77].

of the filter is that of the given frequency-response specification, its effective filter length increases only slightly from that of conventional direct-form design. Due to the substitution of each unit delay with multiple unit delays, the order of the filter increases to LN_F, but because every Lth impulse response value is non-zero, the arithmetic complexity regarding multipliers and adders remains the same. Since a very small fraction of its coefficients are non-zero, its arithmetic complexity is significantly lower than that of the equivalent conventional direct-form realization. This very sparse coefficient-vector leads to very low hardware complexity, round-off noise and coefficient sensitivity. For a lowpass case, the transition band of the overall transfer function can be either the transition band of $F(z^L)$ or $z^{-LN_F/2} - F(z^L)$. If the transition band of $F(z^L)$ is used, the case is referred to as Case A as shown in Fig. 3.9 and if the transition band of $z^{-LN_F/2} - F(z^L)$ is used, the case is referred to as Case B as shown in Fig. 3.10. For both cases the width of the transition bands is $(\phi - \theta)/L$, which means that it is $1/L$th of that of the original prototype filters. The passband and stopband edges of $F(z^L)$ in

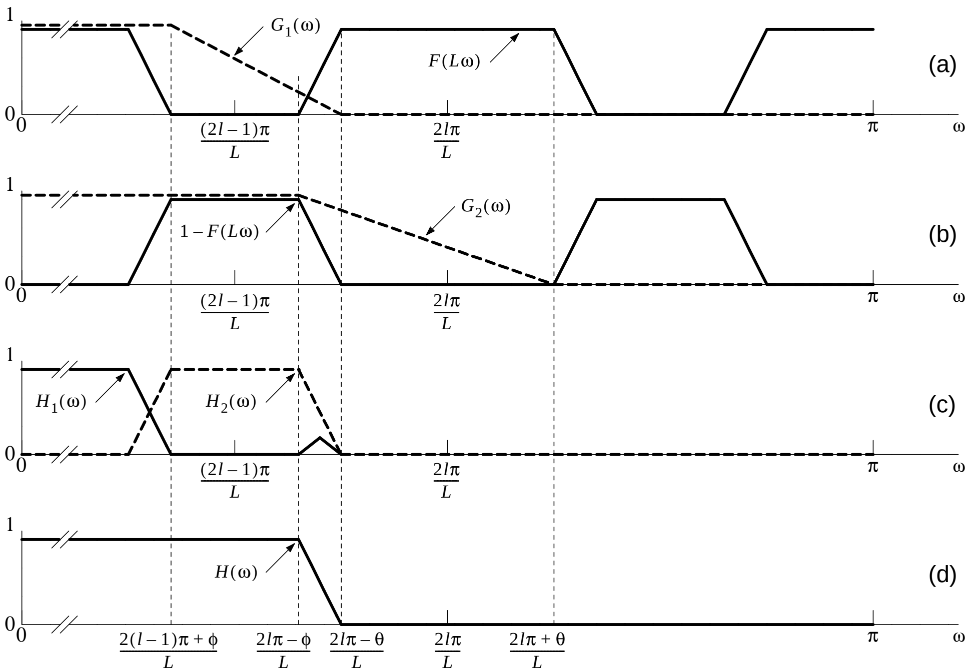


Figure 3.10: Case B design of a lowpass filter using the frequency-response masking technique [77].

Case A are given by

$$\omega_p = (2l\pi + \theta)/L \text{ and } \omega_s = (2l\pi + \phi)/L \tag{3.18}$$

and of Case B as

$$\omega_p = (2l\pi - \phi)/L \text{ and } \omega_s = (2l\pi - \theta)/L, \tag{3.19}$$

respectively, where l is a constant, see Figs. 3.9 and. 3.10. For Case A, the passband and stopband edges of the masking filter with the transfer function $G_1(z)$ are given as

$$\omega_p = (2l\pi + \theta)/L \text{ and } \omega_s = (2(l+1)\pi - \phi)/L \tag{3.20}$$

and those of $G_2(z)$ as

$$\omega_p = (2l\pi - \theta)/L \text{ and } \omega_s = (2l\pi + \phi)/L. \tag{3.21}$$

For Case B, the passband and stopband edges of the masking filter with transfer function $G_1(z)$ are given as

$$\omega_p = (2(l-1)\pi + \phi)/L \text{ and } \omega_s = (2l\pi - \theta)/L, \tag{3.22}$$

and those of $G_2(z)$ as

$$\omega_p = (2l\pi - \phi)/L \text{ and } \omega_s = (2l\pi + \theta)/L. \tag{3.23}$$

To determine the parameters ω_p, ω_s, l, L, θ and ϕ so that the solution fulfills the relation $0 \leq \theta \leq \phi \leq \pi$, the following requirements have to be fulfilled for even order, for Case A:

$$2l\pi/L \leq \omega_p,\ \omega_s \geq ((2l+1)\pi)/L, \tag{3.24}$$

for Case B:

$$(2l-1)\pi/L \leq \omega_p,\ \omega_s \geq (l\pi)/L. \tag{3.25}$$

The number of multipliers is usually lowest when the transition bandwidths are the same and the orders of the masking filters, N_1 and N_2, are the same. The estimated orders are so close to the required order that they can be used to determine the value of L. The orders of $F(z)$ and $G_1(z)$ and $G_2(z)$ can be quite accurately estimated as [83]

$$N_F = \frac{\Phi(\delta_p, \delta_S)}{\phi - \theta}, N_{G1} = \frac{L\Phi(\delta_p, \delta_s)}{2\pi - \phi - \theta}, N_{G2} = \frac{L\Phi(\delta_p, \delta_s)}{\phi + \theta}, \tag{3.26}$$

where $\Phi(\delta_s, \delta_p)$ is given as

$$\begin{aligned}\Phi(\delta_p, \delta_s) = {} & [a_1 \log_{10}(\delta_p)^2 + a_2 \log_{10}\delta_p - a_3]\log_{10}\delta_s \\ & -[a_4 \log_{10}(\delta_p)^2 + a_5 \log_{10}(\delta_p) + a_6]\end{aligned} \tag{3.27}$$

with

$$a_1 = 5.309 \times 10^{-3}, a_2 = 7.114 \times 10^{-2}, a_3 = 0.4761$$

$$a_4 = 2.66 \times 10^{-3}, a_5 = 0.5941, a_6 = 0.4278.$$

The overall order of the resulting filter can be estimated to be $N_F/2 + 1 + \lfloor (N_1+2)/2 \rfloor + \lfloor (N_2+2)/2 \rfloor$ or $N_F + N_1 + N_2$, depending on whether the symmetries on the filter coefficients are exploited or not. The order N_F is approximately $1/L$th of the equivalent conventional direct-form filter. It is very well known that the order of the linear-phase FIR filter is roughly inversely proportional to the transition bandwidth. Since the transition band of the masking filters equals $2\pi/L$, and decreases with increasing L, the masking filter order is usually very low.

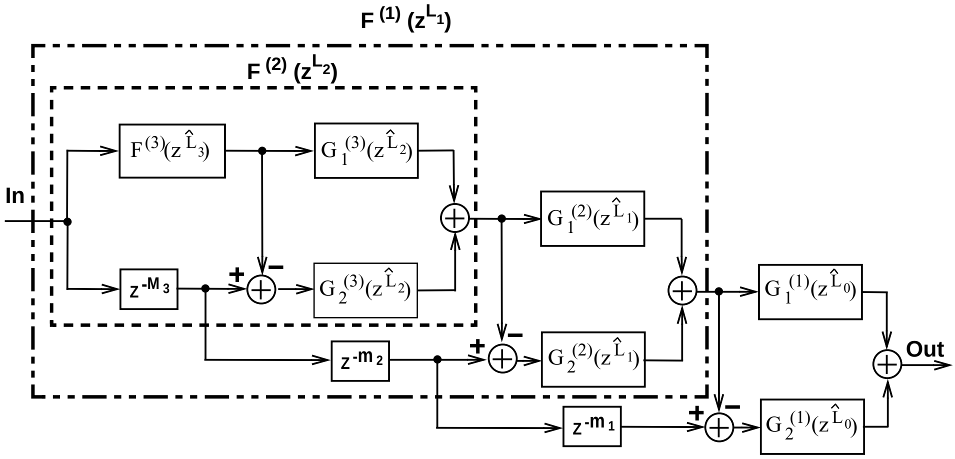


Figure 3.11: An implementation structure using a three-stage frequency-response masking approach [88].

3.3.2 Multistage Approach

The multistage approach [88] is applicable if the order of the prototype filter transfer function, $F(z)$, is too high. The FRM technique can be extended to any arbitrary number of stages implemented recursively. This section briefs the basics of the multistage approach of FRM according to [88]. First, this approach can be used if L can be factorized into $L = L_1 L_2 \cdots L_K$. The overall transfer function is generated as

$$H(z) \equiv F^{(0)}(z) = F^{(1)}(z^{\widehat{L}_1})G_1^{(1)}(z) + [z^{M_1} - F^{(1)}(z^{\widehat{L}_1})]G_2^{(1)}(z)$$

$$F^{(1)}(z^{\widehat{L}_1}) = F^{(2)}(z^{\widehat{L}_2})G_1^{(2)}(z^{\widehat{L}_1}) + [z^{M_2} - F^{(2)}(z^{\widehat{L}_2})]G_2^{(2)}(z^{\widehat{L}_1})$$

$$F^{(2)}(z^{\widehat{L}_2}) = F^{(3)}(z^{\widehat{L}_3})G_1^{(3)}(z^{\widehat{L}_2}) + [z^{M_3} - F^{(3)}(z^{\widehat{L}_3})]G_2^{(3)}(z^{\widehat{L}_2})$$

$$\vdots \quad \vdots \quad \vdots$$

$$F^{(K-1)}(z^{\widehat{L}_{K-1}}) = F^{(K)}(z^{\widehat{L}_K})G_1^{(K)}(z^{\widehat{L}_{K-1}}) + [z^{M_K} - F^{(K)}(z^{\widehat{L}_K})]G_2^{(K)}(z^{\widehat{L}_{K-1}}),$$

where $\widehat{L_1} = 1$ and $\widehat{L_k} = \prod_{n=1}^{k} L_n$ for $k = 1, 2, \ldots, K$, $M_K = \widehat{L}_K N_F^{(K)}/2$ and $M_{K-k} = M_{K-k+1} + m_{K-k}$ for $k = 1, 2, \ldots, K-1$. Fig. 3.11 shows an efficient three-stage implementation structure for the multistage FRM approach.

The order of $F^{(K)}(z)$, is $N_F^{(K)}$ and $N_1^{(k)}$ and $N_2^{(k)}$ are the orders of the masking filters, $G_1^{(k)}$ and $G_2^{(k)}$, respectively.

For the resulting filter to be linear-phase, the following requirements have to be fulfilled [88]:

1. The orders of $G_1^{(k)}(z)$ and $G_2^{(k)}(z)$, for $k = 1, 2, \ldots, K-1$, and that of $F^{(K)}(z)$ have to be even.
2. The orders $N_1^{(K)}$ and $N_2^{(K)}$ can be either even or odd.
3. For $G_2^{(k)}(z)$: if the orders $N_1^{(k)}$, $N_2^{(k)}$ of $G_1^{(k)}(z)$, $G_2^{(k)}(z)$, respectively, are not equal, then $z^{-(N_1^k - N_2^k)/2} G_2^{(k)}(z)$ has to be used instead of $G_2^{(k)}(z)$ and $N_1^{(k)} > N_2^{(k)}$.
4. For $G_1^{(k)}(z)$: if the orders $N_1^{(k)}$, $N_2^{(k)}$ of $G_1^{(k)}(z)$, $G_2^{(k)}(z)$, respectively, are not equal, then $z^{-(N_1^k - N_2^k)/2} G_1^{(k)}(z)$ has to be used instead of $G_1^{(k)}(z)$ and $N_1^{(k)} < N_2^{(k)}$.

3.3.3 FRM approach for Hilbert Transformers

Hilbert transformers are one of the very important classes of FIR filters used in various signal processing applications [9–13, 15, 32, 35, 37, 38, 50, 57, 64–66, 74, 89, 91, 93, 96]. The FRM approach is very efficient for Hilbert transformers if the transition bandwidth is very narrow. This section briefly discusses how to use the FRM approach to realize Hilbert transformers proposed in [50, 51]. The approach uses a correction term produced by masking the periodic filter with a sparse coefficient vector. This correction term is used to sharpen the band edge of a low-order Hilbert transformer. The basic idea when designing this filter is seen in Fig. 3.12 and it is as follows. The band-edge-sharpening filter response, $H_1(e^{j\omega})$, is made periodic with a factor L by replacing each unit delay with L unit delays and the periodicity becomes $2\pi/L$. The masking-filter frequency response $H_L(e^{j\omega})$ is used to mask the unwanted replicas of the passband of the $H_1(e^{jL\omega})$ to produce the frequency response $H_e(e^{j\omega})$, which is the transition-band correction filter. When $H_e(e^{j\omega})$ is added to $H_b(e^{j\omega})$, a very narrow-transition-band Hilbert transformer is obtained. The overall transfer function can be implemented as a parallel connection of two branches as seen in Fig. 3.13, and thereby, given as

$$H(z) = H_1(z^L)H_L(z) + H_b(z),$$

where $H_b(z)$ is a wideband Hilbert transformer, $H_1(z^L)$ is the band-edge-sharpening filter transfer function, and $H_L(z)$ is a masking-filter transfer function. $H_L(z)$ has a very low complexity because its frequency response has a wide transition band.

The total number of nontrivial coefficients, N_{tot}, is given by

$$N_{tot} = N_1 + N_b + N_L = \frac{\Phi_H}{L\Delta} + \frac{\Phi_H}{(1/M) - \Delta} + \frac{\Phi_L}{(1/L) - \Delta}, \tag{3.28}$$

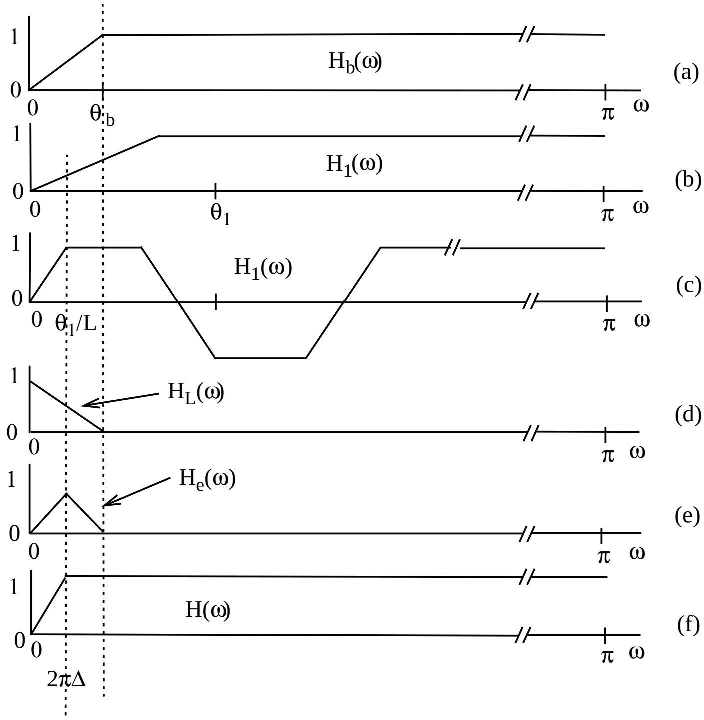


Figure 3.12: Zero-phase frequency responses for subfilters of even length $H_b(z)$. $2\pi\Delta$ is the transition bandwidth of the desired transfer function [50].

where $\Phi_L = 0.22064 - 0.73294\log_{10}(\delta)$, and Φ_Hs are given by

$$\Phi_H = 0.002655(\log_{10}(\delta))^3 + 0.031843(\log_{10}(\delta))^2 - 0.554993\log_{10}(\delta) - 0.049788.$$

The efficiency of the proposed wideband FIR filters with a narrow transition band in Publication-III are compared to the FRM approach, to both one-stage and multistage methods and to Hilbert transformers.

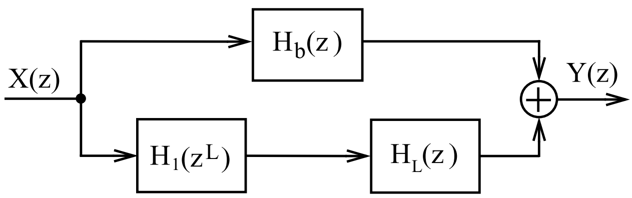


Figure 3.13: Implementation structure for a Hilbert transformer using the FRM technique [50].

3.4 FIR Filters Based on Switching and Resetting IIR Filters

So far the methods described here have been based on the use of complementary filters and/or periodical transfer functions cascaded with interpolator transfer functions. Another approach to synthesize linear-phase FIR filters to reduce the arithmetic complexity is by using FIR filters that mimic the performance of IIR filters. One such method is the principle of switching and resetting between two identical copies of the same IIR filter as introduced in [17] and utilized in [3–5]. The principle of switching and resetting stabilizes the pole-zero cancellation and prevents the quantization errors from growing too much. The purpose of this section is to give a brief review of the principle of switching and resetting [17]. The method is based on IIR filter transfer functions, which are a cascade of stable $G(z)$ and the corresponding unstable $G(z^{-1})$ IIR filter transfer functions. Their impulse response is a shifted and truncated version of the response of $G(z)G(z^{-1})$. These filters can be implemented efficiently by using structures which are a parallel connection of several branches. The truncated version is obtained by using a feedforward term which provides pole-zero cancellation. To briefly explain this idea, consider the stable IIR transfer function

$$G_1(z) = \frac{1}{1 - bz^{-1}}, \tag{3.29}$$

and the unstable pair

$$G_2(z) = \frac{1}{1 - b^{-1}z^{-1}}. \tag{3.30}$$

Pole b in (3.29) is either real or complex and $|b| < 1$ to make the filter $G_1(z)$ stable, which makes $G_2(z)$ unstable. The corresponding FIR filter transfer function of $G_1(z)$ is

$$H_1(z) = \frac{1 - b^N z^{-N}}{1 - bz^{-1}} = \sum_{n=0}^{N-1} b^n z^{-n}, \tag{3.31}$$

and similarly, that of $G_2(z)$ is

$$H_2(z) = \frac{1 - b^{-N} z^{-N}}{1 - b^{-1}z^{-1}} = \sum_{n=0}^{N-1} b^{-n} z^{-n}. \tag{3.32}$$

When cascading the FIR filter transfer functions $H_1(z)$ and $H_2(z)$ in (3.31) and (3.32) the result is a linear-phase FIR filter. The frequency response of $H_1(z)$ can be made as close to that of $G_1(z)$ as desired if N is selected to be appropriately large; the same is true for $H_2(z)$. $H_1(z)$ represents an FIR filter because the pole at $z = b$ is canceled by one of the equispaced zeros on the circle of the radius $|b|$. Similarly, $H_2(z)$ represents an FIR filter because the pole at $z = b^{-1}$ is cancelled by one of the equispaced zeros on the circle of the radius $|b^{-1}|$. However, if the cancellation is inexact due to finite coefficient word length, the transfer function $H_1(z)$ becomes

$$\widehat{H}_1(z) = \frac{1 - \widehat{b}^N z^{-N}}{1 - \widehat{b} z^{-1}}, \tag{3.33}$$

and that of $H_2(z)$ becomes

$$\widehat{H}_2(z) = \frac{1 - \widehat{b}^{-N} z^{-N}}{1 - \widehat{b}^{-1} z^{-1}}, \tag{3.34}$$

$\widehat{b}$ and $\widehat{b}^N$, in 3.33, denote the finite-precision values of b and b^N. Similarly, $\widehat{b}^{-1}$ and $\widehat{b}^{-N}$, in 3.34, denote the finite-precision values of b^{-1} and b^{-N} in $\widehat{H}_2(z)$. The resulting impulse response of $\widehat{H}_1(z)$ is

$$h_1(n) = \begin{cases} \widehat{b}^n, & 0 \le n \le N-1 \\ ((\widehat{b})^N - \widehat{b}^N)(\widehat{b})^{n-N}, & n \ge N \end{cases} \tag{3.35}$$

and that of $\widehat{H}_2(z)$ is

$$h_2(n) = \begin{cases} (\widehat{b}^{-1})^n, & 0 \le n \le N-1 \\ ((\widehat{b}^{-1})^N - \widehat{b}^{-N})(\widehat{b}^{-1})^{n-N}, & n \ge N. \end{cases} \tag{3.36}$$

In both cases, the pole is not exactly removed and its effect appears at the filter output for $n \ge N$. In case of $H_2(z)$, the effect of an inexact pole-zero cancellation, $\widehat{b}^{-N} \neq (\widehat{b}^{-1})^N$, is growing by $|\widehat{b}^{-1}|^n$, for $n \ge N$. From (3.36), it is obvious that even with exact pole-zero cancellation the noise generated by multiplication round-off errors will be large.

To avoid the above-mentioned problems, both $\widehat{H}_1(z)$ and $\widehat{H}_2(z)$ given by (3.33) and (3.34) can be implemented by using the switching and resetting technique proposed in [17]. To explain this idea, first assume that the input e.g. to the $\widehat{H}_1(z)$ consists of a set of $2N$ samples denoted by $x(0), x(1), \cdots, x(2N-1)$. Assume that the last N values $x(N), x(N+1), \cdots, x(2N-1)$ are zero. If $\widehat{H}_1(z)$ worked in an ideal way, i.e., finite word-length effects would not exist, the output would be exactly equal to zero at the time $n = 2N - 1$. This raises the idea to reset the state variables at the time $n = 2N - 1$ and thereby, to avoid the effect of inexact pole-zero cancellation from growing too much.

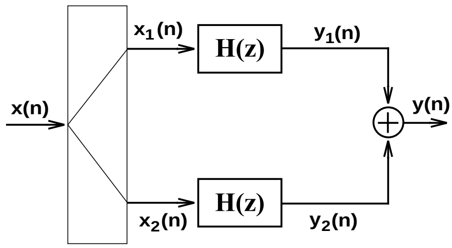


Figure 3.14: Implementation structure based on switching and resetting, where a demultiplexer is used to decompose $x(n)$ into two signals $x_1(n)$ and $x_2(n)$, $H(z)$ can be $H_1(z)$ or $H_2(z)$.

To extend this idea to arbitrary inputs, the input $x(n)$ is divided into two subsequences as follows.

$$x(n) = x_1(n) + x_2(n), \tag{3.37}$$

where

$$x_1(n) = \begin{cases} x(n), & rN \le n \le (r+1)N - 1, \quad r = 0, 2, 4, \ldots \\ 0, & \text{otherwise} \end{cases} \tag{3.38}$$

and

$$x_2(n) = \begin{cases} x(n), & rN \le n \le (r+1)N - 1, \quad r = 1, 3, 5, \ldots \\ 0, & \text{otherwise.} \end{cases} \tag{3.39}$$

The subsequencies $x_1(n)$ and $x_2(n)$ are fed into two identical copies of $H_1(z)$ and the outputs are added as seen in Fig. (3.14). This is equivalent to filtering the whole sequence $x(n)$ by $H_1(z)$ as implied by the superposition in (3.38) and (3.39). The only difference is the finite word-length effects. Because in every Nth data set, there is a set of N zero-valued samples, it is possible to reset the filter at the time $n = 2N - 1$, i.e., when the last zero-valued sample enters the filter, to avoid the output noise from growing too much.

Filter synthesis methods introduced in Publication-II and Publication-III use the principle of switching and resetting in practical implementations.

3.5 Thinning Digital Filters: A Piecewise-Exponential Approximation Approach

Boudreaux and Parks [7] proposed an algorithm to synthesize recursive digital FIR filters requiring only a few multipliers. The approach to reduce the number of multipliers is based on thinning the filter. Thinning means that a filter of several zero-valued coefficients is obtained by removing some of the non-zero coefficients. This creates a sparse filter vector. According to Boudreaux and Parks, thinning comes from antenna theory, where a thinned antenna array is an array of non-uniformly spaced elements derived from an equally spaced array by systematically removing certain elements. Various optimal criteria are used to choose the elements to be removed. Dynamic programming techniques are used to find the best least-squared piecewise-exponential approximation to a desired impulse response of length $N+1$. These filters are implemented by using recursive structures with FIR and IIR filter sections. Because of the recursive implementation, the number of arithmetic operations is independent of the filter length $N+1$ and dependent only on the number of pieces or segments S used in the approximation. The number of segments is much smaller than the filter length, i.e., $S << N+1$. The cascade implementation of finite-length piecewise approximation reduces the number of multiplications to a maximum of $2(S+2)$ per output sample. Symmetric piecewise-exponential sequences need only $(S+4)$ multiplications. Compared to a direct-form FIR filter this approach requires only 1/2 up to 1/4 of the number of multiplications to implement the filter.

3.6 The Piecewise-Polynomial Approach by Chu and Burrus

This section reviews briefly the basic idea of the original Chu-Burrus [11,12] approach to synthesize piecewise-polynomial-(sinusoidal) impulse responses for a lowpass case. The piecewise technique is based on a time-domain approximation and a frequency-domain optimization to obtain the filter coefficients. The original Chu-Burrus approach starts with ideal impulse responses with infinite duration for lowpass filters as follows:

$$h_{id}(n) = \frac{\sin(\omega_c n)}{\pi n}, \quad -\infty < n < \infty, \tag{3.40}$$

where ω_c is the cutoff frequency. To make it an odd-length finite duration filter, it is cascaded with a finite-duration window function as

$$h(n) = w(n)h_{id}(n), \tag{3.41}$$

where $w(n) = 0$ for $|n| > N - 1/2$ and $w(0) = 1$. The impulse response of a practical odd-length filter is

$$h(n) = \frac{w(n)}{\pi n} \sin(\omega_c n), \tag{3.42}$$

where ω_c is the cutoff frequency and $w(n)$ is a window function. Thereby, when using piecewise-polynomials/-polynomial-sinusoidals to represent the impulse response in (3.42), we have

$$h(n) = E(n) \sin(\omega_c n), \tag{3.43}$$

where the envelope $E(n) = w(n)/\pi n$ is approximated with a piecewise polynomial. In a conventional design with the window technique, the window is the same for all lowpass filters with different cutoff frequencies. This corresponds to the new filter using the same envelope but different impulse response oscillating frequencies for different cutoff lowpass filters.

A piecewise polynomial is used to represent the impulse response of narrowband filters. Equivalently, piecewise-polynomial-sinusoidals are used to represent the impulse response of wideband filters according to [12].

The filters are optimized by using the Fletcher-Powell algorithm, which minimizes the error norm l_p of

$$E = \sum_{i=0}^{L-1} [W(\omega_i)(H(\omega_i) - H_{id}(\omega_i))]^{2p}, \tag{3.44}$$

Since this algorithm converges for low values of p, the algorithm starts with $p = 1$ and uses the result as a starting point for the next round of optimization for higher values of p. This approach does not utilize the coefficient symmetry for linear-phase FIR filters in implementation.

Chapter 4

FIR Filters with a Piecewise-Polynomial-(Sinusoidal) Impulse Response

This chapter is based on Publication-I – Publication-IV, where synthesis schemes are introduced by using piecewise-polynomial or piecewise-polynomial-sinusoidal impulse responses and recursive implementation structures. Two different approaches of the piecewise notion are exploited in this thesis. The aim is first to explain why piecewise polynomials are very well suited to create impulse responses for linear-phase FIR filters. Secondly, the aim is to cover how the notion of piecewise is utilized in Publication-I – Publication-IV. The names of the variables used in this chapter are the same as in the above-mentioned publications.

4.1 Proposed Piecewise-Polynomial Methods

4.1.1 Why Piecewise-Polynomials?

If we consider an impulse response of an optimum linear-phase FIR filter, it is seen in Fig. 4.1(a) that the impulse response has a very smooth shape, which means that there is a strong correlation between successive impulse-response values.

Because of the smoothness of the impulse response and the smoothness of polynomial shapes (as seen in Fig. 4.2), we use these characteristics to our advantage to reduce the number of coefficients and to obtain efficient implementation structures. Therefore, a piecewise use of polynomials is motivated to create such impulse responses.

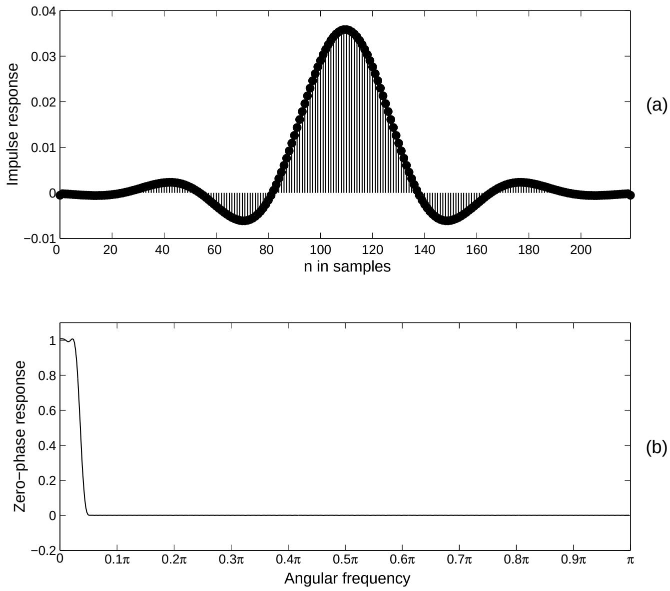


Figure 4.1: Typical (a) impulse response and (b) zero-phase frequency response for a narrowband linear-phase FIR filter. The filter has been optimized by using the Remez multiple exchange algorithm and it has the minimum order, $N = 215$, to meet the specifications: Passband edge $\omega_p = 0.025\pi$, stopband edge, $\omega_s = 0.05\pi$, passband ripple $\delta_p = 0.01$, stopband ripple $\delta_s = 0.001$.

4.1.2 The Concept of Piecewise

Two different piecewise- concepts are utilized: one used in Publication-I and Publication-IV and an approach according to Definition 1.4.6 used in Publication-II and Publication-III. We start with the approach utilized in Publication-I and Publication-IV and proceed to the approaches proposed in Publication-II and Publication-III.

A function is said to be piecewise if it contains a finite number of functions connected together either with or without discontinuities [104]. Publication-I and Publication-IV use this notion of piecewise when creating the impulse response in the time-domain by using polynomials. The polynomials are of the form:

$$f_l(n) = \left[\frac{n-(N-1)/2}{(N-1)/2}\right]^l \text{ for } n = 0, 1, \ldots, N-1, \tag{4.1}$$

where N is the length of the block and l is the order of the polynomial. Some

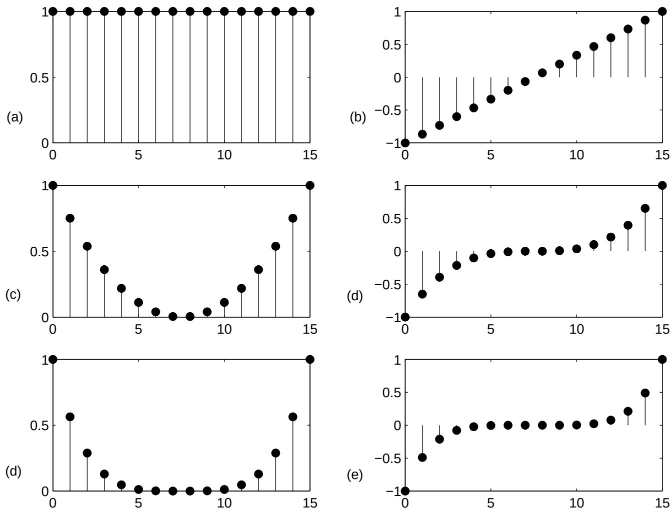


Figure 4.2: Basis functions given by (4.1) when $N = 16$: (a) $l = 0$, (b) $l = 1$, (c) $l = 2$, (d) $l = 3$, (e) $l = 4$, (f) $l = 5$.

of the polynomial shapes are shown in Fig. 4.2. Definition 1.4.6 of piecewise given in Chapter 1 is utilized in Publication-II and in Publication-III. The polynomials are of the form

$$p_k^{(L)}(n) = \sum_{r=0}^{L} a_m^{(L)}(r) n^r, \text{ for } 0 \leq n \leq N-1. \tag{4.2}$$

where L is the degree of the polynomial and N is the length of the slice (see Publication-II Eq.(30)-(31)). Each polynomial cover one slice according to Definition 1.4.7.

4.1.3 The Concept of Piecewise by Saramäki and Mitra

This subsection explains the concept of the piecewise used in the approach by Saramäki and Mitra [85]. In this approach, the overall impulse response is divided into M blocks of length L in each subresponse. The number of subresponses is $R+1$, where R is the degree of a polynomial, Fig. 4.3. Each block is created by polynomials of a given degree, R. In this approach, the number of multipliers is $2\lfloor((M+1)/2\rfloor(R+1) + (M - 2\lfloor M/2\rfloor)\lfloor(R+2)/2\rfloor$. When exploiting the linear-phase FIR filter symmetry property we have $\lfloor((M+1)/2\rfloor(R+1)+(M-2\lfloor M/2\rfloor)\lfloor(R+2)/2\rfloor$ distinct filter coefficients. This approach is referred to as a "blockwise" use of polynomials in this thesis.

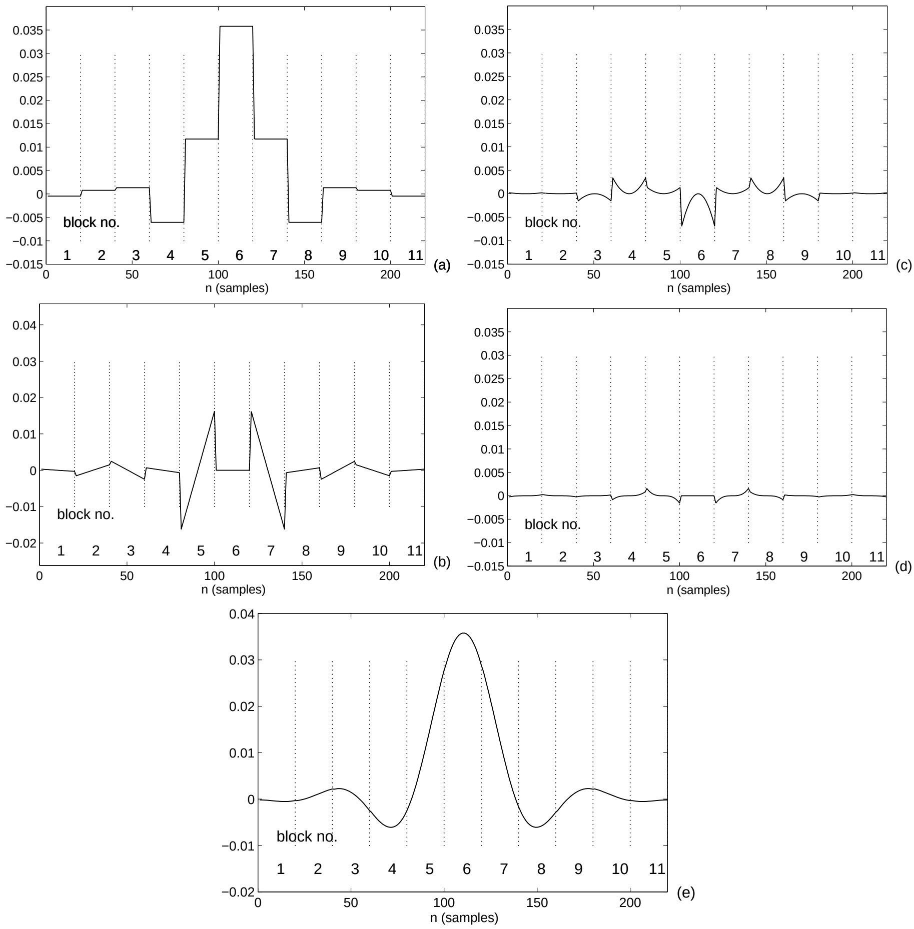


Figure 4.3: Impulse responses for the optimized filter for $M = 11$ blocks with length $L = 20$ in each block and $N = ML = 220$, and $R = 3$ with the design criteria: $\omega_p = 0.025$, $\omega_s = 0.05$, $\delta_p = 0.01$, and $\delta_s = 0.001$: (a) The first subimpulse response by a zeroth-order polynomial in each block.(b) The second subimpulse response by a first-order polynomial in each block. (c) The third subimpulse response by a second-order polynomial in each block. (d) The fourth subimpulse response by a third-order polynomial in each block. (e) The overall impulse response.

Consider an example, where $M = 11$ and $L = 20$, and $R = 3$. In this case the number of subresponses is four. Every block in the first subresponse is created by a polynomial of degree zero, $f_0(n)$. Similarily, each block in the second subresponse is created by the first-degree polynomial, $f_1(n)$.

Every block in the third subresponse is created by using the second-degree polynomial, $f_2(n)$. Finally, the last, i.e., the fourth subresponse is created also blockwise by using the third-degree polynomial, $f_3(n)$.

Depending on the desired linear-phase type, after the center of symmetry of the impulse response, the subresponse, see e.g. Fig. 4.2 (a), is created either symmetrically or anti-symmetrically. After optimization by summing these subresponses the overall impulse response is formed.

The overall impulse response with M blocks is shown in Fig. 4.3(e). If a third-degree polynomial is used as an approximating function, the basis functions of order 0, 1, 2 and 3 shown in Figs. 4.2 (a), (b), (c) and (d) are used to create the subimpulse responses in a blockwise manner shown in Figs. 4.3(a), 4.3(b), 4.3(c) and 4.3(d) in order to obtain the overall impulse response as shown in Fig. 4.3(e) by summing the subimpulse responses.

The differentiator design approach proposed in Publication-IV is based on the Saramäki-Mitra approach by using anti-symmetrical piecewise-polynomial impulse responses.

4.1.4 The Concept of Piecewise by Lehto, Saramäki and Vainio

This section explains the piecewise concept utilized in Publication-II and Publication-III. The definition of piecewise exploited in Publication-II and Publication-III is given in Definition 1.4.6, where the polynomials are defined by (4.2). To describe the basic idea, consider the following example with the number of slices $M = 5$, and the polynomial degree $L = 3$. The idea is to divide the impulse response into M slices defined by (1.4.7) and to generate each slice with polynomials of a given degree L and of length N_m, for $m = 1, 2, \ldots, M$. The slices are of different lengths and they form the subresponses as shown in Fig. 4.4, in the case $M = 5$. Thereby, each block of the impulse response consists of a different number of polynomials, which are shown in Fig. 4.4: e.g. the first block consists of one polynomial, the second block of two polynomials, and the (left) middle block of the impulse response consists of five polynomials. Due to the symmetry, there are altogether ten blocks in the overall impulse response. After summing up the subresponses in Fig. 4.4, the overall impulse response is obtained and shown in Fig. 4.5. This approach is thus referred to as a "slicewise" use of polynomials in this thesis. It has the same smooth and piecewise-polynomial shape as the optimum linear-phase FIR filter in Fig. 4.1(a).

When designing an FIR filter with a piecewise-polynomial-sinusoidal impulse response in Publication-III, the polynomials in each slice are multiplied with a sinusoidal function depending on a linear-phase type to create the

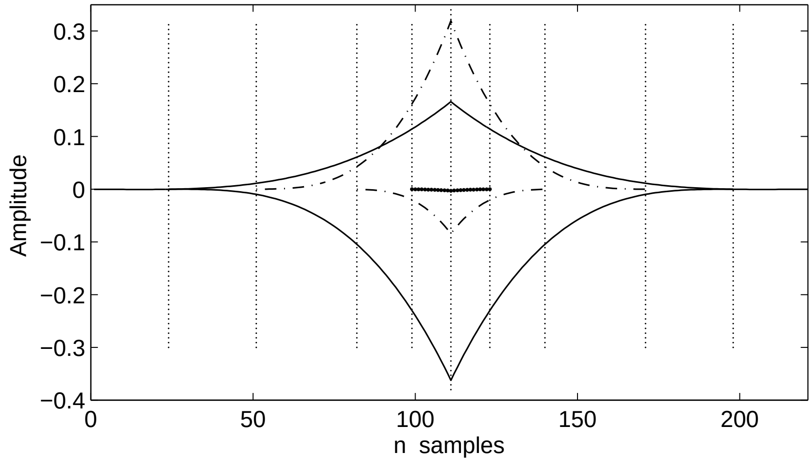


Figure 4.4: Subimpulse-responses, solid up, the longest subresponse, solid down, the second longest, dashdotted up and down, the third and fourth longest subresponses, and the shortest subresponse in the middle. The vertical lines mark the block edges. The first block consists of one polynomial, the second one of two polynomials, the third one of three polynomials, the fourth one of four and the (left) middle block of five polynomials.

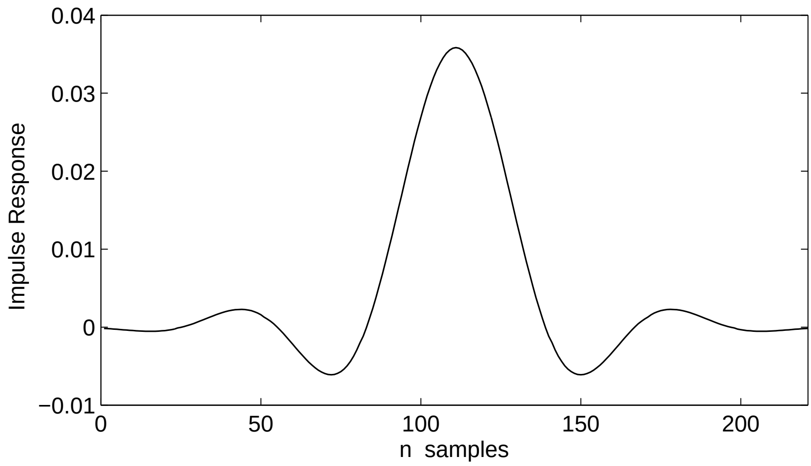


Figure 4.5: Overall impulse response obtained by summing up the five subresponses of Fig. 4.4.

subresponses by optimization. In the case of a Type 4 linear-phase FIR filter the subresponses lack the sinusoidal part (see Publication-III). Thereby, they coincide with the piecewise approach in Publication-II.

4.1.5 Implementation of the Saramäki - Mitra Approach

The approach proposed in [83] is implemented recursively by using a parallel connection of several branches. These branches create the subresponses in the filtering operation. The efficiency of the structures is based on the parallel implementation of the overall transfer function in the form $G_l(z^L)F_l(z)$, where each $F_l(z)$ requires no real multipliers. This can be obtained by making the coefficients in $F_l(z)$s integer-valued. Additionally, by using integer-valued coefficients with modulo arithmetic e.g. fixed-point arithmetic (two's complement arithmetic) and worst-case scaling, the proposed structures in Publication-I and Publication-IV, do not suffer from overflows and harmful effects from miscalculations disappear at the filter output in finite time. Thereby, the structures do not need initial resetting. In Publication-I, the formulas for making $F_l(z)$s integer-valued for efficient implementations are proposed. The differentiators proposed in Publication-IV can be implemented by using the recursive structures of the Saramäki–Mitra approach.

4.1.6 Implementation of the Proposed Approaches

This section considers the implementation of the filters, which are proposed in Publication-II and Publication-III. The proposed filters are implemented by using recursive structures with the aid of accumulators. There are feedback loops at the end of the structure and thereby a pole in each feedback loop at $z = 1$. This means that there might occur some problems during the filtering operation, e.g., the pole-zero cancellation is not exact in a practical implementation. These problems can be avoided as follows. Before starting to use the structures, the state variables should be reset. Pole-zero cancellation should be done in the overall structure by quantizing the polynomial coefficients before deriving the coefficients in the structures. Furthermore, to avoid overflows, the worst-case scaling described in section 2.4 is required in combination with two's complement arithmetic. There are no overflows in the structures provided that the actual rounding or truncation is performed after accumulators. The accumulators have a double word length.

Additionally, the structure does not automatically recover from temporary data errors in the accumulators. Such a data corruption can be caused by overflows, power surges, system start-up, and random errors in general. These problems including the inexact pole-zero cancellation can be avoided by using a parallel structure following the principle of switching and resetting stated in section 3.4. The structure shown in Fig. 3.14 can be used in the implementation. In this case, if the overall filter order is N, the input sequence is split into sets of N data samples so that the first set, third set and so on are fed to the upper branch, whereas the remaining sets are fed to the lower branch. The performance of the overall system is guaranteed by resetting the state variables in the upper (lower) branch at time instants $n = 2\rho N - 1$ ($n = (2\rho - 1)N - 1$), where ρ is an integer. In this case, rounding or truncation is allowed after the multiplications by the filter coefficients. The switching and resetting principle can also be implemented by using multiplexing and one structure. This means that single delays are replaced by double delays, i.e., the number of delay elements is doubled. The structure in Fig. 7 of Publication-III, has poles on the unit circle, i.e., $e^{j\omega_c}$, which leads to instability problems. One way of dealing with the problem is by slightly moving the $e^{j\omega_c}$s off the unit circle by adding a real-valued coefficient $r < 1$ as $re^{j\omega_c}$ and the switching and resetting principle as stated in section 3.4. It may cause more difficulties in implementation regarding the pole-zero cancellation.

Practical implementation according to Publication-II requires, for Type 1, $(L+1)M + \lfloor (L+1)/2 \rfloor$ multipliers, $2M(L+1)$ adders, $2M(L+1) + L + \lfloor (L+1)/2 \rfloor$ additions, $(M+1)L + 1 + \lfloor (L+1)/2 \rfloor + N + 1$ delays, where L is the polynomial degree, M is the number of slices and N is the filter order. The number of additions, multiplications and delays is approximately the same for all linear-phase types. The difference between the structures is caused by the mid section, from which it follows that the number of additions differs approximately by one between the linear-phase types. As a simple example of arithmetic complexity, consider Case 2 in Publication-II with the specifications: $\omega_p = 0.00625$, $\omega_s = 0.0125$, $\delta_p = 0.01$, $\delta_s = 0.001$ (see also Table 4.2). The criteria were met by the filter order 870, the number of polynomial slices $M = 8$ with the polynomial degree $L = 3$ in each slice. In this case the arithmetic complexity adds up to 34 multiplications, 70 additions, 64 adders and 901 delays.

The practical implementation according to Publication-III has the number of multiplications, additions, adders and delays increased by the number of complex multipliers and by the arithmetic elements and operations caused by the direct-form FIR filter section (see Publication-III, Fig. 7.). The number of complex multipliers is, $M(L+2) + L + \lfloor (L+2)/2 \rfloor$, the number of adders in direct-form filter section is $T-1$, the number of multipliers $\lceil T/2 \rceil$, when exploiting the coefficient symmetry, the number of delays $T + N + 2$, and the number of additions is $T-1$, where T is the length of the direct-form FIR filter and N is the order of the recursive filter section.

4.1.7 Summary of Comparisons of the Methods

This section summarizes the arithmetic complexity of the proposed filters in Publication-II and Publication-III by comparing them to the IFIR and FRM approaches and the Saramäki–Mitra approach [39, 85] regarding Structures A and B as seen in Tables, 4.1, 4.2, 4.3, and 4.4. As can be seen in Tables 4.2, 4.3, and 4.4, the proposed approaches require less multipliers and adders than IFIR and FRM [79, 80, 82] implementations. It should be noted that the switching and resetting principle stated in section 3.4 affects the arithmetic complexity. There may be several ways of perfoming the switching and resetting, at least one is according the section 3.4, and another is by multiplexing the structure.

Additionally, if multiplexing is used, the proposed structures are forced to operate at a double sampling rate, which consumes more power. Regarding the arithmetic complexity of the 4.1 Saramäki–Mitra approach in Table 4.1, delays and adders can be shared between the branch transfer functions $G_0(z^L)$, $G_1(z^L)$, $G_2(z^L)$ and $G_3(z^L)$ before using the coefficients of $F_l(z)$ (see Structure A and B in Publication-I and Publication-IV).

Table 4.1: Arithmetic Complexity for Case 1 Design Criteria in Publication-II, i.e., $\omega_p = 0.025\pi$, $\omega_s = 0.05\pi$, $\delta_p = 0.01$, $\delta_s = 0.001$.

Filter structure	Number of multipliers	Number of adders/ additions	Number of delays
The lowpass filter in Publication-II	22	42/45	242
IFIR	24	45	227
Saramäki-Mitra Structure B	22+10	39	224
Saramäki-Mitra Structure A	22+7	50	233
Direct-form	109	216	216

In Table 4.2 the design result of IFIR is obtained with the parameter $L = 19$. The orders of $F(z^L)$ and $G(z)$ are 855 and 40, respectively.

In Table 4.3(*), the multiplication of two complex numbers can be performed either by using 1) four real multiplications, one addition, and one

Table 4.2: Arithmetic Complexity for Case 2 Design Criteria in Publication-II, i.e., $\omega_p = 0.00625\pi$, $\omega_s = 0.0125\pi$, $\delta_p = 0.01$, $\delta_s = 0.001$.

Filter structure	Number of multipliers	Number of adders/ additions	Number of delays
The lowpass filter in Publication-II	34	66/69	901
IFIR	44	84	834
Direct-form	432	863	863

Table 4.3: Arithmetic Complexity for Type 1 Design Criteria in Publication-III, i.e., $\omega_p = 0.4\pi$, $\omega_s = 0.402\pi$, $\delta_p = 0.01$, $\delta_s = 0.001$.

Filter structure	Number of real multipliers	Number of complex multipliers$^{(*)}$	Number of adders/ additions of real numbers$^{(**)}$	Number of delays
Type 1 lowpass filter in Publication-III	37	36	70/76	3081
FRM, one-stage	168	-	330	2690
FRM, two-stage	107	-	204	2920
FRM, three-stage	94	-	174	3196
Direct-form	1280	-	2558	2558

Table 4.4: Arithmetic Complexity for the Hilbert Transformer in Publication-III, i.e., the passband region $[0.00125\pi, 0.99875\pi]$ and $\delta_p = 0.0001$.

Filter structure	Number of multipliers	Number of adders/ additions	Number of delays
The Hilbert Transformer in Publication-III	48	90/96	4146
FRM	107	211	3056

subtraction additions, or by using 2) three real multiplications, two additions, and three subtractions. In Table 4.3(**), additions, which complex multipliers cause, are not taken into account.

Chapter 5

Summary of the Results and Discussion

In this thesis new optimization-based methods were introduced to synthesize digital linear-phase FIR filters. Furthermore, formulas and algorithms for the integer-valued implementation coefficients of $F_l(z)$ for the dedicated implementation structures of linear-phase FIR filters with a piecewise-polynomial impulse response by Saramäki and Mitra, were introduced in Publication-I. The algorithms make the implementation coefficients of $F_l(z)$ integer-valued so that their number is minimized. The algorithms are based on alteration of the basis functions. It was shown that in order to achieve efficient implementations the following properties should be satisfied. First, provided that fixed-point modulo arithmetic (e.g. one's or two's complement arithmetic) and worst-case scaling is used, the outputs are correct even though internal overflows may occur in the accumulators. Secondly, there is no need for resetting and the effect of temporary miscalculations vanishes from the outputs in finite time.

In Publication-II a new method to synthesize narrowband linear-phase FIR filters with a piecewise-polynomial impulse response was introduced. The method is based on the creation of impulse responses in the time domain with the aid of polynomials with a given degree in a slicewise manner. The filters are optimized in the frequency domain. The arithmetic complexity in optimization depends on the number of slices and the polynomial degree. In the practical implementation additional coefficients are needed, i.e., one coefficient for every other polynomial degree, referred to as β-coefficients in Publication-II. The recursive implementation structures were introduced for all four linear-phase types. In Appendix A, a derivation of implementation structures for Types 3 and 4 is given. It was advisable that the pole-zero cancellation in the feedback loops can be ensured by putting two structures in parallel so that half of the sample values are fed in the upper structure and the other half in the lower structure and by resetting the whole system at the time instant $2N-1$. Furthermore, it was shown by means of examples that savings in arithmetic complexity, compared to other methods for narrowband linear-phase FIR filters, were achieved.

In Publication-III a new method to synthesize wideband linear-phase FIR filters with a piecewise-polynomial-sinusoidal impulse response was proposed. The idea of the method arises from a windowing technique and it is an extension of the method introduced in Publication-II. The design scheme is extended by multiplying the polynomials with a sinusoidal function in the time-domain. It was shown that the implementation structure for Type 1 in Publication-II was extended by adding complex multipliers. Additionally, it was shown that for Type 4 the structure is the same as that proposed in Publication-II. For Type 4, the arithmetic complexity in optimization depends on the number of slices and the polynomial degree. For Type 1, the arithmetic complexity depends also on the number of complex multipliers. Furthermore, it was shown by means of two examples that savings in the arithmetic complexity were achieved compared to the FRM technique.

In Publication-IV a differentiator design approach was proposed based on the piecewise-polynomial Saramäki–Mitra approach. The efficiency of the approach in the case of differentiators was shown by the means of an example. The number of non-integer multipliers was found to be approximately the same as with the approach proposed in Publication-II. Differentiators were designed to be Type 3 or 4 linear-phase FIR filters.

In Publication-II and Publication-III the results regarding arithmetic complexity in terms of the number of multipliers required to meet the design specifications were compared to other computationally efficient approaches. Narrowband linear-phase FIR filters proposed in Publication-II were compared to the IFIR approach regarding the number of multipliers needed to meet the filter design requirements. It was found that the proposed filters require fewer multipliers than equivalent IFIR filters. It was observed that the narrower the transition band gets, the bigger the difference grows in terms of the number of multipliers required to meet the given specifications in favor to the proposed approach.

It is known according to [22] that the number of multipliers consume a significant portion of overall power. Of course the power dissipation also depends on the actual hardware implementation and the word length used in multipliers. There is a large number of ways to implement on the hardware level. The large number of multipliers also makes the silicon area large in VLSI implementations, which results in more costly implementations. Thereby, the number of multipliers is of importance when developing computationally efficient synthesis methods. The scope of the thesis is not on the hardware level, so we only note the hardware issues on the overall level and not more deeply than this.

Even though future work described below includes investigating the coefficient sensitivity, already some preliminary results have been obtained for narrowband filters in Publication-II. Case 1 lowpass filter in Publication-II requires $1 + 33$ bits. The effective number of bits is 21 because there are at least 12 leading zeros in each coefficient. It is natural to think that if the required number of coefficients is approximately one tenth compared to the direct-form equivalent case, the number of bits required has to be higher. This follows from the fact that a lot fewer coefficients have to carry the same amount of information as in the direct-form case. That is why there necessarily have to be more bits required in a practical implementation. The implementation also requires that the optimization of filter coefficient values has to be done carefully by tuning in the optimization tool. In this manner, the obtained impulse response of the filter is close enough to the theoretical impulse response, i.e., most of the implemented impulse response values have an error approximately equal or less than 10^{-15} compared to the theoretical one.

The approach proposed in Publication-II is not suitable to synthesize wideband filters because the number of required polynomial slices needed would be impractical. When the polynomials are multiplied with a sinusoidal function, the number of polynomials needed becomes smaller. Therefore, the proposed approach in Publication-III is suitable to synthesize wideband linear-phase FIR filters. The proposed wideband filters are compared to the frequency-response masking approach. The proposed filters require fewer multipliers to meet the given filter design specifications than the frequency-response masking equivalent FIR filters.

The piecewise use of polynomials in Publication-I and Publication-IV can also be considered to be compressive in nature because of the following observations. As seen in Fig. 4.3, the first subresponse with the zeroth-order polynomial contains most of the energies, the second subresponse created by the first-order polynomial contains the second most of the energies and so forth. Thereby, it is enough to use polynomials of the lowest degrees to create the filter, and the high-degree subresponses can be discarded.

Future work includes deriving estimation formulas for the filter parameters, N_ms, of the filters introduced in Publication-II and Publication-III. Additionally, even more research work is going to be done to investigate the finite-word-length effects of the proposed filters in Publication-II and Publication-III by using Alterra boards especially with regard to coefficient sensitivity and output noise. Also, other synthesis methods are going to be derived for applications requiring narrowband and computationally efficient realizations. The aim is to obtain even more efficient narrowband realizations especially with regard to finite-word-length effects, coefficient sensitivity, sampling rate, silicon area and power dissipation. Also, finite-word-length effects of the proposed filters in Publication-IV and in Publication-I could be investigated.

Appendix A

Implementation Structures

Structures for Type 3 and 4 FIR filters with a Piecewise-Polynomial Impulse Response The purpose of this chapter is to show derivations for Type 3 and 4 piecewise-polynomial FIR filter structures proposed in Publication-II. These derivations are not given in Publication-II.

Based on the design technique proposed in Publication-II, the linear-phase FIR filter transfer functions for Types 3 and 4 are of the form:

$$H(z) \equiv H^{(0)}(z) = \begin{cases} \displaystyle\sum_{n=0}^{N-1} h^{(0)}(n)[z^{-n} - z^{-(2N-n)}] & \text{for Type 3} \\ \displaystyle\sum_{n=0}^{N-1} h^{(0)}(n)[z^{-n} - z^{-(2N-1-n)}], & \text{for Type 4,} \end{cases} \tag{A.1}$$

where N is an integer and $h^{(0)}(n)$ is given by

$$h^{(0)}(n) = p^{(L)}(n) = \sum_{r=0}^{L} a^{(L)}(r)n^r \quad \mathit{for} \quad 0 \le n \le \widehat{N}_0 \tag{A.2}$$

with

$$\widehat{N}_0 = \ N - 1. \tag{A.3}$$

In (A.2), $p^{(L)}(n)$ is an Lth-degree polynomial. Hence, there exist $L+1$ unknowns in the above transfer functions. In (A.1), $h^{(0)}(2N-n) = -h^{(0)}(n)$ for $n = 0, 1, \ldots, 2N$; and $h^{(0)}(2N-1-n) = -h^{(0)}(n)$ for $n = 0, 1, \ldots, 2N-1$; respectively. Note that due to the above symmetry condition, $h^{(0)}(N) = 0$ for Type 3. In order to implement efficiently the above $H(z)$ in terms of accumulators, starting with

$$p^{(L)}(n) = \sum_{r=0}^{L} a^{(L)}(r)n^r, \tag{A.4}$$

the following polynomials are determined recursively for $k = 1, 2, \ldots, L$:

$$p^{(L-k)}(n) = \\ p^{(L-k+1)}(n+1) - p^{(L-k+1)}(n) = \sum_{r=0}^{L-k} a^{(L-k)}(r) n^r, \tag{A.5}$$

where

$$a^{(L-k)}(r) = \sum_{s=0}^{L-k-r} \binom{L-k+1-s}{r} a^{(L-k+1)}(L-k+1-s). \tag{A.6}$$

In order to develop proper structures to implement the above transfer functions in terms of accumulators, the first step is to express these transfer functions in the following form:

$$H^{(0)}(z) = E^{(1)}(z)/(1 - z^{-1}). \tag{A.7}$$

This $E^{(1)}(z)$ can be written as

$$E^{(1)}(z) = \sum_{n=0}^{\widetilde{N}} e^{(1)}(n) z^{-n}, \tag{A.8}$$

where

$$\widetilde{N} = \begin{cases} 2N+1 & \text{for Type 3} \\ 2N & \text{for Type 4} \end{cases} \tag{A.9}$$

and

$$e^{(1)}(n) = h^{(0)}(n) - h^{(0)}(n-1), \text{ for } n = 0, 1, \ldots, \widetilde{N}. \tag{A.10}$$

Based on the impulse responses of the original transfer functions, the resulting impulse responses of $E^{(1)}(z)$ for Types 3 and 4 can be expressed, after some manipulations, in terms of the polynomials $p^{(L)}(n)$ and $p^{(L-1)}(n)$, given by (A.4)–(A.6), as follows:

$$e^{(1)}(n) = \begin{cases} p^{(L)}(0), & n = 0 \\ p^{(L-1)}(n-1), & n = 1, 2, \ldots, N-1 \\ -p^{(L)}(N-1), & n = N \text{ and } n = N+1 \\ p^{(L-1)}(2N-n), & n = N+2, N+3, \ldots, 2N \\ p^{(L)}(0), & n = 2N+1 \\ 0, & \text{otherwise,} \end{cases} \tag{A.11}$$

and

$$e^{(1)}(n) = \begin{cases} p^{(L)}(0), & n = 0 \\ p^{(L-1)}(n-1), & n = 1, 2, \ldots, N-1 \\ -2p^{(L)}(n-1) & n = N \\ p^{(L-1)}(2N-1-n), & n = N+1, N+2, \ldots, 2N-1 \\ p^{(L)}(0), & n = 2N \\ 0, & \text{otherwise,} \end{cases} \tag{A.12}$$

respectively. To be able to generate the desired overall filter structure using accumulators, it is desirable to first express, for reasons to be explained later, the above impulse responses in the following form:

$$e^{(1)}(n) = h^{(1)}(n-1) + f^{(1)}(n) \text{ for } n = 0, 1, \ldots, \widetilde{N}. \tag{A.13}$$

Secondly, after some manipulations, it is favorable to express $h^{(1)}(n)$ and $f^{(1)}(n)$ in terms of the polynomials $p^{(L)}(n)$ and $p^{(L-1)}(n)$ as follows:

$$h^{(1)}(n) = \begin{cases} p^{(L-1)}(n), & n = 0, 1, \ldots, N-1 \\ p^{(L-1)}(2N-1-n), & n = N, N+1, \ldots, 2N-1 \\ 0, & \text{otherwise} \end{cases} \tag{A.14}$$

and

$$f^{(1)}(n) = \begin{cases} p^{(L)}(0), & n = 0 \text{ and } n = 2N+1 \\ -p^{(L)}(N), & n = N \text{ and } n = N+1 \\ 0, & \text{otherwise;} \end{cases} \tag{A.15}$$

as well as

$$h^{(1)}(n) = \begin{cases} p^{(L-1)}(n), & n = 0, 1, \ldots, N-1 \\ p^{(L-1)}(2N-2-n), & n = N, N+1, \ldots, 2N-2 \\ 0, & \text{otherwise} \end{cases} \tag{A.16}$$

and

$$f^{(1)}(n) = \begin{cases} p^{(L)}(0), & n = 0 \text{ and } \\ & n = 2N+1 \\ -p^{(L)}(N) - p^{(L)}(N-1), & n = N \\ 0, & \text{otherwise;} \end{cases} \tag{A.17}$$

respectively.
The above equations make it possible to express $E^{(1)}(z)$ as

$$E^{(1)}(z) = F^{(1)}(z) + z^{-1}H^{(1)}(z), \tag{A.18}$$

and the importance and the roles of $H^{(1)}(z)$ and $F^{(1)}(z)$ are discussed next. First, $H^{(1)}(z)$ is the polynomial part of the transfer function and can be expressed as

$$H^{(1)}(z) = \begin{cases} \displaystyle\sum_{n=0}^{N-1} h^{(1)}(n) \left[z^{-n} - z^{-(2N-1-n)}\right] \\ \displaystyle h^{(1)}(N-1) + \sum_{n=0}^{N-2} h^{(1)}(n) \left[z^{-n} + z^{-(2N-2-n)}\right] \end{cases} \tag{A.19}$$

for Types 3 and 4, respectively. Here, $h^{(1)}(n)$ is given in terms of the $(L-1)$th-order polynomial $p^{(L-1)}(n)$, given by (A.4)–(A.6), as follows:

$$h^{(1)}(n) = p^{(L-1)}(n) \text{ for } 0 \leq n \leq \widehat{N}_1, \tag{A.20}$$

where

$$\widehat{N}_1 = \ N - 1. \tag{A.21}$$

When comparing $H^{(1)}(z)$ and $H^{(0)}(z)$, the following differences are observed. First, $H^{(1)}(z)$ for the original Types 3 and 4 are of Types 2 and 1, respectively. Secondly, both the filter order and the order of the polynomial for the impulse responses that follow are reduced by one. This makes it possible, at the second step by means of the second accumulator, to reduce the order and the degree of $H^{(1)}(z)$ in a similar manner. This will be discussed after explaining the role of $F^{(1)}(z)$. The key idea is to give $F^{(1)}(z)$ for the Types 3 and 4 in such a form that $H^{(1)}(z)$ is of a form similar to that of the original transfer function $H^{(0)}(z)$. $F^{(1)}(z)$ is the non-polynomial part, i.e., it does not have a polynomial form and can be expressed as

$$F^{(1)}(z) = \begin{cases} p^{(L)}(0)[1 + z^{-(2N+1)}] - p^{(L)}(N)z^{-N}[1 + z^{-1}] \\ p^{(L)}(0)[1 + z^{-2N}] - [p^{(L)}(N) + p^{(L)}(N-1)]z^{-N} \end{cases} \tag{A.22}$$

for Types 3 and 4, respectively. It contains the remainder terms of $E^{(1)}(z)$ so that $H^{(1)}(z)$ can be expressed in the above form.

In order to further decrease the polynomial order, the second step is to add the second accumulator in the above step by expressing $H^{(1)}(z)$ as

$$H^{(1)}(z) = E^{(2)}(z)/(1 - z^{-1}), \tag{A.23}$$

where, after derivation similar to that performed for the first step, $E^{(2)}(z)$ can be written in the following form:

$$E^{(2)}(z) = (1 - z^{-1})H^{(1)}(z) \equiv F^{(2)}(z) + z^{-1}H^{(2)}(z). \tag{A.24}$$

Here,

$$H^{(2)}(z) = \begin{cases} \displaystyle\sum_{n=0}^{(N-1)-1} h^{(2)}(n)[z^{-n} - z^{-(2(N-1)-n)}] \\ \displaystyle\sum_{n=0}^{(N-1)-1} h^{(2)}(n)[z^{-n} - z^{-(2(N-1)-1-n)}]. \end{cases} \tag{A.25}$$

$H^{(2)}(z)$ is the polynomial part of the transfer function and $F^{(2)}(z)$ is the non-polynomial part and can be expressed as

$$F^{(2)}(z) = \begin{cases} p^{(L-1)}(0)[1 - z^{-2N}] & \text{for Type 3} \\ p^{(L-1)}(0)[1 - z^{-(2N-1)}] & \text{for Type 4.} \end{cases} \tag{A.26}$$

In $H^{(2)}(z)$, $h^{(2)}(n)$ is given in terms of the $(L-2)$th-order polynomial $p^{(L-2)}(n)$, as given by (A.4)–(A.6) as follows:

$$h^{(2)}(n) = p^{(L-2)}(n) = \sum_{r=0}^{L-2} a^{(L)}(r)n^r \text{ for } 0 \le n \le \widehat{N}_2 \tag{A.27}$$

where

$$\widehat{N}_2 = \ (N-1) - 1. \tag{A.28}$$

The above process is continued to generate $F^{(k)}(z)$ and $H^{(k)}(z)$ for $k = 3, 4, \ldots, L+1$. As a result of the overall procedure, the original transfer function can be expressed as

$$H^{(0)}(z) = [F^{(1)}(z) + z^{+1}H^{(1)}(z)]/[1 - z^{-1}] \tag{A.29}$$

$$H^{(1)}(z) = [F^{(2)}(z) + z^{-1}H^{(2)}(z)]/[1 - z^{-1}] \tag{A.30}$$

$$\vdots \qquad \vdots \qquad \vdots$$

$$H^{(L-1)}(z) = [F^{(L)}(z) + z^{-1}H^{(L)}(z)]/[1 - z^{-1}] \tag{A.31}$$

$$H^{(L)}(z) = F^{(L+1)}(z)/[1 - z^{-1}], \tag{A.32}$$

where $F^{(k)}(z)$ for $k = 1, 2, \ldots, L+1$ are the transfer functions to be implemented. The first L transfer functions for Types 3 and 4 are given by

$$F^{(2l+1)}(z) =$$

$$\begin{cases} p^{(L-2l)}(0)[1 + z^{-(2(N-l)+1)}] - p^{(L-2l)}(N-l)z^{-(N-l)}[1 + z^{-1}] \\ p^{(L-2l)}(0)[1 + z^{-2(N-l)}] - [p^{(L-2l)}(N-l) + p^{(L-2l)}((N-l)-1)]z^{-(N-l)} \end{cases} \tag{A.33}$$

for $l = 0, 1, \ldots, \lfloor (L+1)/2 \rfloor - 1$, respectively, and by

$$F^{(2l+2)}(z) = \begin{cases} p^{(L-2l-1)}(0)[1 - z^{-2(N-l)}] & \text{for Type 3} \\ p^{(L-2l-1)}(0)[1 - z^{-(2(N-l)-1)}], & \text{for Type 4} \end{cases} \tag{A.34}$$

for $l = 0, 1, \ldots, \lfloor L/2 \rfloor - 1$.

Alternatively, these transfer functions, for Types 3 and 4, can be expressed as

$$F^{(2l+1)}(z) =$$

$$\begin{cases} \alpha_{2l+1}[1 + z^{-(2(N-l)+1)}] + \beta_{2l+1} z^{-(N-l)}[1 + z^{-1}] & \text{for Type 3} \\ \alpha_{2l+1}[1 + z^{-2(N-l)}] + \beta_{2l+1} z^{-(N-l)} & \text{for Type 4} \end{cases} \tag{A.35}$$

for $l = 0, 1, \ldots, \lfloor (L+1)/2 \rfloor - 1$, respectively, and as

$$F^{(2l+2)}(z) = \begin{cases} \alpha_{2l+2}[1 - z^{-2(N-l)}] & \text{for Type 3} \\ \alpha_{2l+2}[1 - z^{-(2(N-l)-1)}] & \text{for Type 4} \end{cases} \tag{A.36}$$

for $l = 0, 1, \ldots, \lfloor L/2 \rfloor - 1$. Here, α_ks, for $k = 1, 2, \ldots, L$, are related to the corresponding polynomials, $p^{(L-1+k)}(n)$s, as given by (A.4), (A.5) and (A.6), through

$$\alpha_k = p^{(L+1-k)}(0), \; k = 1, 2, \ldots, L, \tag{A.37}$$

whereas non-zero values of β_k's for Types 3 and 4 are given by

$$\beta_{2k-1} = -p^{(L-2(k-1))}(N-k), \tag{A.38}$$

for $k = 1, 2, \ldots, \lfloor (L+1)/2 \rfloor$, and

$$\beta_{2k-1} = -p^{(L-2(k-1))}(N-k) - p^{(L-2(k-1))}((N-k)-1), \tag{A.39}$$

for $k = 1, 2, \ldots, \lfloor (L+1)/2 \rfloor$, respectively.

The remaining transfer function $F^{(L+1)}(z)$ can be expressed as

$$F^{(L+1)}(z) = \begin{cases} p^{(0)}(0)[1 - z^{-(N-L/2)}(1 + z^{-1}) + z^{-2(N-L/2)+1}] & \text{for Type 3} \\ p^{(0)}(0)[1 - 2z^{-(N-L/2)} + z^{-2(N-L/2)}] & \text{for Type 4} \end{cases} \tag{A.40}$$

and

$$F^{(L+1)}(z) = \begin{cases} p^{(0)}(0)[1 - z^{-(2N-L+1)}] & \text{for Type 3} \\ p^{(0)}(0)[1 - z^{-(2N-L)}] & \text{for Type 4} \end{cases} \tag{A.41}$$

for L even and odd, respectively.

These transfer functions can be rewritten in the form

$$F^{(L+1)}(z) = \begin{cases} \alpha_{L+1}[1 + z^{-2(N-L/2)+1}] + \beta_{L+1} z^{-(N-L/2)}(1 + z^{-1}) \\ \alpha_{L+1}[1 + z^{-2(N-L/2)}] + \beta_{L+1} z^{-(N-L/2)} \end{cases} \tag{A.42}$$

and

$$F^{(L+1)}(z) = \begin{cases} \alpha_{L+1}[1 - z^{-(2N-L+1)}] & \text{for Type 3} \\ \alpha_{L+1}[1 - z^{-(2N-L)}] & \text{for Type 4} \end{cases} \tag{A.43}$$

for L even and odd, respectively. Here, for even L, $\alpha_{L+1} = p^{(0)}(0)$, and

$$\beta_{L+1} = -p^{(0)}(0) \text{ and } \beta_{L+1} = -2p^{(0)}(0)$$

for Types 3 and 4, respectively.

Based on the above, the overall structures for Types 3 and 4 can be developed as in Publication-II and the implementation structures are shown in Publication-II, where delay terms of the structures are given as $T_{m-1} = N_m - N_{m-1}$ for $m = 2, 3, \ldots, M$, $T_{M1} = N - N_M$ and $T_{M2} = N - N_M$ (see Publication-II).

Bibliography

[1] J. W. Adams and A. N. Wilson Jr., “A new approach to FIR digital filters with fewer multipliers and reduced sensitivity,” *IEEE Trans. Circuits and Systems,*, vol. 30, no. 5, pp. 277–283, 1983.

[2] A. Antoniou, *Digital Signal Processing: Signal, Systems and Filters.* McGraw-Hill, 2005.

[3] P. Arian, “Computationally efficient decimators, interpolators, and narrow transition-band linear-phase finite impulse response (FIR) filters,” *Dr. Tech. Thesis*, pp. 1–229, 2007.

[4] 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 Region 10 Conference TENCON 2004*, vol. 1, 2004, pp. 60–63.

[5] P. Arian, T. Saramäki, and A. T. Fam, “A new class of linear-phase FIR filters based on switching and resetting of IIR filters,” in *Proc. Norchip Conference*, 2004, pp. 167–170.

[6] M. G. Bellanger, “Improved design of long FIR filters using the frequency masking technique,” in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 3, 1996, pp. 1272–1275.

[7] G. Boudreaux and T. Parks, “Thinning digital filters: A piecewise-exponential approximation approach,” *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 31, no. 1, pp. 105–113, 1983.

[8] T. G. Campbell and T. Saramäki, “Recursive linear-phase FIR filter structures with a piecewise-polynomial impulse response,” in *Proc. 6th Int. Symp. Networks, Syst. Signal Processing, Zagreb, Yuogoslavia*, 1989, pp. 16–19.

[9] S. Chand and H. Om, "Phase estimation in video transmission using Hilbert transform," in *Proc. IEEE Int. Conf. Signal Processing, Communications, Networking*, 2007, pp. 157–161.

[10] T. Chien-Cheng and P. Soo-Chang, "Design of discrete-time fractional Hilbert transformer," in *Proc. IEEE Int. Symp. Circuits and Syst., Geneva, 2000*, vol. 5, 2000, pp. 525–528.

[11] S. Chu and C. S. Burrus, "Efficient recursive realizations of FIR filters, part I: The filter structures," *Circuits, Syst. and Signal processing*, vol. 3, no. 1, pp. 3–20, 1984.

[12] S. Chu and C. S. Burrus, "Efficient recursive realizations of FIR filters, part II: Design and applications," *Circuits, Systems and Signal processing, Birkhauser Boston*, vol. 3, no. 1, pp. 21–57, 1984.

[13] V. Cizek, "Discrete Hilbert transform," *IEEE Trans. Audio and Electroacoustics*, vol. 18, no. 4, pp. 340–343, 1970.

[14] R. Crochiere and L. Rabiner, "Optimum FIR digital filter implementations for decimation, interpolation, and narrow-band filtering," *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. 23, no. 5, pp. 444–456, 1975.

[15] S. C. Dutta Roy and A. Agrawal, "Digital low-pass filtering using the discrete Hilbert transform," *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. 26, no. 5, pp. 465–467, 1978.

[16] A. T. Fam, "MFIR filters: Properties and applications," *IEEE trans. Acoustics, Speech, and Signal Processing*, vol. 29, no. 6, pp. 1128–1136, 1981.

[17] A. T. Fam, "FIR filters that approach IIR filters in their computationally complexity." in *Proc. Twenty-first Asilomar Conference on Signals, Systems and Computers*, Nov. 1987, pp. 28–30.

[18] N. J. Fliege, *Multirate Digital Signal Processing.* Chichester: John Wiley and Sons, 1994.

[19] T. W. Fox, "Bandlimited FIR differentiators with high stopband attenuation and low stopband energy," in *Proc. IEEE Pacific Rim Conf. Communications, Computers and signal Processing, 2005.*, 2005, pp. 546–549.

[20] O. Gustafsson, H. Johansson, and L. Wanhammar, “Single frequency-response masking FIR filters,” *Journal of Circuits, Systems, and Computers*, vol. 12, no. 5, pp. 601–630, 2003.

[21] O. Hermann, L. R. Rabiner, and D. S. K. Chan, “Practical design rules for optimum finite impulse response low-pass digital filters,” *Bell Syst. Tech. J.*, vol. 52, no. 6, pp. 769–799, July-Aug. 1973.

[22] S. Hong, S. S. Chin, S. Kim, and W. Hwang, “Multiplier architecture power consumption characterization for low-power DSP applications.” in *Proc. Electronics, Circuits and Systems*, Nov 2002, pp. 741–744.

[23] K. Ichige, M. Iwaki, and R. Ishii, “A new estimation formula for minimum filter length of optimum FIR digital filters,” in *Proc. Fourth Int. Conf. Signal Processing 1998.*, 1998, pp. 89–92.

[24] K. Ichige, M. Iwaki, and R. Ishii, “Accurate estimation of minimum filter length for optimum FIR digital filters,” *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing*, vol. 47, no. 10, pp. 1008–1016, 2000.

[25] K. Ichige, M. Kamada, and R. Ishii, “A simple scheme of decomposition and reconstruction of continuous-time signals by B-splines,” in *Proc. 1998 IEEE Int. Symp. Circuits and Syst., 1998*, vol. 5, 1998, pp. 453–456.

[26] Z. Jing and A. T. Fam, “A new structure for narrow transition band, lowpass digital filter design,” *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. ASSP-32, no. 2, pp. 362–370, 1984.

[27] H. Johansson, “Two classes of frequency-response masking linear-phase FIR filters for interpolation and decimation,” *Circuits, Systems and Signal processing,Birkhäauser, Boston*, vol. 25, no. 2, pp. 175–200, 2006.

[28] H. Johansson and P. Löwenborg, “On linear-phase FIR filters with variable bandwidth,” *IEEE Trans. Circuits and Systems II: Express Briefs*, vol. 51, no. 4, pp. 181–184, 2004.

[29] H. Johansson and L. Wanhammar, “High-speed recursive digital filters based on the frequency-response masking approach,” *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing*, vol. 47, no. 1, pp. 48–61, 2000.

[30] J. F. Kaiser, "Nonrecursive digital filter design using Io-sinh window function," in *Proc. IEEE Int. Symp. Circuits Syst.*, Apr. 1974, pp. 20–23.

[31] J. F. Kaiser and R. W. Hamming., "Sharpening the response of a symmetric nonrecursive filter by multiple use of the same filter," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 25, no. 5, pp. 415–422, 1977.

[32] S. C. Kak, "The discrete Hilbert transform," *Proc. of the IEEE*, vol. 58, no. 4, pp. 585–586, 1970.

[33] R. Kavanagh, "Signal processing techniques for improved digital tachometry," *Proc IEEE Int. Symp. Industrial Electronics*, vol. 2, pp. 511–517, 2002.

[34] I. R. Khan and M. Okuda, "Finite-impulse-response digital differentiators for midband frequencies based on maximal linearity constraints," *IEEE Trans. Circuits and Systems II: Express Briefs*, vol. 54, no. 3, pp. 242–246, 2007.

[35] I. R. Khan and M. Okuda, "Narrow transition band FIR Hilbert transformers with flat magnitude response," *IEEE Signal Processing Letters*, vol. 14, no. 9, pp. 613–616, 2007.

[36] I. R. Khan, M. Okuda, and R. Ohba, "Digital differentiators for narrow band applications in the lower to midband range," in *Proc. IEEE Int. Symp. Circuits and Systems, 2005.ISCAS 2005.*, vol. 4, 2005, pp. 3721–3724.

[37] I. Kollar, R. Pintelon, and J. Schoukens, "Optimal FIR and IIR Hilbert transformer design via LS and minimax fitting," *IEEE Trans. Instrumentation and Measurement*, vol. 39, no. 6, pp. 847–852, 1990.

[38] J. Le Bihan, "Coefficients of FIR digital differentiators and Hilbert transformers for midband frequencies," *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing*, vol. 43, no. 3, pp. 272–274, 1996.

[39] R. Lehto, *Design and Implementation of FIR Filters with a Piecewise-Polynomial Impulse response, Master of Science Thesis.* Tampere University of Technology, 2004.

[40] R. Lehto, T. Saramäki, and O. Vainio, "Synthesis of narrowband linear-phase FIR filters with a piecewise-polynomial impulse response," in *Proc. IEEE Int. Symp. Circuits and Systems, 2005.*, vol. 3, 2005, pp. 2012–2015.

[41] R. Lehto, T. Saramäki, and O. Vainio, "Synthesis of wideband linear-phase FIR filters with a piecewise-polynomial-sinusoidal impulse response," in *Proc. IEEE Int. Symp. Circuits and Systems*, 2007, pp. 2052–2055.

[42] R. Lehto, T. Saramäki, and O. Vainio, "Hilbert transformers with a piecewise-polynomial-sinusoidal impulse response," in *Proc. IEEE Int. Symp. Circuits and Systems*, 2008, pp. 2450–2453.

[43] C. M. Leung and W.-S. Lu, "Detection of edges of noisy images by 1-d and 2-d linear-phase digital filters," *Proc. of Pac. Rim.*, pp. 228 – 231, March 1993.

[44] Y. Lian, "Complexity reduction for FRM-based FIR filters using prefilter-equalizer technique," *Circuits, Systems and Signal Processing*, vol. 22, no. 2, pp. 137–155, 2003.

[45] Y. Lian, "A modified frequency-response masking structure for high-speed FPGA implementation of sharp FIR filters," *Journal of Circuits, Systems, and Computers*, vol. 12, no. 5, pp. 643–654, 2003.

[46] Y. Lian and W.-S. Lu, "Guest editorial: Computationally efficient digital filters: Design techniques and applications," *Circuits, Systems and Signal Processing*, vol. 25, no. 2, pp. 125–130, 2006.

[47] Y. C. Lim, "Frequency-response masking approach for the synthesis of sharp linear phase digital filters," *IEEE Trans. Circuits and Systems*, vol. 33, no. 4, pp. 357–364, 1986.

[48] Y. C. Lim and Y. Lian, "The optimum design of one- and two-dimensional FIR filters using the frequency response masking technique," *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing*, vol. 40, no. 2, pp. 88–95, 1993.

[49] Y. C. Lim and Y. Lian, "Frequency-response masking approach for digital filter design: Complexity reduction via masking filter factorization," *IEEE Trans. Circuits and Systems II: Analog and Digital Signal process*, vol. 42, no. 8, pp. 518–525, Aug 1994.

[50] Y. C. Lim and Y. J. Yu, "Synthesis of very sharp Hilbert transformer using the frequency-response masking technique," *IEEE Trans. Signal Processing*, vol. 53, no. 11, pp. 2595–2597, 2005.

[51] Y. C. Lim, Y. J. Yu, and T. Saramäki, "Optimum masking levels and coefficient sparseness for Hilbert transformers and half-band filters designed using the frequency-response masking technique," *IEEE Trans. Circuits and Systems I: Regular Papers*, vol. 52, no. 11, pp. 2444–2453, 2005.

[52] Y. C. Lim, "Efficient special purpose linear programming for FIR filter design," *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. 31, no. 4, pp. 963–968, 1983.

[53] P. Löwenborg and H. Johansson, "Minimax design of adjustable-bandwidth linear-phase FIR filters," *IEEE Trans. Circuits and Systems I: Regular Papers*, vol. 53, no. 2, pp. 431–439, 2006.

[54] J. H. McClellan., T. W. Parks., and L. R. Rabiner, "A computer program for designing optimum FIR linear phase digital filters," *IEEE Trans. Audio and Electroacoustics*, vol. AU-21, no. 6, pp. 506–526, 1973.

[55] S. K. Mitra, *Digital Signal Processing: A Computer Based Approach, third edition*. McGraw-Hill, 2006.

[56] S. K. Mitra, A. Mahalonobis, and T. Saramäki, "A generalized structural subband decomposition of FIR filters and its application in efficient fir filter design and implementation," *IEEE Trans Circuits and Syst. II: Analog and Digital Signal Processing*, vol. 40, no. 6, pp. 363–374, 1993.

[57] G. Mollova, "Design of FIR LS Hilbert transformers through fullband differentiators," in *Conf. Record of the Forty-First Asilomar Conf. Signals, Systems and Computers, 2007*, 2007, pp. 1121–1125.

[58] D. G. Munoz, O. Gustafsson, and L. Wanhammar, "Evolution of filter order equations for linear-phase FIR filters using gene expression programming," in *Proc. Radio Vetenskap och Kommunikation*, June 2005, pp. 679–681.

[59] S. Nakamura and S. K. Mitra, "Design of FIR digital filters using tapped cascaded FIR subfilters," in *Circuits, Systems and Signal Processing*, vol. 1, 1982, pp. 43–56.

[60] S. L. Netto, L. C. R. de Barcellos, and P. S. R. Diniz, “Efficient design of narrowband cosine-modulated filter banks using a two-stage frequency-response masking approach,” *Journal of Circuits, Systems, and Computers*, vol. 12, no. 5, pp. 631–642, 2003.

[61] Y. Neuvo, C.-Y. Dong, and S. K. Mitra, “Interpolated finite impulse response filters,” *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 32, no. 3, pp. 563–570, 1984.

[62] Y. Neuvo, G. Rajan., and S. K. Mitra, “Design of narrow-band FIR bandpass digital filters with reduced arithmetic complexity,” *IEEE Trans. Circuits and Systems*, vol. 34, no. 4, pp. 409–419, 1987.

[63] K. K. Parhi, *VLSI Digital Signal Processing Systems: Design and Implementation*. N.Y.: John Wiley and Sons, 1999.

[64] S. C. Pei and P. H. Wang, “Analytical design of digital nonrecursive maximally flat fractional Hilbert transformer,” in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 3, 1999, pp. 175–178.

[65] S. C. Pei and P. H. Wang, “Closed-form design of maximally flat FIR Hilbert transformers, differentiators, and fractional delayers by power series expansion,” *IEEE Trans. Circuits and Systems I: Fundamental Theory and Applications*, vol. 48, no. 4, pp. 389–398, 2001.

[66] S. C. Pei and M. H. Yeh, “Discrete fractional Hilbert transform,” *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing*, vol. 47, no. 11, pp. 1307–1311, 2000.

[67] J. M. Rabaey, A.Chandrakasan, and B. Nikolic, *Digital Integrated Circuits: A Design Perspective*. Upper saddle River, New Jersey: Pearson Education International, 2003.

[68] L. R. Rabiner and R. E. Crochiere, “A novel implementation for narrow-band FIR digital filters,” *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. 23, no. 5, pp. 457–464, 1975.

[69] G. Rajan, Y. Neuvo, and S. K. Mitra, “On the design of sharp cutoff wide-band FIR filters with reduced arithmetic complexity,” *IEEE Trans. Circuits and Syst*, vol. 35, no. 11, pp. 1447–1454, 1988.

[70] T. A. Ramstad and T. Saramäki, “Efficient multirate realization for narrow transition-band FIR filters,” in *Proc. IEEE Symp. Circuits and Systems, 1988.*, vol. 3, 1988, pp. 2019–2022.

[71] T. A. Ramstad and T. Saramaki, "Multistage, multirate fir filter structures for narrow transition-band filters," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 3, May 1990, pp. 2017–2021.

[72] L. Rosenbaum, P. Lowenborg., and H. Johansson, "Cosine and sine modulated FIR filter banks utilizing the frequency-response masking approach," in *Proc. 2003 IEEE Int. Symp. Circuits and Systems, 2003.*, vol. 3, 2003, pp. 882–885.

[73] S. Samadi, M. O. Ahmad, and M. N. S. Swamy, "Exact fractional-order differentiators for polynomial signals," *IEEE Signal Processing Letters*, vol. 11, no. 6, pp. 529–532, 2004.

[74] S. Samadi, Y. Igarashi, and H. Iwakura, "Design and multiplierless realization of maximally flat FIR digital Hilbert transformers," *IEEE Trans. Signal Processing*, vol. 47, no. 7, pp. 1946–1953, 1999.

[75] S. Samadi, H. Iwakura, and A. Nishihara, "Multiplierless and hierarchical structures for maximally flat half-band FIR filters," *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing*, vol. 46, no. 9, pp. 1225–1230, 1999.

[76] T. Saramäki, "Narrowband linear-phase FIR filters requiring a small number of multipliers," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 7, 1982, pp. 278–281.

[77] T. Saramäki, "Design of FIR filters as a tapped cascaded interconnection of identical subfilters," *IEEE Trans. Circuits and Systems*, vol. 34, no. 9, pp. 1011–1029, 1987.

[78] T. Saramäki, "Finite impulse response filter design," in *Handbook for Digital Signal Processing*, S. K. Mitra and J. F. Kaiser, Eds. New York: John Wiley and Sons, 1993, ch. 4, pp. 155–277.

[79] T. Saramäki, *Design of Computationally Efficient FIR Filters Using Periodic Subfilters as Building Blocks*, ser. The Circuits and Filters Handbook. CRC Press, Inc., 1995, pp. 2578–2601.

[80] T. Saramäki, *Design of Computationally Efficient FIR Filters Using Periodic Subfilters as Building Blocks*, ser. The Circuits and Filters Handbook. CRC Press, Inc., 2003, pp. 2654–2677.

[81] 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 and Systems*, vol. 4, 1990, pp. 3271–3274.

[82] T. Saramäki and Y. C. Lim, "Use of the remez algorithm for designing FIR filters utilizing the frequency-response masking approach," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 3, 1999, pp. 449–455.

[83] T. Saramäki and Y. C. Lim, "Use of the Remez-algorithm for designing FIR filters utilizing the frequency-response masking approach," *Circuits, Systems and Signal Processing*, vol. 22, no. 2, pp. 77–97, 2003.

[84] T. Saramäki and S. K. Mitra, "Multiple branch FIR filters for sampling rate conversion," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 2, 1992, pp. 1007–1010.

[85] T. Saramäki and S. K. Mitra, "Design and implementation of narrowband linear-phase FIR filters with piecewise polynomial impulse response," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 3, 1999, pp. 456–461.

[86] T. Saramäki, T. Neuvo, and S. K. Mitra, "Design of computationally efficient interpolated FIR filters," *IEEE Trans. Circuits and Systems*, vol. 35, no. 1, pp. 70–88, 1988.

[87] T. Saramäki and O. Vainio, "Structures for generating polynomial responses," in *Proc. Midwest Symp. Circuits and Systems*, vol. 2, 1994, pp. 1315–1318.

[88] T. Saramäki, J. Yli-Kaakinen, and H. Johansson, "Optimization of frequency response masking based FIR filters," *Journal of Circuits, Syst. Computers*, vol. 12, no. 5, pp. 563–591, Oct. 2003.

[89] H. W. Schussler and P. Steffen, "Halfband filters and Hilbert transformers," *Circuits, Systems and Signal Processing*, vol. 17, no. 2, pp. 137–164, 1998.

[90] I. W. Selesnick, "Narrowband lowpass digital differentiator design," in *Conf. Record of the Thirty-Sixth Asilomar Conf. Signals, Systems and Computers*, vol. 1, 2002, pp. 360–364.

[91] D. Shi, P. J. Unsworth, and R. X. Gao, "Sensorless speed measurement of induction motor using Hilbert transform and interpolated fast fourier transform," *IEEE Trans. Instrumentation and Measurement*, vol. 55, no. 1, pp. 290–299, 2006.

[92] M. Smith and D. Farden, "Thinning the impulse response of FIR digital filters," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 6, 1981, pp. 240–242.

[93] P. Soo-Chang and Y. Min-Hung, "Discrete fractional Hilbert transform;," *IEEE Trans. Circuits and Syst. II: Analog and Digital Signal Processing,*, vol. 47, no. 11, pp. 1307–1311, 2000.

[94] C. C. Tseng, "Design of fractional order digital FIR differentiators," *IEEE Signal Processing Letters*, vol. 8, no. 3, pp. 77–79, 2001.

[95] C. C. Tseng, "Digital differentiator design using fractional delay filter and limit computation," *IEEE Trans. Circuits and Systems I: Regular Papers*, vol. 52, no. 10, pp. 2248–2259, 2005.

[96] C. C. Tseng and S. C. Pei, "Design of discrete-time fractional Hilbert transformer," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 5, 2000, pp. 525–528.

[97] S. Usui and I. Amidror, "Digital low-pass differentiation for biological signal processing," *IEEE Trans. Biomedical Engineering*, vol. BME-29, no. 10, pp. 686–693, 1982.

[98] O. Vainio, R. Lehto, and T. Saramäki, "Fractional-order FIR differentiators with optimum noise attenuation," in *Proc. IEEE Instrumentation and Measurement Technology*, 2007, p. 4p.

[99] O. Vainio, M. Renfors, and T. Saramäki, "Recursive implementation of FIR differentiators with optimum noise attenuation," *IEEE Trans. Instrumentation and Measurement*, vol. 46, no. 5, pp. 1202–1207, 1997.

[100] L. Wanhammar, *DSP Integrated Cicuits.* New York:Academic Press, 1999.

[101] R. Yang, Y. C. Lim, and S. R. Parker, "Design of sharp linear-phase FIR bandstop filters using the frequency-response-masking technique," *Circuits, Systems and Signal Processing*, vol. 17, no. 1, pp. 1–27, 1998.

[102] J. Yu and Y. Lian, "Design equations for jointly optimized frequency-response masking filters," *Circuits, Systems and Signal Processing*, vol. 26, no. 1, pp. 27–42, 2007.

[103] L. Zhou, W. Pei, P. Xi, and Z. He, "Frequency-response masking approach for design of intermediate frequency filters in CDMA and wide-band GSM modules," in *Proc. of IEEE Asia Pacific Conference Circuits and Syst.*, 2006, pp. 736–739.

[104] D. G. Zill and M. R. Gullen, *Advanced Engineeering Mathematics.* Jones abd Bartlett, 2000.

Publications

Publication-I

R. Lehto and T. Saramäki and O. Vainio, Formulas to generate efficient piecewise-polynomial implementations of narrowband Linear-Phase FIR Filters, *Proc. 2006 IEEE Int. Symp. on Circuits and Systems*, pp. 2513–2516, May, 2006, Kos, Greece.

Formulas to Generate Efficient Piecewise-Polynomial Implementations of Narrowband Linear-Phase FIR Filters

Raija Lehto and Tapio Saramäki
Institute of Signal Processing
Tampere University of Technology
Tampere, Finland
Email: raija.lehto@tut.fi, tapio.saramaki@tut.fi

Olli Vainio
Institute of Digital and Computer Systems
Tampere University of Technology
Tampere, Finland
Email: olli.vainio@tut.fi

Abstract— **Two computationally efficient structures to design and implement linear-phase narrowband FIR filters with a symmetrical piecewise-polynomial impulse response have been proposed by Saramäki and Mitra. The efficiency of these structures is based on implementing the overall transfer function as a parallel connection of a few branches of the form $G_l(z^L)F_l(z)$, where each $F_l(z)$ requires no real multipliers. These structures have been generated in an ad-hoc manner. This paper introduces straightforward approaches to generate such $F_l(z)$s for both structures.**

I. Introduction

The main drawback of conventional direct-form FIR filters is that they require a very large number of coefficients when implementing narrowband FIR filters. This is because the order of these filters is roughly inversely proportional to the transition bandwidth, which is not the case with IIR filter designs. This fact makes the implementation of direct-form FIR filters with a narrow transition band very costly compared to the corresponding IIR filters, in terms of their arithmetic complexity, i.e., the required number of adders, multipliers, and delay elements. However, the advantages of FIR filters dominate over the difficulties in such filters.

Recursive implementation structures for FIR filters and the corresponding design techniques have been proposed by several authors [1–6]. Saramäki and Vainio [4] have introduced efficient recursive structures to generate arbitrary polynomial responses. Saramäki and Mitra [5] have presented a straightforward approach for two structures to implement narrowband linear-phase FIR filters with a symmetrical piecewise polynomial impulse response, related to Saramäki's and Vainio's filter structures. The start-up transfer function in the Saramäki-Mitra approach is given by

$$H(z) = \sum_{l=0}^{R} G_l(z^L)F_l(z), \tag{1}$$

where

$$G_l(z) = \sum_{n=0}^{M-1} g_l(n)z^{-n} \text{ and } F_l(z) = \sum_{n=0}^{L-1} f_l(n)z^{-n} \tag{2}$$

with

$$f_l(n) = \left[\frac{n-(L-1)/2}{(L-1)/2}\right]^l \text{ for } n = 0,1,\ldots,L-1 \tag{3}$$

and for even [odd] $g_l(M-1-n) = g_l(n)[g_l(M-1-n) = -g_l(n)]$ for $n = 0,1,\cdots,M-1$. The resulting transfer function is of order $N = ML-1$ and the corresponding impulse response has an even symmetry. Most importantly, the above transfer function makes it possible to devide the impulse response into M blocks of L samples in a such manner that in each block this impulse response a Rth degree polynomial. As shown in [5] after properly selecting $R < L-1$ and M and optimizing the coefficients of $G_l(z)$s results in narrowband cases in filters that closely approximate the minimum-order minimax FIR filter designs to meet the given criteria. When taking into account only the symmetrical or antisymmetrical coefficients of the $G_l(z)$s, this fact yields for small values of R filters with a significantly reduced number of multipliers.

Efficient implementations cannot be achieved directly by using the above $F_l(z)$s as basis fumctions. In order to achieve efficient implementations the $f_l(n)$s and $g_l(n)$s have to be modified. For this purpose, the original transfer function has been rewritten as

$$H(z) = \sum_{l=0}^{R} \tilde{G}_l(z^L)\tilde{F}_l(z), \tag{4}$$

where

$$\tilde{G}_l(z) = \sum_{n=0}^{M-1} \tilde{g}_l(n)z^{-n} \text{ and } \tilde{F}_l(z) = \sum_{n=0}^{L-1} \tilde{f}_l(n)z^{-n} \tag{5}$$

with

$$\begin{aligned}
\tilde{f}_0(n) &= f_0(n)\\
\tilde{f}_1(n) &= \tilde{c}_{11}f_1(n)\\
\tilde{f}_2(n) &= \tilde{c}_{20}f_0(n) + \tilde{c}_{22}f_2(n)\\
\tilde{f}_3(n) &= \tilde{c}_{31}f_1(n) + \tilde{c}_{33}f_3(n)\\
\tilde{f}_4(n) &= \tilde{c}_{40}f_0(n) + \tilde{c}_{42}f_2(n) + \tilde{c}_{44}f_4(n)\\
\tilde{f}_5(n) &= \tilde{c}_{51}f_1(n) + \tilde{c}_{53}f_3(n) + \tilde{c}_{55}f_5(n).
\end{aligned} \tag{6}$$

and

$$\begin{aligned}
&\tilde{g}_0(m) = g_0(m) - \tfrac{\tilde{c}_{20}g_2(m)}{\tilde{c}_{22}} + [\tfrac{g_4(m)}{\tilde{c}_{44}}][\tfrac{\tilde{c}_{20}\tilde{c}_{42}}{\tilde{c}_{22}} - \tilde{c}_{40}]\\
&\tilde{g}_1(m) = g_1(m)/\tilde{c}_{11} - \tilde{c}_{31}g_3(m)/[\tilde{c}_{11}\tilde{c}_{33}]-\\
&-\tilde{c}_{51}g_5(m)/[\tilde{c}_{11}\tilde{c}_{55}]\\
&\tilde{g}_2(m) = g_2(m)/\tilde{c}_{22} - \tilde{c}_{42}g_4(m)/[\tilde{c}_{22}\tilde{c}_{44}]\\
&\tilde{g}_3(m) = g_3(m)/\tilde{c}_{33} - \tilde{c}_{53}g_5(m)/[\tilde{c}_{33}\tilde{c}_{55}]\\
&\tilde{g}_4(m) = g_4(m)/\tilde{c}_{44}\\
&\tilde{g}_5(m) = g_5(m)/\tilde{c}_{55}.
\end{aligned} \tag{7}$$

By rewriting the original transfer function in the above manner, the overall transfer function remains the same. The main goal has been to to select the $\tilde{c}_{lk}$s so that the resulting $\tilde{F}_l(z)$s can be implemented by using accumulators and proper feedforward loops containing a few integer-valued coefficients and the corresponding structures satisfy the following two properties: First, provided that fixed-point modulo arithmetic (e.g. one's or two's complement arithmetic) and worst-case scaling is used, the outputs are correct even though internal overflows may occur in the accumulators. Secondly, there is no need for resetting and the effect of temporary miscalculations vanishes from the outputs in a finite time. In [5], two structures have been generated, based on the above principle, to implement the new basis functions $\tilde{F}_l(z)$s up to degree $R = 4$. However. the derivations have been performed in ad-hoc manners.

Based on the above approaches [4-5] this paper introduces mathematical formulas to generate the above-mentioned structures up to degree $R = 5$.

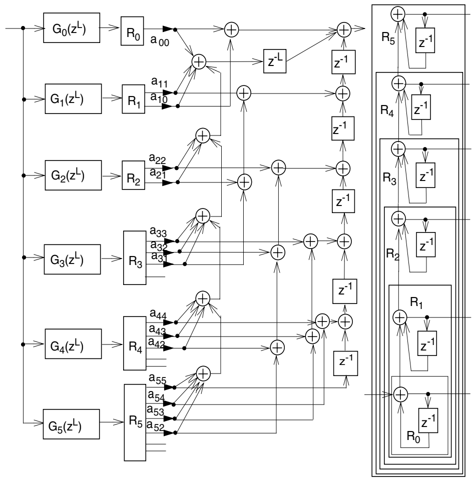


Fig. 1. Case A implementation structure [5].

II. Mathematical Formulas for the Modified Basis Functions

Proper mathematical formulas for the generation of the modified basis functions for the two efficient implementation structures proposed in [5] are introduced in this section, whereas the following section describes the algorithms for practical development of these structures. These modified basis functions are formulated so that they guarantee that the coefficients are in the feedforward loops integer-valued and their number is minimized. The two structures proposed in [5] are shown in Figures 1 and 2 and are referred to as *Case A* implementation and *Case B* implementation, respectively. The main difference compared to the structures proposed in [5] is that both the structures of Figure 1 and Figure 2 contain an additional term. This makes it possible for us to generate these structure up to degree $R = 5$, instead of $R = 4$ that has been used in [5]. For both structures, the main goal is to construct modified basis functions $\tilde{F}_l(z)$ in such a manner that, first, all the a_{lk}s become integer-valued and, secondly, the number of these multipliers is minimized.

In order to achieve the above-mentioned goals for *Case A* implementation, consider the structure of Figure 3, where the roles of the a_{lk}s are the same as in Figure 1. In Figure 3, the $\tilde{f}_l(n)$s for $l = 0, 1, \ldots, 5$ can be written as

$$\tilde{f}_l(n) = \sum_{k=0}^{l} a_{lk}\widehat{q}_k(n) \text{ for } n = 0, 1, \ldots, L-1, \tag{8}$$

where

$$\widehat{q}_0(n) = 1 \tag{9}$$

and

$$\widehat{q}_R(n) = \frac{1}{R!}\prod_{r=1}^{R}(n - r + 1) \text{ for } R = 1, 2, \ldots, 5. \tag{10}$$

As has been discussed earlier, the ultimate goal is to find in the above equations the a_{lk}s in such a manner that the following three properties are satisfied. First, the impulse responses of the resulting modified basis functions $\tilde{F}_l(z)$ for $l = 0, 1, \ldots, 5$ can be expressed as in the forms as given in (6). Secondly, all the a_{lk}s are integer-valued. Thirdly, the number of such a_{lk}s is minimized.

For *Case B* implementation, the generation of the desired structure of Figure 2 is based on the structure of Figure 4. This is because this figure is in practicle the transposed structure of Figure 4. In this case, the the $\tilde{f}_l(n)$s for $l = 0, 1, \ldots, 5$ can be written as

$$\tilde{f}_l(n) = \sum_{k=0}^{l} a_{lk}q_k(n) \text{ for } n = 0, 1, \ldots, L-1, \tag{11}$$

where

$$q_0(n) = 1 \tag{12}$$

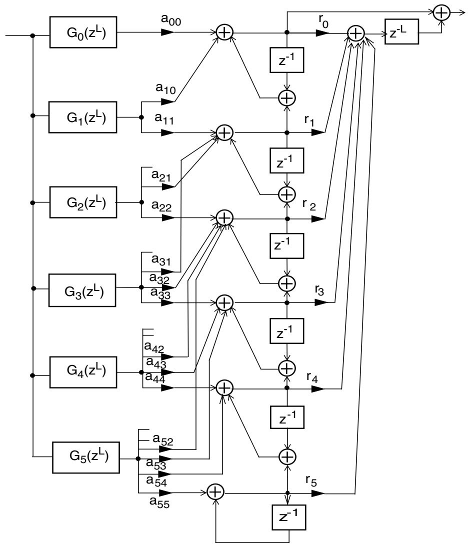


Fig. 2. Case B implementation structure of degree 5 with alternatives $3.II$, $4.II$, $5.I$ in Table III and IV and with $r_0 = 1$ and $r_k = \Pi_{l=1}^{k}(L - 1 + l)$ [5].

and

$$q_R(n) = \frac{1}{R!}\prod_{r=1}^{R}(n + r) \text{ for } R = 1, 2, \ldots, 5. \tag{13}$$

The main goal is to determine in the above equations the a_{lk}s to meet the same three properties as stated earlier for *Case A* implementation.

III. Algorithms to Determine the Desired a_{lk}s

This section describes the algorithms for generation of the a_{lk}s to meet the above-mentioned three properties for *Case A* and *Case B* implementations as well as the corresponding $\tilde{c}_{kl}$s in the forms as given in (6). Developing these algorithms has been based on reasonings that are too long to explain here. All the details behind these algorithms will be explained in more detail in a full-length article to published later. This section concentrates only on the basic steps of these algorithms.

The basis functions can be expressed as (3) [5] except that now the denominator, $(\frac{L}{2} - \frac{1}{2})^l$, is left out at this point in order to simplify the calculations:

$$\widehat{f}_l(n) = (n - \frac{L}{2} + \frac{1}{2})^l,\ n = 0, 1, \ldots, L-1, \tag{14}$$

where L is the length of the impulse response blocks. The denominator part in (3) has been taken into account in formulas for the altered $\tilde{c}_{lk}$s. For *Case A* every member of the sum in equation (15b) is multiplied by 2 in order to simplify the calculations. This value 2 comes from the denominator part of the original basis functions (3). The following formulas for the unknowns are obtained. The formulas have been presented separately for *Case A* and *Case B*.

A. *Case A Implementation*

For Case A implementation there are two sets of unknowns to be solved, namely, the coefficients a_{lk}s and the unknowns c_{lk}s in order to meet the three conditions stated in the previous section. The unknowns are altered so that the denominator part in the basis function (3) is also taken into account as the final step.

For this purpose the whole equation system is derived by using equations (10), (15a), (15b) and (15c) and by exploiting the relation mentioned in (8).

The derivation of the whole equation system and solving the coefficients a_{lk}s and the unknowns c_{lk}s are done as follows: *Algorithm for Case A implementation:*

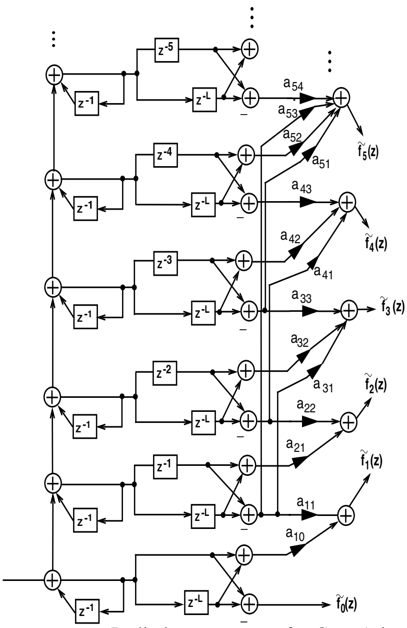


Fig. 3. Preliminary structure for Case A implementation

Step 1 Generate the following polynomials:

$$P_0(n) = c_{00} = 1 \qquad (15a)$$

odd-order polynomials as:

$$P_l(n) = \sum_{k=1}^{(l+1)/2} 2c_{l(2k-1)}\widehat{f}_{2k-1}(n), c_{l1} = 1, \qquad (15b)$$

where l is the odd-degree of the polynomial and *even-order* polynomials as:

$$P_l(n) = \sum_{k=0}^{\frac{l}{2}-1} c_{lk}[\widehat{f}_{(l-2k)}(n) - \widehat{f}_{(l-2k)}(0)], c_{l0} = 1, \qquad (15c)$$

where l is the even-degree of the polynomial. In the above equations, c_{lk}s are the unknowns and $\widehat{f}_{(l-2k)}(n)$s are the basis functions in (14).

Step 2 Solve the unknowns, c_{lk}s, so that $P_l(n)$ for $l > 0$ becomes zero for $n = 1, 2, \ldots$

Step 3 Solve the coefficients, a_{lk}s from

$$\sum_{k=1}^{l}(a_{lk}\widehat{q}_k(n)) - P_l(n) = 0 \text{ for } n = 1, 2, \ldots, R$$

for $l > 1$ and

$$\sum_{k=0}^{l}(a_{lk}\widehat{q}_k(n)) - P_l(n) = 0 \text{ for } l = 1, n = 0, 1, \widehat{q}_0 = 1 \qquad (16)$$

Step 4 Generate the unknowns $\tilde{c}_{lk}$s as follows: for even polynomial degree, l

$$\tilde{c}_{lk} = c_{lk}(\tfrac{L}{2} - \tfrac{1}{2})^k, \ l > 0, \forall \text{ even } k > 0$$
$$\tilde{c}_{l0} = -c_{ll} - c_{lm_1} - c_{lm_2} \cdots - c_{l2}, m_1 = l - 2 \text{ and } m_2 = l - 4 \qquad (17)$$

for odd polynomial degree, l

$$\tilde{c}_{lk} = 2c_{lk}(\frac{L}{2} - \frac{1}{2})^k, \ \forall \text{ odd k} \qquad (18)$$

In the above equations, $(\frac{L}{2} - \frac{1}{2})^k$ is the denominator part of the original basis functions, $f_l(n)$.

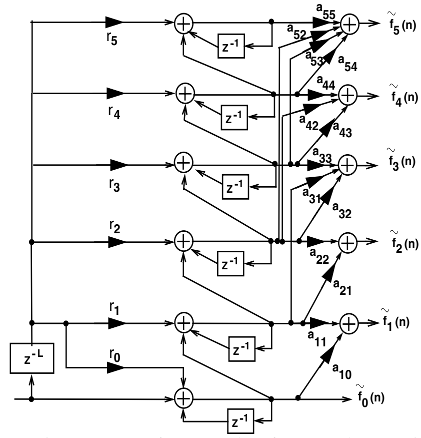


Fig. 4. Basic structure of a cascade of accumulators with $r_0 = 1$ and $r_k = \Pi_{l=1}^{k}(L - 1 + l)$ [4].

From (16) as many equations are obtained as there are unknowns. If basis functions (3) are used, the c_{lk}s are directly the $\tilde{c}_{lk}$s in (6), and from (16) integer-valued coefficients, a_{lk}s, are obtained directly without further alterations. If basis functions (14) are used, the c_{lk}s have to be altered according to (17) and (18) in order to obtain $\tilde{c}_{lk}$s. The same coefficients are obtained by either way. Results from these formulas are in Tables I and II.

B. Case B Implementation

For Case B implementation there are also unknowns to be solved: the coefficients a_{lk}s and the unknowns c_{lk}s and $\tilde{c}_{lk}$s. The equation system is derived by using (14), and by exploiting the relation in (11).

The Roman numbers after the degree number in Tables III and IV indicate the alternatives for the specific polynomial degree, e.g. degrees 3 and 4 have two different alternatives and degree 5 has four alternatives for the combination of the unknowns. The coefficient alternatives in Table III, which have a lower polynomial order should be chosen.

Now the algorithm to solve the coefficients a_{lk}s and the unknowns c_{lk}s and the altered unknowns $\tilde{c}_{lk}$s is given. *The algorithm for Case B implementation:*

Step 1 Generate the following polynomials:

$$P_0(n) = c_{00} = 1 \qquad (19a)$$

even-order polynomials as:

$$P_l(n) = \sum_{k=0}^{\frac{l}{2}} c_{lk}\widehat{f}_{((2k+l)-l)}(n), \ c_{l0} = 1, \qquad (19b)$$

where l is the even-degree of the polynomial; *odd-order* polynomials as:

$$P_l(n) = \sum_{k=1}^{(l+1)/2} c_{l(2k-1)}\widehat{f}_{2k-1}(n), \qquad (19c)$$

where l is the odd-degree of the polynomial. In the above equations, c_{lk}s are the unknowns and $\widehat{f}_{(l-2k)}(n)$s are the basis functions in (14).

Step 2 Solve the coefficients, a_{lk}s, from

$$P_l(n) - \sum_{k=0}^{l} a_{lk}q_k(n) = 0 \qquad (20)$$
$$\text{for } n = 0, 1, 2, \ldots, R.$$

Step 3 Solve the unknowns c_{lk}s so that $P_l(n)$ for $l > 0$ becomes zero for $n = 1, 2, \ldots$. For orders higher than 2, make some of the a_{lk}s equal to zero.

Step 4 Solve a_{lk}s by using c_{lk}s.

Step 5 Separate the largest denominator D of a_{lk}s.
Step 6 Generate the altered unknowns $\tilde{c}_{lk}$s by using

$$\tilde{c}_{lk} = Dc_{lk}(\frac{L}{2} - \frac{1}{2})^k, \text{if } D \neq 1 \quad (21)$$

and

$$\tilde{c}_{lk} = c_{lk}(\frac{L}{2} - \frac{1}{2})^k, \text{if } D = 1, \quad (22)$$

where k is the order of the polynomial, and D is the largest denominator part of the noninteger-valued coefficients.

Even if it may seem that there are unnecessary steps, the steps are determined so that the smallest coefficient values are obtained. Formulas (20), (21) and (22) guarantee that the coefficients, a_{lk}, are integer-valued. Step 6 is carried out if the basis functions in (14) are used. If (3) is used as a basis function, formulas (21) and (22) can be used to solve the altered unknowns, $\tilde{c}_{lk}$s but without the term, $(\frac{L}{2} - \frac{1}{2})^k$. For polynomial orders higher than 5, the coefficients become very large, if the block length, L, is large, so in practice those orders are not realisable. This is due to the fact that the value of coefficients increases with the polynomial degree. This structure does not cause any overflows. Results from these formulas are in Tables III and IV.

IV. Implementation Coefficients

For *Case A* implementation there is only one alternative for the desired combination of the coefficients for every polynomial degree as shown in Table I. *Case B* implementation, in turn, has several implementation structures. The different coefficient alternatives up to degree 5 are shown in Table III for *Case B* implementation. For the polynomial degree higher than 5, the coefficient values become too large for practical implementation purposes especially for large values of L. As seen fom Tables I and III, already for degree 5, some of the coefficient values are at least of order L^3. This is valid for both structures. The $\tilde{c}_{lk}$s , required to modify the original coefficients of the $G_l(z)$s to become of those $\tilde{G}_l(z)$s according to (5), are summarized in Tables II and IV for *Case A* and *Case B*, respectively.

TABLE I
Coefficients for *Case A* Implementation

Degree	Polynomial coefficients
0.	$a_{00} = 1$
1.	$a_{10} = 1 - L, \quad a_{11} = 2$
2.	$a_{20} = 0, \ a_{21} = 2 - L, \ a_{22} = 2$
3.	$a_{30} = 0, \ a_{31} = 6 - 5L + L^2,$ $a_{32} = 18 - 6L, \ a_{33} = 12,$
4.	$a_{40} = 0, \ a_{41} = 0,$ $a_{42} = 24 - 14L + L^2, \ a_{43} = 48 - 12L,$ $a_{44} = 24$
5.	$a_{50} = 0, \ a_{51} = 0,$ $a_{52} = -2L^3 + 120 - 94L + 24L^2,$ $a_{53} = 480 - 216L + 24L^2, \ a_{54} = 600 - 120L,$ $a_{55} = 240$

TABLE II
The $\tilde{c}_{lk}$s for *Case A* Implementation

Degree	$\tilde{c}_{lk}$s
1.	$\tilde{c}_{11} = (L-1)$
2.	$\tilde{c}_{20} = -(L-1)^2/4,$ $\tilde{c}_{22} = (L-1)^2/4$
3.	$\tilde{c}_{31} = -2((L-1)/2)^3,$ $\tilde{c}_{33} = 2((L-1)/2)^3$
4.	$\tilde{c}_{40} = -(L-1)^4/16 + ((L-2)^2+1)/2$ $\times((L-1)/2)^2,$ $\tilde{c}_{42} = -((L-1)^2/4)((L-2)^2+1)/2,$ $\tilde{c}_{44} = (L-1)^4/16$
5.	$\tilde{c}_{51} = (L-1)(L^4/16 - (1/2)L^3 + (11/8)L^2$ $-(3/2)L + 9/16),$ $\tilde{c}_{53} = ((L-1)^3/8)(-L^2+4L-5),$ $\tilde{c}_{55} = 2((L-1)/2)^5$

V. Conclusion

Efficient and easy-to-use mathematical formulas were shown by thorough derivation for the dedicated implementation structures of narrowband linear-phase FIR filters with piecewise-polynomial impulse response.

TABLE III
Coefficients for *Case B* Implementation

Degree	Polynomial coefficients
0.	$a_{00} = 1$
1.	$a_{10} = -1 - L, \quad a_{11} = 2$
2.	$a_{20} = 0, \quad a_{21} = -2 - L, \quad a_{22} = 2$
3.I	$a_{30} = 0, \quad a_{31} = L^2 + 5L + 6, \ a_{32} = -6L - 18,$ $a_{33} = 12$
3.II	$a_{30} = 6 + 11L + 6L^2 + L^3, \quad a_{31} = 0,$ $a_{32} = -12L - 36, a_{33} = 24$
4.I	$a_{40} = 0, \ a_{41} = 0, \quad a_{42} = 2L^2 + 14L + 24,$ $a_{43} = -48 - 12L \ a_{44} = 24$
4.II	$a_{40} = 0, \ a_{41} = L^3 + 9L^2 + 26L + 24, \quad a_{42} = 0$ $a_{43} = -48 - 12L, \quad a_{44} = 24$
5.I	$a_{50} = 0, \ a_{51} = 0, a_{52} = -L^3 - 12L^2 - 47L - 60$ $a_{53} = 12L^2 + 108L + 240, \ a_{54} = -60L - 300,$ $a_{55} = 120$
5.II	$a_{50} = 0, \ a_{51} = -L^4 + -14L^3 - 120 - 154L - 71L^2$ $a_{52} = 0, \ a_{53} = 1200 + 540L + 60L^2$ $a_{54} = -360L - 1800, \quad a_{55} = 720$
5.III	$a_{50} = -225L^2 - 85L^3 - 120 - 274L - 15L^4 - L^5$ $a_{51} = 0 \ a_{52} = 0, \ a_{53} = 120L^2 + 2400 + 1080L$ $a_{54} = -3600 - 720L, \ a_{55} = 1440$
5.IV	$a_{50} = 0, \ a_{51} = -L^4 - 14L^3 - 120 - 154L - 71L^2$ $a_{52} = 5L^3 + 60L^2 + 235L + 300, \quad a_{53} = 0$ $a_{54} = -300 - 600L, a_{55} = 120$

TABLE IV
The $\tilde{c}_{lk}$s for *Case B* Implementation

Degree	$\tilde{c}_{lk}$s
1.	$\tilde{c}_{11} = (L-1)$
2.	$\tilde{c}_{20} = -(L+1)^2/4, \quad \tilde{c}_{22} = (L-1)^2/4$
3.I	$\tilde{c}_{31} = -(L-1)(L+1)^2/4, \ \tilde{c}_{33} = (L-1)^3/4$
3.II	$\tilde{c}_{31} = -((L-1)/2)(3L^2 + 12L + 13), \ \tilde{c}_{33} = (L-1)^3/2$
4.I	$\tilde{c}_{40} = 9/16 + (3/2)L + (11/8)L^2 + (1/2)L^3 + (1/16)L^4$ $\tilde{c}_{42} = -(L-1)^2(L^2 + 4L + 5)/8$ $\tilde{c}_{44} = (L-1)^4/16$
4.II	$\tilde{c}_{40} = 57/16 + (37/4)L + (65/8)L^2 +$ $(11/4)L^3 + (5/16)L^4$ $\tilde{c}_{42} = -(L-1)^2(3L^2 + 18L + 29)/8$ $\tilde{c}_{44} = (L-1)^4/16$
5.I	$\tilde{c}_{51} = ((L-1)/32)(L+3)^2(L+1)^2$ $\tilde{c}_{53} = -(L-1)^3(L^2 + 4L + 5)/16, \ \tilde{c}_{55} = (L-1)^5/32$
5.II	$\tilde{c}_{51} = (L+1)^2(L-1)(7L^2 + 54L + 107)/16$ $\tilde{c}_{53} = -5(L-1)^3(L^2 + 6L + 11)/8$ $\tilde{c}_{55} = 3(L-1)^5/16$
5.III	$\tilde{c}_{51} = (L-1)(790L^2 + 1067+$ $+15L^4 + 180L^3 + 1500L)/8$ $\tilde{c}_{53} = -5(L-1)^3(L^2 + 6L + 11)/4$ $\tilde{c}_{55} = 3(L-1)^5/8$
5.IV	$\tilde{c}_{51} = (L-1)(L+1)^2(3L+13)^2/32$ $\tilde{c}_{53} = -(L-1)^3 5(17 + L^2 + 8L)/16$ $\tilde{c}_{55} = (L-1)^5/32$

References

[1] G. F. Boudreaux and T. W. Parks, "Thinning digital filters: A piecewise-exponential approximation approach,“ *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 105–113, Feb. 1983.

[2] S. Chu and S. Burrus, "Efficient recursive realizations of FIR filters, Part I: The filter structures; Part II: Design and applications, " *Circuits, Systems, and Signal Processing*, vol. 3, no. 1, pp. 2–57, 1984.

[3] T. G. Campbell and T. Saramäki, "Recursive linear-phase FIR filter structures with piecewise-polynomial impulse response," in *Proc. Sixth Int. Symp. Networks, Systems and Signal Processing*, Zagreb, Yuogoslavia, pp. 16–19, June 1989.

[4] T. Saramäki and O. Vainio, "Structures for generating polynomial responses," in *Proc. 37th Midwest Symp. Circuits and Systems*, vol. 2, pp. 1315–1318, Aug. 1994.

[5] T. Saramäki and S. K. Mitra, "Design and implementation of narrow-band linear-phase FIR filters with piecewise polynomial impulse response," in *Proc. IEEE Int. Symp. Circuits and Systems*, Orlando, FL, vol 3, pp.456–461, July 1999.

[6] R. Lehto, T. Saramäki and O. Vainio, "Synthesis of narrowband linear-phase FIR filters with a piecewise-polynomial impulse response," in *Proc. IEEE Int. Symp. Circuits and Systems*, Kobe, Japan, pp. 2012–2015, May 2005.

Publication-II

R. Lehto, T. Saramäki and O. Vainio, Synthesis of narrowband linear-phase FIR filters with a piecewise-polynomial impulse response, *IEEE Trans. Circuits and Systems 1: Regular Papers*, vol. 54, no. 10, pp. 2262–2276, Oct. 2007.

Synthesis of Narrowband Linear-Phase FIR Filters with a Piecewise-Polynomial Impulse Response*

Raija Lehto, Tapio Saramäki, *Fellow, IEEE,* and Olli Vainio, *Senior Member, IEEE*

* *This paper is dedicated to the memory of our friend T. George Campbell.*

***Abstract*— Classes of linear-phase finite impulse response (FIR) filters with a piecewise-polynomial impulse response are proposed for the four types of linear-phase FIR filters. In addition, very efficient recursive structures to implement these filters in a straightforward and consistent manner are proposed.**

The desired impulse response is created by using a parallel connection of several filter branches. Only one branch has an impulse response of the full filter length, whereas the impulse responses are shorter for the remaining branches but the center is at the same location. The arithmetic complexity of these filters is proportional to the number of branches and the common polynomial order for each branch, rather than the actual filter order. In order to generate the overall piecewise-polynomial impulse response the polynomial coefficients are found, with the aid of linear programming, by optimizing the responses in the minimax sense, for both narrowband conventional filters and narrowband differentiators.

The generation of these structures is based on the use of accumulators so that after using an accumulator, the resulting impulse response is divided into two parts. The first part follows the desired polynomial form, and the second part is what is left after the division, i.e., the non-polynomial part. This same procedure can be used for all the following accumulators.

Several examples are included, illustrating the benefits of the proposed filters, in terms of a reduced number of unknowns used in the optimization and the reduced number of multipliers required in the actual implementation.

***Index Terms*— FIR digital filters, linear-phase filters, polynomials, piecewise polynomial, linear programming, filter structures.**

I. Introduction

In many filtering applications, finite-impulse response (FIR) digital filters are preferred over their infinite-impulse response (IIR) counterparts due to their many favorable properties. These properties include, among others, the following. First, an FIR filter can be designed with an exact linear phase, which means that no phase distortion is caused in the signal during the filtering operation. Secondly, both the output noise due to multiplication round-off errors and the sensitivity to variations in filter coefficients are lower. The main drawback of FIR filters is a higher number of multipliers needed in a conventional implementation when a narrow transition band is required [1]. This fact makes the implementation of direct-form linear-phase FIR filters with a narrow transition band very costly compared to the corresponding IIR filters, in terms of their arithmetic complexity, i.e., the required number of adders, multipliers, and delay elements. However, in many applications, the various advantages of FIR filters compared to their IIR counterparts in practice often outweigh the difficulties in implementing these filters, especially if the correlation between the impulse response samples is exploited by using proper implementations structures (see, for example, the references in [1] and [8]).

An efficient approach to overcome the above-mentioned problem is to synthesize linear-phase FIR filters so that their impulse response is piecewise-polynomial and the implementation is performed using recursive structures [2–8]. The arithmetic complexity of these filters is proportional to the number of impulse-response pieces and the overall polynomial order rather than the actual filter order.

Boudreaux and Parks [2] were among the first to propose a recursive piecewise-polynomial approximation of an impulse response of FIR filters. The piecewise-polynomial filters by Chu and Burrus [3–4] are a generalization of the filters proposed by Boudreaux and Parks. The design scheme of Chu and Burrus suffers from the following drawbacks. First, the derivation of the overall filter structure, based on the polynomial coefficients used to generate the overall impulse response, is very complicated and it does not arrive at the best available implementation form. Secondly, the optimization of the polynomial coefficients has been performed using nonlinear optimization. In order to overcome these problems, Campbell and Saramäki [5] presented a new preliminary filter structure based on and properly modifying the synthesis scheme used by Chu and Burrus. Saramäki and Mitra [7] have a more straightforward approach for their piecewise-polynomial filters with the restriction that all the blocks, where the impulse response follows a piecewise-polynomial, are of the equal length. In this approach the overall filter length is the number of blocks multiplied by the block length. Saramäki and Vainio [6] have made a proposal to synthesize narrowband linear-phase FIR filters with a piecewise-polynomial impulse response. The proposed structures are based on polynomials of increasing degrees and a cascade of accumulators.

Also other approaches have been developed to reduce the arithmetic complexity of FIR filters in narrowband applications. One of the most efficient techniques to be used is the interpolated FIR (IFIR) filters [1], [9], which are constructed from two subfilters, one of which uses multiple delay elements

Manuscript was received. This work was supported by the Academy of Finland, Project No. 213462 (Finnish Centre of Excellence Program (2006 - 2011)).

Some parts of the material in this paper were presented at the 2005 IEEE International Symposium on Circuits and Systems, Kobe, Japan.

Raija Lehto and Tapio Saramäki are with the Institute of Signal Processing at Tampere University of Technology, FI-33101 Tampere, Finland, email: raija.lehto@tut.fi, tapio.saramaki@tut.fi.

Olli Vainio is with the Institute of Digital and Computer Systems at Tampere University of Technology, FI-33101 Tampere, Finland, email: olli.vainio@tut.fi.

instead of unit delays. Its frequency response is therefore periodic. The unwanted replicated passbands and transition bands of the periodic filter are removed with a nonperiodic filter, which has a wider transition band than in the overall specification. This approach leads to a significantly reduced number of multipliers and adders required to provide the desired transition band.

Type 3 and 4 FIR filters can be used for narrowband differentiator applications. Differentiators are used to perform a differentiation operation on discrete-time signals. They can be used e.g. in biomedical engineering and motion analysis [10] and several other applications [11–13]. Full-band differentiators cause noise amplification in a digital differentiation process. Lowpass differentiators can be used in order to avoid this undesirable phenomenon. Some of these designs are optimized for frequencies around $\omega = \pi/2$ [14–16].

In this paper, an approach is proposed to synthesize piecewise-polynomial linear-phase FIR filters, based on the preliminary structure proposed by Campbell and Saramäki. This approach overcomes the drawbacks of the design scheme of Chu and Burrus by first introducing a straightforward approach to synthesize the overall filter for all four types of linear-phase FIR filters, and leading to efficient implementation structures for all types. These structures are derived in Section V. Secondly, linear programming is used effectively to optimize the unknown polynomial coefficients in the overall filter. Several design examples for both conventional filters and narrowband differentiators are included in order to show the benefits of the proposed design approach and the resulting filters.

II. Motivation

Why do we use a piecewise-polynomial approach to create impulse responses? If we consider an impulse response of an optimum linear-phase FIR filter, it is seen in Fig. 1(a) that the impulse response has a very smooth shape, which means that there is a strong correlation between successive impulse response values. Therefore, piecewise use of polynomials is motivated to create such impulse responses. In order to reduce the number of coefficients and to obtain efficient implementation structures, we use this characteristic to our advantage.

How do we exploit this piecewise-polynomial shape? The idea is to divide the impulse response into subresponses and to generate each subresponse with polynomials of a given degree. What makes this piecewise? These subresponses are of different lengths and after summing them up the overall impulse response is obtained with a different number of polynomials forming each block of the overall impulse response. Consider the filter in Case 1, with the polynomial degree three, in Section VI. It has the same specification as the optimum linear-phase FIR filter in Fig. 1. The filter in Case 1 consists of five subresponses of different lengths; thereby, each block of the impulse response consists of a different number of polynomials, which are shown in Fig. 2: e.g. the first block consists of one polynomial, the second block of two polynomials, and the (left) middle block of the impulse response consists of five polynomials. There are totally ten blocks in the overall impulse response.

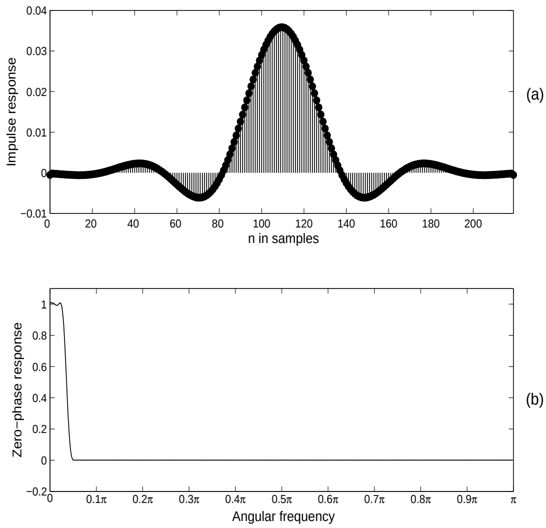


Fig. 1. Typical (a) impulse response and (b) zero-phase frequency response for a narrowband linear-phase FIR filter. The filter has been optimized by using the Remez multiple exchange algorithm and it has the minimum order, $N = 215$, to meet the specifications: Passband edge $\omega_p = 0.025\pi$, stopband edge $\omega_s = 0.05\pi$, passband ripple $\delta_p = 0.01$, stopband ripple $\delta_s = 0.001$.

After summing up these subresponses in Fig. 2, the overall impulse response is obtained and it is shown in Fig. 3. It has the same smooth and piecewise-polynomial shape as the optimum linear-phase FIR filter in Fig. 1(a). The five subresponses along with the polynomial degree determine the number of filter coefficients in both optimization and implementation. The subresponses are generated with polynomials of degree three in each subresponse and, therefore, the number of coefficients becomes 20 in the optimization. Now we propose

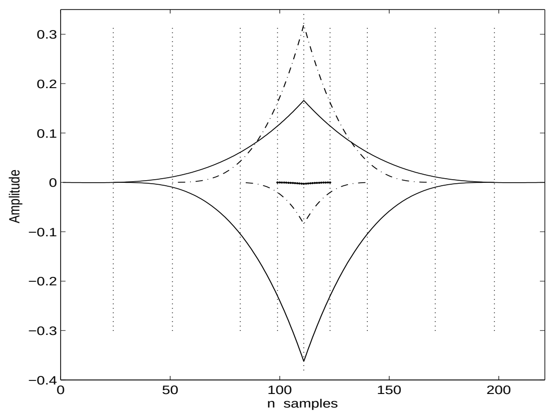


Fig. 2. Subimpulse-responses, solid up, the longest subresponse, solid down, the second longest, dashdotted up and down, the third and fourth longest subresponses, and the shortest subresponse in the middle. The vertical lines mark the block edges. The first block consists of one polynomial, the second one of two polynomials, the third one of three polynomials, the fourth one of four and the (left) middle block of five polynomials.

the linear-phase FIR filter classes which make this possible.

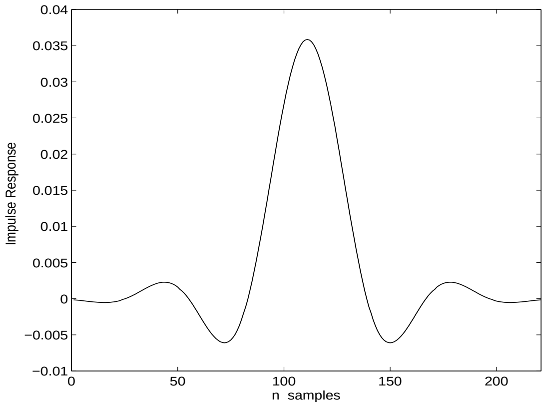


Fig. 3. Overall impulse response obtained from five subresponses with the polynomial degree three in each.

III. Proposed Linear-Phase FIR Filter Classes

The basic idea is to generate an arbitrary piecewise-polynomial impulse response for each of the four linear-phase FIR filter types. It starts by constructing the overall transfer function $H(z)$ as a parallel connection of M branch filter transfer functions as follows:

$$H(z) = \sum_{m=1}^{M} z^{-N_m} H_m(z). \tag{1}$$

The desired overall transfer function of order $2N$ for Types 1 and 3 and of order $2N-1$ for Types 2 and 4 can be generated as follows. First, the transfer functions, $H_m(z)$ for $m = 1, 2, \ldots, M$, are constructed for Types 1, 2, 3, and 4 as

$$H_m(z) = \begin{cases} h_m(N-N_m)z^{-(N-N_m)} + \\ \displaystyle\sum_{n=0}^{(N-N_m)-1} h_m(n)[z^{-n} + z^{-(2(N-N_m)-n)}] \\ \displaystyle\sum_{n=0}^{(N-N_m)-1} h_m(n)[z^{-n} + z^{-(2(N-N_m)-1-n)}] \\ \displaystyle\sum_{n=0}^{(N-N_m)-1} h_m(n)[z^{-n} - z^{-(2(N-N_m)-n)}] \\ \displaystyle\sum_{n=0}^{(N-N_m)-1} h_m(n)[z^{-n} - z^{-(2(N-N_m)-1-n)}], \end{cases} \tag{2}$$

respectively, where the integers N_m in the delay terms z^{-N_m} satisfy

$$N_1 = 0 \text{ and } N_{m+1} > N_m \text{ for } m = 1, 2, \ldots, M-1. \tag{3}$$

Secondly, an approximation of each subresponse of the overall impulse response by an Lth order polynomial is achieved by expressing $h_m(n)$ in terms of an Lth-degree polynomial as

$$h_m(n) = p_k^{(L)}(n) = \sum_{r=0}^{L} a_m^{(L)}(r) n^r \text{ for } 0 \le n \le R_m, \tag{4}$$

where

$$R_m = \begin{cases} N - N_m & \text{for Type 1} \\ N - N_m - 1 & \text{for Types 2, 3, and 4.} \end{cases} \tag{5}$$

The transfer functions, $H_m(z)$, as given by (2), are generated so that their impulse responses automatically satisfy the desired symmetry conditions for Types 1, 2, 3, and 4. The corresponding impulse responses for these four types are respectively given by

$$h_m(n) = \begin{cases} \displaystyle\sum_{r=0}^{L} a_m^{(L)}(r) n^r, & n = 0, 1, \ldots, (N-N_m) \\ h_m(2(N-N_m)-n), & n = (N-N_m)+1, \\ & \quad \ldots, 2(N-N_m) \\ 0, & \text{otherwise,} \end{cases} \tag{6a}$$

$$\begin{cases} \displaystyle\sum_{r=0}^{L} a_m^{(L)}(r) n^r, & n = 0, 1, \ldots, (N-N_m)-1 \\ h_m(2(N-N_m)-n), & n = (N-N_m), (N-N_m)+1, \\ & \quad \ldots, 2(N-N_m)-1 \\ 0, & \text{otherwise,} \end{cases} \tag{6b}$$

$$\begin{cases} \displaystyle\sum_{r=0}^{L} a_m^{(L)}(r) n^r, & n = 0, 1, \ldots, (N-N_m)-1 \\ 0, & n = (N-N_m) \\ -h_m(2(N-N_m)-n), & n = (N-N_m)+1, \ldots, 2(N-N_m) \\ 0, & \text{otherwise,} \end{cases} \tag{6c}$$

and

$$\begin{cases} \displaystyle\sum_{r=0}^{L} a_m^{(L)}(r) n^r, & n = 0, 1, \ldots, (N-N_m)-1 \\ -h_m(2N-1-n), & n = (N-N_m), (N-N_m)+1, \\ & \quad \ldots, 2(N-N_m)-1 \\ 0, & \text{otherwise.} \end{cases} \tag{6d}$$

Therefore, for the mth transfer function, $H_m(z)$, the filter order and the center of the symmetry are given by

$$N_m^{order} = \begin{cases} 2(N-N_m) & \text{for Types 1 and 3} \\ 2(N-N_m)-1 & \text{for Types 2 and 4} \end{cases} \tag{7a}$$

and

$$N_m^{center} = \begin{cases} N-N_m & \text{for Types 1 and 3} \\ (2(N-N_m)-1)/2 & \text{for Types 2 and 4,} \end{cases} \tag{7b}$$

respectively. It is worth emphazising that, due to the symmetry condition for Type 3 filters, $h((N-N_m)/2) = 0$. Therefore, the time interval, $[0, R_m]$ with R_m as given by (5), is the widest one where the impulse response can follow the Lth-degree polynomial, as given by (4).

The center of the resulting impulse response of $H_m(z)$ for $m > 1$ is not located at

$$N_m^{center} = \begin{cases} N & \text{for Types 1 and 3} \\ (2N-1)/2 & \text{for Types 2 and 4,} \end{cases} \tag{8}$$

as required when generating Type 1 and 3 filters of order $2N$ and Type 2 and 4 filters of order $2N-1$. Therefore, the branch filter transfer functions for $m > 1$ in the overall transfer function, as given by (1), have additional delay terms z^{-N_m} shifting the center of each impulse response to the desired location. When using these additional delay terms, the impulse response of the overall filter can be expressed as

$$h(n) = \sum_{m=1}^{M} \widehat{h}_m(n), \tag{9}$$

where $\widehat{h}_m(n)$ is given by

$$\widehat{h}_m(n) = h_m(n - N_m). \tag{10}$$

The resulting impulse response, $\widehat{h}_m(n)$, for $m = 1, 2, \ldots, M$ is thus given by $\widehat{h}_m(n) =$

$$\begin{cases} 0, & n = 0, 1, \ldots, N_m - 1 \\ \sum_{r=0}^{L} a_m^{(L)}(r)(n - N_m)^r, & n = N_m, N_m + 1, \ldots, N \\ \widehat{h}_m(2N - n), & n = N + 1, N + 2, \ldots, 2N - N_m \\ 0 & \text{otherwise,} \end{cases} \tag{11}$$

$$\begin{cases} 0, & n = 0, 1, \ldots, N_m - 1 \\ \sum_{r=0}^{L} a_m^{(L)}(r)(n - N_m)^r, & n = N_m, N_m + 1, \ldots, N - 1 \\ \widehat{h}_m(2N - n), & n = N, N + 1, \ldots, 2N - N_m - 1 \\ 0 & \text{otherwise,} \end{cases} \tag{12}$$

$$\begin{cases} 0, & n = 0, 1, \ldots, N_m - 1 \\ \sum_{r=0}^{L} a_m^{(L)}(r)(n - N_m)^r, & n = N_m, N_m + 1, \ldots, N - 1 \\ 0, & n = N \\ -\widehat{h}_m(2N - n), & n = N + 1, N + 2, \ldots, 2N - N_m - 1 \\ 0 & \text{otherwise,} \end{cases} \tag{13}$$

and

$$\begin{cases} 0, & n = 0, 1, \ldots N_m - 1 \\ \sum_{r=0}^{L} a_m^{(L)}(r)(n - N_m)^r, & n = N_m, N_m + 1, \ldots, N - 1 \\ -\widehat{h}_m(2N - n), & n = N, N + 1, \ldots, 2N - 1 \\ 0 & \text{otherwise,} \end{cases} \tag{14}$$

respectively. The above impulse response for Types 1 and 3 can be regarded as a filter of order $2N$ and for Types 2 and 4 as a filter of order $2N-1$. It is characterized by the fact that there exist $N_m - 1$ zero-valued samples at both the beginning and end of the impulse response. The only exception is $\widehat{h}_1(n)$, for which $N_1 = 0$ according to (3). Therefore, this impulse response has no zero-valued samples at the beginning and end of the overall impulse response and is given as

$$h_1(n) = \sum_{r=0}^{L} a_1^{(L)}(r) n^r \tag{15}$$

for $n = 0, 1, \ldots, N$, for Type 1 and for $n = 0, 1, \ldots, N-1$ for Types 2, 3 and 4. $h_1(n)$ satisfies the same symmetry conditions as the response $\widehat{h}_m(n)$ for $m = 2, 3, \ldots, M$. Thereby, the overall response can be expressed as M slices, X_m, as follows:

$$X_1 = \left[N_1, N^{order}\right] \tag{16a}$$

$$X_m = \left[N_m, N^{order} - N_m\right] \tag{16b}$$

for $m = 2, 3, \ldots, M$, where

$$N^{order} = \begin{cases} 2N & \text{for Types 1 and 3} \\ 2N - 1 & \text{for Types 2 and 4} \end{cases} \tag{16c}$$

and N_ms are in increasing order for $m = 1, 2, \ldots, M$, i.e., $N_m < N_{m+1} < \ldots < N_M$. It is possible also to express slices X_m, with the aid of blocks, as follows:

$$\widehat{X}_1 = [N_1, N_2 - 1] \cup \left[N^{order} - N_2 + 1, N^{order}\right] \tag{17a}$$

$$\widehat{X}_m = [N_m, N_{m+1} - 1] \cup \left[N^{order} - N_{m+1} + 1, N^{order} - N_m\right] \tag{17b}$$

for $m = 2, 3, \ldots, M - 1$ and

$$\widehat{X}_M = \left[N_M, N^{order} - N_M\right], \tag{17c}$$

where the center section of the slice X_m is missing in each $\widehat{X}_m$ except in the $\widehat{X}_M$. Thereby one slice consists of two blocks as shown in Fig. 2, e.g., the first part of the $\widehat{X}_1$ is located from 0 to the first vertical line and the second part of the $\widehat{X}_1$ is located from the last vertical line to N^{order}. In the first part of each slice X_m for $m = 1, 2, \ldots, M-1$ as well as in slice X_M for $n = N_M, \ldots, \widehat{N}$, where $\widehat{N}$ is N for Type 1 and $N-1$ for Types 2, 3, and 4, the impulse-response of the overall filter can be generated to follow an arbitrary Lth-degree polynomial. For the rest of each slice, the impulse response is automatically generated so that the overall response satisfies the desired conditions for each type of linear-phase FIR filters.

This is based on the following facts. First, with each $\widehat{X}_m$ it holds that

$$h(n) = \sum_{m=1}^{M} \tilde{h}_m(n). \tag{18}$$

Secondly, based on the above-mentioned definitions of the $\widehat{h}_m(n)$s in (11), (12), (13), and (14), the overall impulse response in $\widehat{X}_m$ can be expressed for Types 1, 2, 3, and 4 as follows:

$$\tilde{h}_m(n) = \begin{cases} \sum_{k=1}^{m} \left[\sum_{r=0}^{L} a_k^{(L)}(r)(n - N_m)^r\right] \\ \text{for } n = N_m, N_m + 1, \ldots, N_{m+1} - 1 \\ h(2N - n) \\ \text{for } n = 2N - N_{m+1} + 1, 2N - N_{m+1}, \ldots, 2N - N_m \\ 0, \quad \text{otherwise,} \end{cases} \tag{19}$$

$$
\tilde{h}_m(n) = \begin{cases} \sum_{k=1}^{m} \left[\sum_{r=0}^{L} a_k^{(L)}(r)(n-N_m)^r \right] \\ \text{for } n = N_m, N_m+1, \ldots, N_{m+1}-1 \\ \\ h(2N-n) \\ \text{for } n = 2N-1-N_{m+1}, 2N-N_{m+1}+1, \\ \qquad \ldots, 2N-1-N_m \\ 0, \quad \text{otherwise}, \end{cases} \tag{20}
$$

$$
\tilde{h}_m(n) = \begin{cases} \sum_{k=1}^{m} \left[\sum_{r=0}^{L} a_k^{(L)}(r)(n-N_m)^r \right] \\ \text{for } n = N_m, N_m+1, \ldots, N_{m+1}-1 \\ \\ 0 \quad \text{for } n = N \\ \\ -h(2N-n) \\ \text{for } n = 2N-N_{m+1}+1, 2N-N_{m+1} \\ 0, \quad \text{otherwise}, \end{cases} \tag{21}
$$

and

$$
\tilde{h}_m(n) = \begin{cases} \sum_{k=1}^{m} \left[\sum_{r=0}^{L} a_k^{(L)}(r)(n-N_m)^r \right] \\ \text{for } n = N_m, N_m+1, \ldots, N_{m+1}-1 \\ \\ -h(2N-n) \\ \text{for } n = 2N-1-N_{m+1}, 2N-N_{m+1}, \\ \qquad \ldots, 2N-1-N_m \\ 0, \quad \text{otherwise}, \end{cases} \tag{22}
$$

respectively.

In each block of $\widehat{X}_m$ for $m = 1, 2, \ldots, M$, an arbitrary Lth-degree polynomial impulse response can be provided, as proposed in (19), (20), (21), and (22). Based on the above equations, it is seen that in each block $\widehat{X}_m$ for $m = 2, 3, \ldots M$ compared to block $\widehat{X}_{m-1}$ there is one more polynomial, namely, $\sum_{r=0}^{L} a_k^{(m)}(r)(n-N_m)^r$ providing an additional arbitrary Lth-degree polynomial contribution to this block. The only exception in the above consideration is block $\widehat{X}_1$, where only $\sum_{r=0}^{L} a_k^{(1)}(r)(n-N_m)^r$ provides its contribution.

The above can also be considered so that the slices are of variable or equal length with the center at the same location. The first slice is of the same length as the overall impulse response. The rest of the slices are shorter and, therefore, there are zeros at the beginning and end of the impulse response. If slices N_ms are chosen so that $|N_2 - N_1| = |N_3 - N_2| = \cdots = |N_M - N_{M-1}|$, where $N_1 = 0$ and M is the number of slices in the overall impulse response, they are called uniformly spaced and otherwise non-uniformly spaced.

IV. Filter Optimization Algorithm

FIR filter optimization using linear programming can be stated as a problem of minimizing

$$
\epsilon = \max_{\omega \in [0, \omega_p][\omega_s, \pi]} |W(\omega)||H(\omega) - D(\omega)|, \tag{23}
$$

where $W(\omega)$ is a weight function and $D(\omega)$ is the desired zero-phase frequency response. In the above optimization problem, ω_p, and ω_s are the passband and stopband edge angles, respectively. The most crucial feature for the filters under consideration is that the zero-phase frequency response, $H(\omega)$ [1], is linear with respect to the unknown polynomial coefficients. This fact makes it possible to solve the *Optimization Problem* (stated later) by using linear programming. In the practical optimization problem the zero-phase frequency, $H(\omega)$, is in the following form: $H(\omega) = \sum_{m=1}^{M} \sum_{r=1}^{L} H(\omega, m, r)$, where

$$
H(\omega, m, r) = \begin{cases} a_m^{(L)}(r)(\widetilde{N}_m)^r + \\ \sum_{n=1}^{\widetilde{N}_m} a_m^{(L)}(r)(\widetilde{N}_m - n)^r 2\cos(n\omega), \\ \sum_{n=1}^{\widetilde{N}_m - 1} a_m^{(L)}(r)(\widetilde{N}_m - 1 - n)^r 2\cos((n-0.5)\omega), \\ \sum_{n=1}^{\widetilde{N}_m} a_m^{(L)}(r)(\widetilde{N}_m - n)^r 2\sin(n\omega), \\ \sum_{n=1}^{\widetilde{N}_m - 1} a_m^{(L)}(r)(\widetilde{N}_m - 1 - n)^r 2\sin((n-0.5)\omega), \end{cases} \tag{24}
$$

where $\widetilde{N}_m = N - N_m$, for Types 1, 2, 3, and 4 respectively. The main concern, for the optimization problem, are the parameters, i.e., how to choose them so that after applying linear programming, the polynomial coefficients, which make (23) equal to or less than unity, can be found as defined by (24)–(28). The following parameters should be chosen: the number of slices M, the filter order and the common polynomial degree L in each slice and the values N_m, for the mth slice for $m = 1, 2, \ldots, M$, so that the arithmetic complexity is minimized. These parameters are found by a trial-and-error technique. No straightforward way to find a unique optimum combination of parameters has been found. Some advice on how to choose the overall filter order can be found from the fact that the filter order is slightly greater than the minimum order of a direct-form FIR filter with the same design criteria. Based on the results in Section VI, the general basic rule, if the passband and stopband edges are divided (multiplied) by a factor, the $N_m s$ should be multiplied (divided) by the same factor, could be applied.

Optimization Problem:

1) Given ω_p, ω_s, δ_p and δ_s.
2) Choose the overall filter order and the number of slices M, the common polynomial degree L in each slice and the values N_m, for the mth slice for $m = 1, 2, \ldots, M$.
3) Form the polynomials for each slice.
4) Form the following weighted error function:

$$
E(\omega) = |W(\omega)||H(\omega) - D(\omega)|, \tag{25}
$$

with respect to the unknown polynomial coefficients $a_k^{(m)}$. Here,

$$
D(\omega) = \begin{cases} D_p(\omega) & \text{for } \omega \in [0,\ \omega_p] \\ 0 & \text{for } \omega \in [\omega_s,\ \pi], \end{cases} \tag{26}
$$

with

$$D_p(\omega) = \begin{cases} 1 & \text{for Types 1, and 2} \\ \omega & \text{for Types 3, and 4,} \\ & \text{i.e., for a differentiator design} \end{cases} \quad (27)$$

and

$$W(\omega) = \begin{cases} 1/\delta_p & \text{for } \omega \in [0,\ \omega_p] \\ 1/\delta_s & \text{for } \omega \in [\omega_s,\ \pi]. \end{cases} \quad (28)$$

5) Solve the unknown polynomial coefficients $a_k^{(m)}$s with a linear programming algorithm.
6) If the polynomial coefficients $a_k^{(m)}$s make the quantity in (23) equal to or less than unity, the design criteria are automatically met and the filter is successfully designed, otherwise continue from step 2.

V. Implementation Structures

This section shows how to generate an arbitrary piecewise-polynomial impulse response for each of the four linear-phase FIR filter types.

The main idea when generating the overall impulse response is to reduce the order and the length of the polynomials by one at every step until the polynomial order of zero is reached. All the types have their own implementation structures. In order to achieve the desired goal, first an implementation block is derived for a single branch. Based on this structure an overall implementation structure is generated.

Based on the design technique proposed in Section III, the linear-phase FIR filter transfer functions for Types 1 to 4 are of the form:

$$H(z) \equiv H^{(0)}(z) = \begin{cases} h^{(0)}(N)z^{-N} + \sum_{n=0}^{N-1} h^{(0)}(n)[z^{-n} + z^{-(2N-n)}] \\ \sum_{n=0}^{N-1} h^{(0)}(n)[z^{-n} + z^{-(2N-1-n)}] \\ \sum_{n=0}^{N-1} h^{(0)}(n)[z^{-n} - z^{-(2N-n)}] \\ \sum_{n=0}^{N-1} h^{(0)}(n)[z^{-n} - z^{-(2N-1-n)}], \end{cases} \quad (29)$$

respectively, where N is an integer and $h^{(0)}(n)$ is given by

$$h^{(0)}(n) = p^{(L)}(n) = \sum_{r=0}^{L} a^{(L)}(r)n^r \quad \text{for} \quad 0 \le n \le \widehat{N}_0 \quad (30)$$

with

$$\widehat{N}_0 = \begin{cases} N & \text{for Type 1} \\ N-1 & \text{for Types 2, 3, and 4.} \end{cases} \quad (31)$$

In (30), $p^{(L)}(n)$ is an Lth-degree polynomial in n. Hence, there exist $L+1$ unknowns in the above transfer functions.

For Types 1, 2, 3, and 4, the impulse responses satisfy $h^{(0)}(2N-n) = h^{(0)}(n)$ for $n = 0,1,\dots,2N$; $h^{(0)}(2N-1-n) = h^{(0)}(n)$ for $n = 0,1,\dots,2N-1$; $h^{(0)}(2N-n) = -h^{(0)}(n)$ for $n = 0,1,\dots,2N$; and $h^{(0)}(2N-1-n) = -h^{(0)}(n)$ for $n = 0,1,\dots,2N-1$; respectively. Note that due to the above symmetry condition, $h^{(0)}(N) = 0$ for Type 3. For Types 1 and 2 (Types 3 and 4), the impulse response is symmetrical (antisymmetrical). Furthermore, for Types 1 and 3 (Types 2 and 4), the filter order is $2N$ ($2N-1$), where the order is even (odd).

In order to implement the above $H(z)$ efficiently by using accumulators for the four types, starting with

$$p^{(L)}(n) = \sum_{r=0}^{L} a^{(L)}(r)n^r, \quad (32)$$

the following polynomials are determined recursively for $k = 1,2,\dots,L$:

$$p^{(L-k)}(n) = p^{(L-k+1)}(n+1) - p^{(L-k+1)}(n) = \sum_{r=0}^{L-k} a^{(L-k)}(r)n^r, \quad (33)$$

where

$$a^{(L-k)}(r) = \sum_{j=r+1}^{L-k+1} \binom{j}{r} a^{(L-k+1)}(r). \quad (34)$$

The first step is to express these transfer functions in the following form:

$$H^{(0)}(z) = E^{(1)}(z)/(1-z^{-1}), \quad (35)$$

where

$$E^{(1)}(z) = (1-z^{-1})H^{(0)}(z). \quad (36)$$

This $E^{(1)}(z)$ can be written as

$$E^{(1)}(z) = \sum_{n=0}^{\widetilde{N}} e^{(1)}(n)z^{-n}, \quad (37)$$

where

$$\widetilde{N} = \begin{cases} 2N+1 & \text{for Types 1 and 3} \\ 2N & \text{for Types 2 and 4} \end{cases} \quad (38)$$

and

$$e^{(1)}(n) = h^{(0)}(n) - h^{(0)}(n-1) \quad \text{for} \quad n = 0,1,\dots,\widetilde{N}. \quad (39)$$

Hereinafter, only Types 1 and 2 are discussed because by deriving the implementation structure for Types 1 and 2 also structures for Types 3 and 4 are automatically obtained. This is because Types 1 and 4 as well as Types 2 and 3 alternate when forming the structure as seen later on. However, the implementation coefficients are given for all four types, because the final structures are presented for all four types. Based on the impulse responses of the original transfer functions, the resulting impulse responses of $E^{(1)}(z)$ for Types 1 and 2 can be expressed, after some manipulations, in terms of the polynomials $p^{(L)}(n)$ and $p^{(L-1)}(n)$, given by (32)–(34), as follows:

$$e^{(1)}(n) = \begin{cases} p^{(L)}(0), & n = 0 \\ p^{(L-1)}(n-1), & n = 1,2,\dots,N \\ -p^{(L-1)}(2N-n), & n = N+1, N+2, \dots, 2N \\ -p^{(L)}(0), & n = 2N+1 \\ 0, & \text{otherwise,} \end{cases} \quad (40)$$

$$e^{(1)}(n) = \begin{cases} p^{(L)}(0), & n = 0 \\ p^{(L-1)}(n-1), & n = 1, 2, \ldots, N-1 \\ 0, & n = N \\ -p^{(L-1)}(2N-1-n), & n = N+1, N+2, \\ \quad \ldots, 2N-1 & \\ -p^{(L)}(0), & n = 2N \\ 0, & \text{otherwise,} \end{cases} \tag{41}$$

respectively.

To be able to generate the desired overall filter structure for all the four filter types using accumulators, it is desirable to first express, for the reasons to be explained later, the above impulse responses in the following form:

$$e^{(1)}(n) = h^{(1)}(n-1) + f^{(1)}(n) \quad \text{for} \quad n = 0, 1, \ldots, \widetilde{N}. \tag{42}$$

Secondly, after some manipulations, it is favorable to express $h^{(1)}(n)$ and $f^{(1)}(n)$ for Types 1 and 2, in terms of the polynomials $p^{(L)}(n)$ and $p^{(L-1)}(n)$ as follows:

$$h^{(1)}(n) = \begin{cases} p^{(L-1)}(n), & n = 0, 1, \ldots, N-1 \\ -p^{(L-1)}(2N-1-n), & n = N, N+1, \\ & \quad \ldots, 2N-1 \\ 0, & \text{otherwise} \end{cases} \tag{43}$$

and

$$f^{(1)}(n) = \begin{cases} p^{(L)}(0), & n = 0 \\ -p^{(L)}(0), & n = 2N+1 \\ 0, & \text{otherwise;} \end{cases} \tag{44}$$

as well as

$$h^{(1)}(n) = \begin{cases} p^{(L-1)}(n), & n = 0, 1, \ldots, N-2 \\ 0, & n = N-1 \\ -p^{(L-1)}(2N-2-n), & n = N, N+1, \\ & \quad \ldots, 2N-2 \\ 0, & \text{otherwise} \end{cases} \tag{45}$$

and

$$f^{(1)}(n) = \begin{cases} p^{(L)}(0), & n = 0 \\ -p^{(L)}(0), & n = 2N \\ 0, & \text{otherwise;} \end{cases} \tag{46}$$

respectively.

The above equations make it possible to express $E^{(1)}(z)$ as

$$E^{(1)}(z) = F^{(1)}(z) + z^{-1}H^{(1)}(z), \tag{47}$$

and the importance and the roles of $H^{(1)}(z)$ and $F^{(1)}(z)$ are discussed next.

First, $H^{(1)}(z)$ is the polynomial part of the transfer function and can be expressed as

$$H^{(1)}(z) = \begin{cases} \sum_{n=0}^{N-1} h^{(1)}(n)\left[z^{-n} - z^{-(2N-1-n)}\right] \\ \sum_{n=0}^{N-2} h^{(1)}(n)\left[z^{-n} - z^{-(2N-2-n)}\right] \end{cases} \tag{48}$$

for Types 1 and 2, respectively. In (48), $h^{(1)}(n)$ is given in terms of the $(L-1)$th-order polynomial $p^{(L-1)}(n)$, as given by (43) and (45) with (32)–(34).

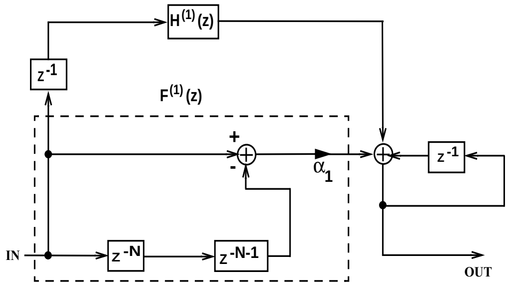


Fig. 4. Implementation block of the overall transfer function in terms of $F^{(1)}(z)$ and $H^{(1)}(z)$ for the structure of Type 1 filters. In this structure $\alpha_1 = p^{(L)}(0)$.

When comparing these first two steps in derivation in (29), and (48), the following observations can be made. First, it can be seen that $H^{(1)}(z)$ for Types 1 and 2 corresponds to Types 4 and 3 in $H^{(0)}(z)$. So Types 1 and 4 as well as Types 2 and 3 alternate when forming the structures. Secondly, both the filter order and the order of the polynomial for the impulse responses that follow are reduced by one. This makes it possible, at the second step by means of the second accumulator, to reduce the order and the degree of $H^{(1)}(z)$ in a similar manner. This will be discussed after explaining the role of $F^{(1)}(z)$. The key idea is to give $F^{(1)}(z)$ for the filter Types 1 and 2 in such a form that $H^{(1)}(z)$ is of a form similar to that of the original transfer function $H^{(0)}(z)$, as given by (29)–(31).

$F^{(1)}(z)$ is the non-polynomial part, i.e., it does not have a polynomial form and can be expressed as

$$F^{(1)}(z) = \begin{cases} p^{(L)}(0)[1 - z^{-(2N+1)}] \\ p^{(L)}(0)[1 - z^{-2N}] \end{cases} \tag{49}$$

for Types 1 and 2, respectively. It contains the remainder terms of $E^{(1)}(z)$ so that $H^{(1)}(z)$ can be expressed in the above form. The implementation of the overall transfer function in terms of $F^{(1)}(z)$ and $H^{(1)}(z)$ is shown in Fig. 4 for Type 1 FIR filters.

In order to further decrease the polynomial order, the second step is to add the second accumulator in the structure of Fig. 4 by expressing $H^{(1)}(z)$ as

$$H^{(1)}(z) = E^{(2)}(z)/(1 - z^{-1}), \tag{50}$$

where, after a derivation similar to that performed for the first step, $E^{(2)}(z)$ can be written in the following form:

$$E^{(2)}(z) = (1 - z^{-1})H^{(1)}(z) \equiv F^{(2)}(z) + z^{-1}H^{(2)}(z). \tag{51}$$

Here,

$$H^{(2)}(z) = \begin{cases} h^{(2)}(N-1)z^{-(N-1)} + \\ \quad \sum_{n=0}^{(N-1)-1} h^{(2)}(n)[z^{-n} + z^{-(2(N-1)-n)}] \\ \quad \sum_{n=0}^{(N-1)-1} h^{(2)}(n)[z^{-n} + z^{-(2(N-1)-1-n)}] \end{cases} \tag{52}$$

for Types 1 and 2, respectively. $H^{(2)}(z)$ is the polynomial part of the transfer function and and $F^{(2)}(z)$ is the non-polynomial

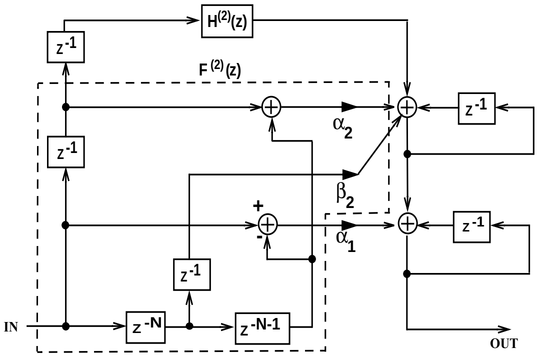


Fig. 5. Implementation block of the overall transfer function in terms of $F^{(1)}(z)$, $F^{(2)}(z)$, and $H^{(2)}(z)$ for the structure of Type 1 filters. In this structure $\alpha_1 = p^L(0)$, $\alpha_2 = p^{(L-1)}(0)$, and $\beta_2 = -p^{(L-1)}(N) - p^{(L-1)}(N-1)$.

part and can be expressed as

$$F^{(2)}(z) = \begin{cases} p^{(L-1)}(0)[1+z^{-2N}]- \\ \quad [p^{(L-1)}(N)+p^{(L-1)}(N-1)]z^{-N} \\ p^{(L-1)}(0)[1+z^{-(2N-1)}]- \\ \quad p^{(L-1)}(N-1)z^{-(N-1)}[1+z^{-1}] \end{cases} \tag{53}$$

for Types 1 and 2, respectively.

In $H^{(2)}(z)$, $h^{(2)}(n)$ is given in terms of the $(L-2)$th-order polynomial $p^{(L-2)}(n)$, as given by (32)–(34) as follows:

$$h^{(2)}(n) = p^{(L-2)}(n) = \sum_{r=0}^{L-2} a^{(L)}(r)n^r \quad \text{for} \quad 0 \le n \le \widehat{N}_2 \tag{54}$$

with

$$\widehat{N}_2 = \begin{cases} N-1 & \text{for Type 1} \\ (N-1)-1 & \text{for Type 2}. \end{cases} \tag{55}$$

The above $H^{(2)}(z)$ is thus similar to $H^{(0)}(z)$ except that for $H^{(2)}(z)$, both the filter order and the degree of the polynomial for the impulse responses that follow are decreased by two.

The resulting structure in terms of $F^{(2)}(z)$ and $H^{(2)}(z)$ for Type 1 filters is seen in Fig. 5. The above process is continued to generate $F^{(k)}(z)$ and $H^{(k)}(z)$ for $k = 3, 4, \ldots, L+1$. As a result of the overall procedure, the original transfer function can be expressed as

$$H^{(0)}(z) = [F^{(1)}(z) + z^{+1}H^{(1)}(z)]/[1-z^{-1}] \tag{56}$$

$$H^{(1)}(z) = [F^{(2)}(z) + z^{-1}H^{(2)}(z)]/[1-z^{-1}] \tag{57}$$

$$\vdots \quad \vdots \quad \vdots$$

$$H^{(L-1)}(z) = [F^{(L)}(z) + z^{-1}H^{(L)}(z)]/[1-z^{-1}] \tag{58}$$

$$H^{(L)}(z) = F^{(L+1)}(z)/[1-z^{-1}], \tag{59}$$

where $F^{(k)}(z)$ for $k = 1, 2, \ldots, L+1$ are the transfer functions to be implemented. The first L transfer functions for Types 1 and 2 are given respectively by

$$F^{(2l+1)}(z) = \begin{cases} p^{(L-2l))}(0)[1-z^{-(2(N-l)+1)}] \\ p^{(L-2l)}(0)[1-z^{-2(N-l)}] \end{cases} \tag{60}$$

for $l = 0, 1, \ldots, \lfloor (L+1)/2 \rfloor - 1$ and by

$$F^{(2l+2)}(z) = \begin{cases} p^{(L-2l-1)}(0)[1+z^{-2(N-l)}]- \\ \quad [p^{(L-2l-1)}(N-l)+p^{(L-2l-1)}((N-l)-1)]z^{-(N-l)} \\ p^{(L-2l-1)}(0)[1+z^{-(2(N-l)-1)}] - p^{(L-2l-1)}((N-l)-1) \\ \quad \times z^{-((N-l)-1)}[1+z^{-1}] \end{cases} \tag{61}$$

for $l = 0, 1, \ldots, \lfloor L/2 \rfloor - 1$.

Alternatively, these transfer functions can be expressed as

$$F^{(2l+1)}(z) = \begin{cases} \alpha_{2l+1}[1-z^{-(2(N-l)+1)}] \\ \alpha_{2l+1}[1-z^{-2(N-l)}] \end{cases} \tag{62}$$

for $l = 0, 1, \ldots, \lfloor (L+1)/2 \rfloor - 1$ and as

$$F^{(2l+2)}(z) = \begin{cases} \alpha_{2l+2}[1+z^{-2(N-l)}] + \beta_{2l+2}z^{-(N-l)} \\ \alpha_{2l+2}[1+z^{-(2(N-l)-1)}] + \beta_{2l+2}z^{-((N-l)-1)}[1+z^{-1}] \end{cases} \tag{63}$$

for $l = 0, 1, \ldots, \lfloor L/2 \rfloor - 1$. Next, the structure coefficients, α_ks and β_ks, are given for all the four types as mentioned earlier. Here, α_ks, for $k = 1, 2, \ldots, L$, are related to the corresponding polynomials, $p^{(L-1+k)}(n)$s, as given by (32), (33) and (34), through

$$\alpha_k = p^{(L+1-k)}(0) = a^{(L-k+1)}(0), \; k = 1, 2, \ldots, L, \tag{64}$$

whereas non-zero values of β_k's for Types 1, 2, 3, and 4 are given by

$$\beta_{2k} = -p^{(L-2k+1)}(N+1-k) - p^{(L-2k+1)}(N-k), \quad k = 1, 2, \ldots, \lfloor L/2 \rfloor, \tag{65a}$$

$$\beta_{2k} = -p^{(L-2k+1)}(N-k), \; k = 1, 2, \ldots, \lfloor L/2 \rfloor, \tag{65b}$$

$$\beta_{2k-1} = -p^{(L-2(k-1))}(N-k), \; k = 1, 2, \ldots, \lfloor (L+1)/2 \rfloor, \tag{65c}$$

and

$$\beta_{2k-1} = -p^{(L-2(k-1))}(N-k) - p^{(L-2(k-1))}((N-k)-1), \quad k = 1, 2, \ldots, \lfloor (L+1)/2 \rfloor, \tag{65d}$$

respectively.

The remaining transfer function $F^{(L+1)}(z)$ for Types 1 and 2 can be expressed as

$$F^{(L+1)}(z) = \begin{cases} p^{(0)}(0)[1-z^{-2(N-L/2)+1}] \\ p^{(0)}(0)[1-z^{-2(N-L/2)}] \end{cases} \tag{66}$$

and

$$F^{(L+1)}(z) = \begin{cases} p^{(0)}(0)[1-2z^{-(N-(L-1)/2)} + z^{-(2N-L+1)}] \\ p^{(0)}(0)[1-z^{-(N-(L-1)/2)}(1+z^{-1})+ \\ \quad z^{-(2N-L)}] \end{cases} \tag{67}$$

for L even and odd, respectively.

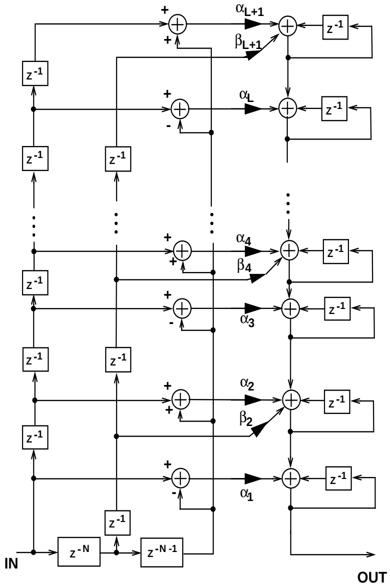


Fig. 6. Implementation block for Type 1 filters.

These transfer functions can be rewritten in the form

$$F^{(L+1)}(z) = \begin{cases} \alpha_{L+1}[1 - z^{-2(N-L/2)+1}] \\ \alpha_{L+1}[1 - z^{-2(N-L/2)}] \end{cases} \tag{68}$$

and

$$F^{(L+1)}(z) = \begin{cases} \alpha_{L+1}[1 + z^{-(2N-L+1)}] + \beta_{L+1}z^{-(N-(L-1)/2)} \\ \alpha_{L+1}[1 + z^{-(2N-L)}] + \\ \quad \beta_{L+1}z^{-(N-(L-1)/2)}(1 + z^{-1}) \end{cases} \tag{69}$$

for L even and odd respectively. Here, $\alpha_{L+1} = p^{(0)}(0)$, whereas for even L,

$$\beta_{L+1} = -p^{(0)}(0) \text{ and } \beta_{L+1} = -2p^{(0)}(0)$$

for Types 3 and 4, respectively, and for odd L,

$$\beta_{L+1} = -2p^{(0)}(0) \text{ and } \beta_{L+1} = -p^{(0)}(0)$$

for Types 1 and 2, respectively.

An efficient implementation block for the above transfer function for Type 1 is shown in Fig. 6 for odd L. For even L, the corresponding block is obtained by first discarding the uppermost branch and then replacing L by $L + 1$.

Based on the above structure, a similar structure can be developed for each of the branch filters, $H_m(z)$ described in (2).

To generate the corresponding coefficients, α_ks and β_{2k}s, the impulse response given by (4) is used. The resulting α_ks and β_{2k}s are denoted by $\alpha_k^{(m)}$ and $\beta_{2k}^{(m)}$. An efficient implementation structure for the overall filter is shown in Fig. 10, where $T_m = N_m$ for $m = 1, 2, \ldots, M-1$, $T_{M1} = N - N_M$ and $T_{M2} = N - N_M + 1$. The role of the T_ms is to keep the center of the symmetry of the blocks at the center of the overall filter in order to maintain the shape of the linear-phase type.

Furthermore, the β_{2k}s are the sums of the corresponding $\beta_{2k}^{(m)}$s. In order to keep the overall diagram simple, the coefficients $\alpha_k^{(m)}$s have been drawn twice. In the practical implementation, the overall number of multipliers can be reduced by first adding [subtracting] the inputs of the left-hand side $\alpha_k^{(m)}$s and the right-hand side $\alpha_k^{(m)}$s [$-\alpha_k^{(m)}$s] and then multiplying the result by $\alpha_k^{(m)}$s.

Based on the above formulas the implementation structures for Types 2, 3, and 4 are shown in Figs. 11, 12, and 13, respectively, where $T_m = N_m$ for $m = 1, 2, \ldots, M-1$, $T_{M1} = N - N_M$, and $T_{M2} = N - N_M$.

VI. Design Examples

In this section, the properties and efficiency of the proposed filter classes over conventional direct-form FIR filters and IFIR filters [1], [9] are shown by means of examples in narrowband cases. The following two design criteria are considered: The passband and the stopband edges are
Case 1: $\omega_p = 0.025\pi$, $\omega_s = 0.05\pi$
Case 2: $\omega_p = 0.0063\pi$, $\omega_s = 0.0125\pi$
and both with $\delta_p = 0.01$, and $\delta_s = 0.001$ (60 dB attenuation).

For Case 1, the given criteria are met by a filter of order 220, $M = 5$, $L = 3$, $N_1 = 0$, $N_2 = 23$, $N_3 = 50$, $N_4 = 81$, and $N_5 = 98$ as shown in Table I.

This filter thus consists of five slices with the polynomial degree, $L = 3$ in each slice. In this case, the number of unknowns required in the optimization is 20. The number of multipliers is 22 in the practical implementation. The magnitude and the impulse response for the optimized filter are shown in Fig. 7(a). For comparison, also cases with polynomial degree $L = 2$ and $L = 4$ are shown in Table I. With polynomial degree two, the minimum number of unknowns is 30. For

TABLE I
Filter Parameters for Case 1 Design Criteria

Polynomial degree L	Number of slices M	Number of unknowns	Filter order	Slices, N_ms
3	5	20	220	0, 23, 50, 81, 98.
2	10	30	220	0, 10, 21, 31, 43, 53, 65, 76, 87, 98.
4	4	20	220	0, 31, 71, 98.

TABLE II
Filter Parameters for Case 2 Design Criteria

Polynomial degree L	Number of slices M	Number of unknowns	Filter order	Slices, $N_m s$
3	8	32	870	0, 87, 136, 195, 252, 319, 355, 413.

Case 2, the given criteria are met by a filter of order 870, $M = 8$, $L = 3$, and $N_1 = 0$, $N_2 = 87$, $N_3 = 136$, $N_4 = 195$, $N_5 = 252$, $N_6 = 319$, $N_7 = 355$, and $N_8 = 413$ as shown in Table II. This filter thus consists of eight slices with the polynomial degree, $L = 3$ in each slice. The number of unknowns required in the optimization is 32. The magnitude and the impulse response for the optimized filter are shown in Fig. 7(b).

TABLE III

FILTER PARAMETERS OF REFERENCE DESIGNS FOR CASE 1 DESIGN CRITERIA

Filter	Order of $F(z^L)G(z)$	Filter Order	Number of Multipliers	Interpolation Factor L
Direct-form FIR	-	215	108	-
IFIR	26+19		24	8

TABLE IV

FILTER PARAMETERS OF REFERENCE DESIGNS FOR CASE 2 DESIGN CRITERIA

Filter	Order of $F(z^L)G(z)$	Filter Order	Number of Multipliers	Interpolation Factor L
Direct-form FIR	-	863	432	-
IFIR	50+34		44	16

Tables III and IV show designs with corresponding criteria for direct-form FIR and IFIR filters.

The IFIR filter in Table III, i.e., $H(z) = F(z^L)G(z)$, needs altogether 24 multipliers compared to our design, which needs 22 multipliers in a practical implementation.

The IFIR filter in Table IV needs altogether 44 multipliers compared to our design, which needs 34 multipliers in a practical implementation.

The following observation can be made for both cases: the arithmetic complexity becomes the lesser, the narrower the transition band gets compared to direct-form design, i.e., the reduction in the number of multipliers is from 82 to 92 percent. The downside is that the filter length increases. Compared to IFIR filter designs the arithmetic complexity is slightly less.

As also shown in Table I, if the polynomial degree is decreased, the number of slices has to be increased in order to meet the given design criteria.

Next, the properties and efficiency of the proposed filter class for Type 3 and 4 FIR filters are shown by means of an example in a narrowband differentiator case. Details of the example are shown in Table V.

Consider the design of narrowband differentiators with the specifications: $\omega_p = 0.025\pi$, $\omega_s = 0.05\pi$, $\delta_p = 0.01$, and $\delta_s = 0.001$. For the proposed approach, the given criteria

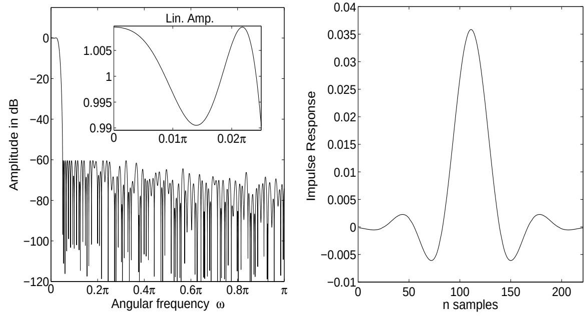
TABLE V

A NARROWBAND DIFFERENTIATOR WITH CASE 1 DESIGN CRITERIA

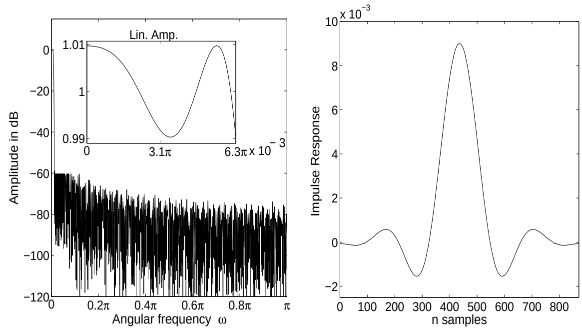
Polynomial degree, L	Number of slices, M	Number of unknowns	Filter order	Overall ripple value
3	5	20	332	9.13964×10^{-4}

are met by a differentiator of order 332, $M = 5$, $L = 3$, $N_1 = 0$, $N_2 = 37$, $N_3 = 74$, $N_4 = 111$ and $N_5 = 148$. This FIR differentiator thus consists of five uniformly-spaced slices with the polynomial degree, $L = 3$ in each slice.

In this case, the number of unknowns required is 20 in the optimization and 22 in the practical implementation. The magnitude response for the optimized differentiator is shown in Fig. 8.



(a)



(b)

Fig. 7. Design examples of proposed linear-phase FIR filters: (a) Case 1, (b) Case 2.

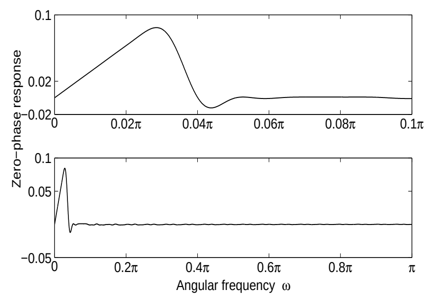


Fig. 8. A design example of the proposed narrowband lowpass FIR differentiator with Case 1 design criteria.

The corresponding direct-form FIR differentiator has a minimum order of 223, whereby the number of coefficients needed is 112 when exploiting coefficient symmetry compared to our design, which requires only 22 coefficients in the practical implementation.

VII. PRACTICAL IMPLEMENTATION ASPECTS

In the implementation structures in Figs. 10–13 there are accumulators having a pole at $z = 1$ at the end of the structures. There are ways to avoid possible problems, which this fact may create. The following property of two's complement arithmetic with a fixed-point representation is utilized: If there are several values to be added and/or subtracted, overflows

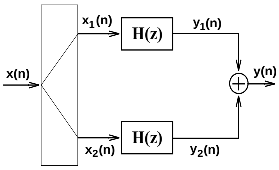


Fig. 9. Practical implementation based on switching and resetting, where a demultiplexer is used to decompose $x(n)$ into two signals $x_1(n)$ and $x_2(n)$.

are allowed if the final result is in the range $[-1, 1 - 2^{-b}]$ (b is the number of fractional bits). First, before starting to use the structures of Figs. 10–13, the state variables should be reset. Secondly, pole-zero cancellation should be done in the overall structure by quantizing the polynomial coefficients before deriving the coefficients in the structures. Third, to avoid overflows, worst-case scaling is required, i.e., the sum of the absolute values of the impulse response has to be less than or equal to unity. This implies calculating the sum of the absolute values of the impulse response of the overall filter and dividing the input signal or the coefficients by this sum. Finally, the filter output should be multiplied by this sum. There are no overflows in the structures provided that after multiplication of the data samples no roundings or truncations are performed [6]. The actual rounding or truncation is performed after accumulators before multiplying by the above-mentioned sum.

These structures suffer from the following drawbacks. First, the word-length is double in the accumulators. Secondly, the structure does not automatically recover from temporary data errors in the accumulators. Such data corruption can be caused by overflows, power surges, system start-up, and random errors in general. These problems can be avoided by using a parallel structure, where all the data registers are doubled, following the principle proposed in [17]. If the overall filter order is N, the input sequence is split into sets of N data samples so that the first set, third set and so on are fed to the upper branch, whereas the remaining sets are fed to the lower branch. The performance of the overall system is guaranteed by resetting the state variables in the upper (lower) branch at time instants $n = 2\rho N - 1$ ($n = (2\rho - 1)N - 1$), where ρ is an integer. This structure is shown in Fig. 9. In this case, rounding or truncation is allowed after the multiplications by the filter coefficients.

VIII. Conclusion

A straightforward approach to design linear-phase FIR filters with a piecewise-polynomial impulse response for all the four types has been proposed. It is also shown by means of design examples that the relative advantage of the resulting filters in arithmetic complexity increases, the narrower the transition band gets, compared to other narrowband FIR filter designs. Examples have been given and comparisons have been made to direct-form and IFIR filters in order to show the benefits of the proposed FIR filter classes. Computationally efficient recursive implementation structures were derived for all four types of linear-phase piecewise-polynomial FIR filters. Formulas for implementation coefficients based on the optimized polynomial coefficients have been proposed. Some practical implementation considerations, e.g. regarding the practical implementation structure, overflows, and scaling, have been given. The future work includes finding the formulas to calculate the optimization parameters such as the filter order, and how to choose the slices N_ms. It is found that the best polynomial degree to be chosen is three because with higher polynomial degrees there may be more coefficients in practical implementation, i.e., more β-coefficients. If the polynomial degree is higher than four the complexity of the practical implementation increases; the number of unknowns may become too high to reduce the arithmetic complexity in both optimization and implementation compared to IFIR filters.

References

[1] T. Saramäki, “Finite impulse response filter design,“ in *Handbook for Digital Signal Process.*, Editors S. K. Mitra and J.F. Kaiser, New York: Wiley, 1993, Chapter 4, pp. 155–277.

[2] G. F. Boudreaux and T. W. Parks, “Thinning digital filters: A piecewise-exponential approximation approach,“ *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-31, pp. 105–113, Feb. 1983.

[3] S. Chu and S. Burrus, “Efficient Recursive Realizations of FIR Filters, Part I: The filter Structures,” *Circuits, Syst. and Signal Process.*, vol. 3, no. 1, pp. 2–20, 1984.

[4] S. Chu and S. Burrus, “Efficient Recursive Realizations of FIR Filters, Part II: Design and Applications,” *Circuits, Systems and Signal Process.*, vol. 3, no. 1, pp. 21–57, 1984.

[5] T. G. Campbell and T. Saramäki, “Recursive linear-phase FIR filter structures with piecewise-polynomial impulse response,” *Proc. Sixth Int. Symp. Networks, Systems and Signal Process.*, Zagreb, Yuogoslavia, pp. 16–19, June 1989.

[6] T. Saramäki and O. Vainio, “Structures for generating polynomial responses,” *Proc. 37th Midwest Symp. Circuits Syst. 1994*, vol. 2, pp. 1315–1318, August 1994.

[7] T. Saramäki and S. K. Mitra, “Design and implementation of narrow-band linear-phase FIR filters with piecewise polynomial impulse response,” *Proc. IEEE Int. Symp. Circuits Syst.*. ISCAS ’99, Orlando, FL, vol 3, pp. 456–461, July 1999.

[8] S. K. Mitra, A. Mahalanobis, and T. Saramäki, “A generalized structural subband decomposition of FIR filters and its application in efficient FIR filter design and implementation,” *IEEE Trans. Circuits Syst.II: Analog and Digit. Signal Process.*, vol. 40, pp. 363–374, June 1993.

[9] T. Saramäki, Y. Neuvo and S. K. Mitra, “ Design of computationally efficient interpolated FIR filters,” *IEEE Trans. Circuits Syst.*, vol. 35, no. 1, pp. 70 – 88, Jan. 1988.

[10] O. Vainio, M. Renfors and T. Saramäki, “ Recursive implementation of FIR differentiators with optimum noise attenuation,” *IEEE Trans. Instrum. Meas.*, vol. 46, pp. 1202–1207, Oct. 1997.

[11] B. Kumar and S. C.Dutta Roy, “Coefficients for maximally linear, FIR digital differentiator for low frequencies,“ *Electronics Letters*, vol. 24, pp. 563–565, April 1998.

[12] J. Le Bihan, “Coefficients of FIR digital differentiators and Hilbert Tran sforms for Midband Frequencies,“ *IEEE Trans. Circuits Syst. II: Analog and Digit. Signal Process.*, 1996, vol. 43, pp. 272–274, March 1996.

[13] Ivan W. Selesnick, “Maximally Flat Lowpass Digital Differentiator,” *IEEE Trans. Circuits Syst. II: Analog and Digit. Signal Process.*, vol. 49, pp. 219– 223, March 2002.

[14] M. R. R. Reddy, S. C. Dutta Roy and B. Kumar, “Design of efficient second and higher degree FIR digital differentiators for midband frequencies,“ *Circuits, Devices and Systems, IEEE Proc. G*, vol. 138, pp. 29-33, 1991.

[15] I. R. Khan, M. Okuda and R. Ohba, “Digital differentiators for narrow band applications in the lower to midband range,” *Proc. of the IEEE Int. Symp. Circuits Syst.*, vol. 4, pp. 3721 – 3724, May 2005.

[16] I. R. Khan, and M. Okuda, “Finite-Impulse-Response Digital Differentiators for Midband Frequencies Based on Maximal Linearity Constraints,” *IEEE Trans. Circuits Syst. II: Express Briefs*, vol. 54, no. 3, pp. 242 – 246, March 2007.

[17] T. Saramäki and A. Fam, “Properties and structures of linear-phase FIR filters based on switching and resetting IIR filters,” Proc. *IEEE Int. Symp. Circuits Syst.*, vol. 4, pp. 3271 – 3274, May 1990.

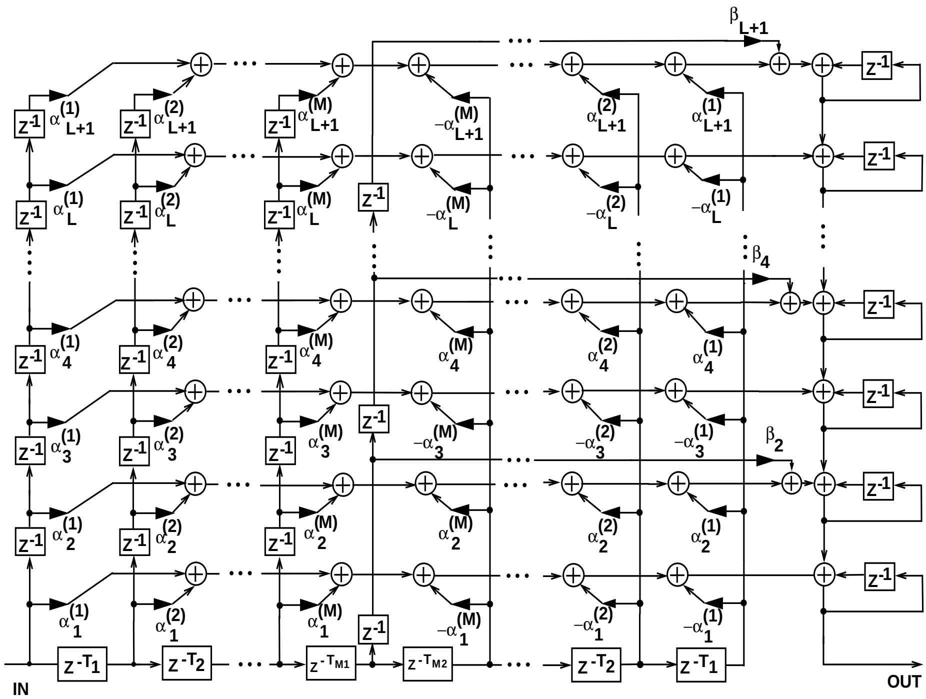


Fig. 10. Implementation structure of Type 1 filters.

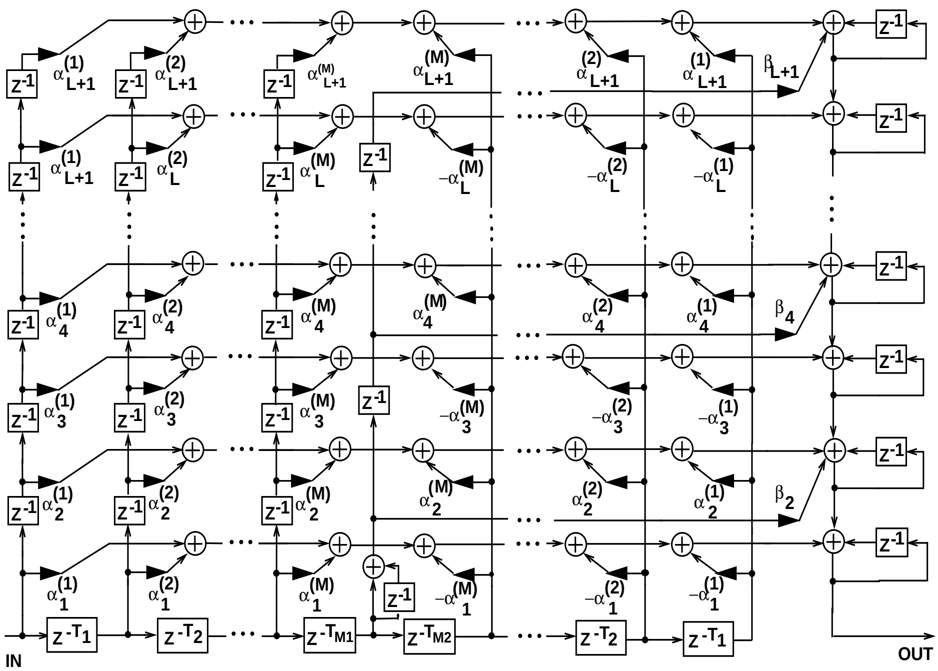


Fig. 11. Implementation structure of Type 2 filters.

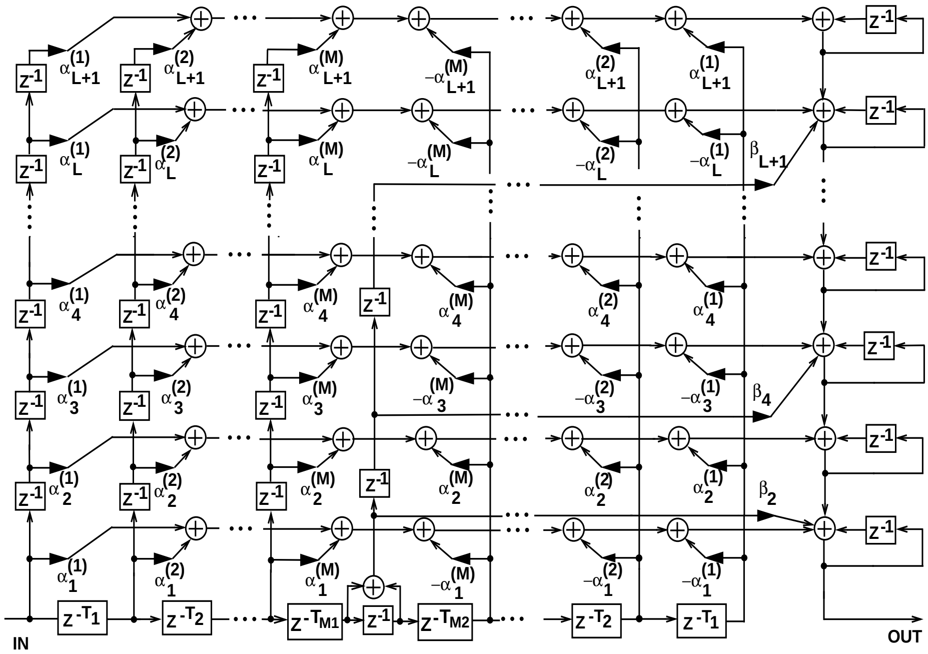


Fig. 12. Implementation structure of Type 3 filters.

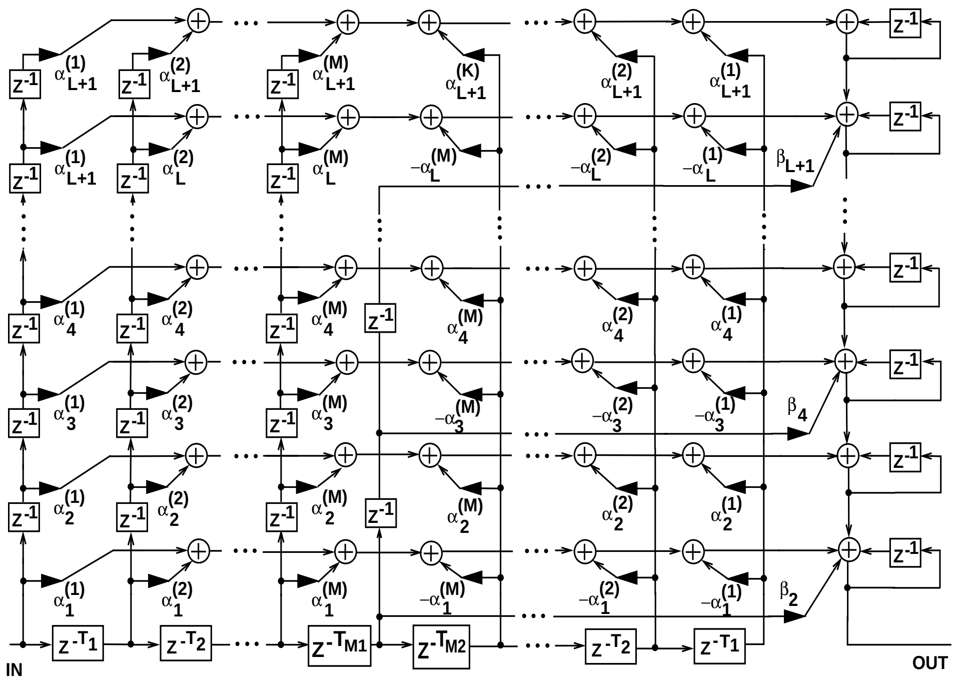
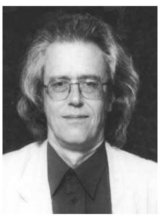


Fig. 13. Implementation structure of Type 4 filters.

Raija Lehto studied mathematics and computer science at the University of Lund, Lund, Sweden, in the 1980s. She also received the degree of Master of Science (Engineering) in information technology from Tampere University of Technology (TUT), Tampere, Finland in 2004. Since 2005 she has been a researcher at the Institute of Signal Processing at TUT and working towards the doctoral degree.



Tapio Saramäki (M'98–SM'01–F'02) was born in Orivesi, Finland, on June 12, 1953. He received the Diploma Engineer (with honors) and Doctor of Technology (with honors) degrees in electrical engineering from the Tampere University of Technology (TUT), Tampere, Finland, in 1978 and 1981, respectively. Since 1977, he has held various research and teaching positions at TUT, where he is currently a Professor of signal processing and a Docent of telecommunications. He is also a Cofounder and a System-Level Designer of VLSI Solution Oy, Tampere, originally specializing in VLSI implementations of Sigma-Delta modulators and analog and digital signal processing algorithms for various applications. He is also the President of Aragit Oy Ltd., Tampere, which was founded by four TUT professors, specializing in various services for the industry, including the application of information technology to numerous applications. In 1982, 1985, 1986, 1990, and 1998, he was a Visiting Research Fellow (Professor) with the University of California, Santa Barbara, in 1987 with the California Institute of Technology, Pasadena, and in 2001 with the National University of Singapore. His research interests are in digital signal processing, especially filter and filterbank design, VLSI implementations, and communications applications, as well as approximation and optimization theories. He has written more than 250 international journal and conference articles and various international book chapters and holds three world-wide used patents.

Dr. Saramäki received the 1987 and 2006 Guillemin-Cauer Award for the Best Paper of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, as well as two other Best Paper awards. In 2004, he received honorary membership of the A. S. Popov Society for Radio-Engineering, Electronics, and Communications (the highest membership grade in the society and the 80th honorary member since 1945) for "great contributions to the development of DSP theory and methods and great contributions to the consolidation of relationships between Russian and Finnish organizations." He is a founding member of the Median-Free Group International. He was an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: ANALOG AND DIGITAL SIGNAL PROCESSING (2000–2001), and is currently an Associate Editor of *Circuits, Systems, and Signal Processing (2003–2008)*. He was also a Distinguished Lecturer of the IEEE Circuits and Systems Society (2002–2003) and the Chairman of the IEEE Circuits and Systems DSP Technical Committee (May 2002–May 2004).

Olli Vainio (S'84-M'88-SM'94) received the Diploma Engineer and Doctor of Technology degrees in electrical engineering from Tampere University of Technology, Tampere, Finland, in 1984 ad 1988, respectively.

He has held research and teaching positions at Tampere University of Technology and the Academy of Finland. In 1986-1987 he was a visiting scholar at the University of California, Santa Barbara, working on high-speed integrated circuit design. He has twice been a visitor at the Royal Institute of Technology, Stockholm, Sweden. In 2001-2002, he was a senior scientist of the Academy of Finland. He is currently a professor of computer systems engineering at Tampere University of Technology. He is also a docent of microelectronics at Lappeenranta University of Technology, Lappeenranta, Finland. His research interests are in computationally efficient digital filters, low-power DSP implementations, and asynchronous circuits.

Publication-III

R. Lehto, T. Saramäki and O. Vainio, Synthesis of wideband linear-phase FIR filters with a piecewise-polynomial-sinusoidal Impulse Response, accepted 1.12.2008 to appear in *Circuits, Systems and Signal Processing*, Birkh äuser Boston.

Circuits Syst Signal Process (2010) 29: 25–50
DOI 10.1007/s00034-009-9111-x

LOW POWER DIGITAL FILTERS

Synthesis of Wideband Linear-Phase FIR Filters with a Piecewise-Polynomial-Sinusoidal Impulse Response

Raija Lehto · Tapio Saramäki · Olli Vainio

Received: 18 August 2008 / Revised: 28 October 2008 / Published online: 20 June 2009

Abstract A method is presented to synthesize wideband linear-phase finite-impulse-response (FIR) filters with a piecewise-polynomial-sinusoidal impulse response. The method is based on merging the earlier synthesis scheme proposed by the authors to design piecewise-polynomial filters with the method proposed by Chu and Burrus. The method uses an arbitrary number of separately generated center coefficients instead of only one or none as used in the method by Chu–Burrus. The desired impulse response is created by using a parallel connection of several filter branches and by adding an arbitrary number of center coefficients to form it. This method is especially effective for designing Hilbert transformers by using Type 4 linear-phase FIR filters, where only real-valued multipliers are needed in the implementation. The arithmetic complexity is proportional to the number of branches, the common polynomial order for each branch, and the number of separate center coefficients. For other linear-phase FIR filter types the arithmetic complexity depends additionally on the number of complex multipliers. Examples are given to illustrate the benefits of this method compared to the frequency-response masking (FRM) technique with regard to reducing the number of coefficients as well as arithmetic complexity.

Some parts of this paper were presented at the 2007 IEEE International Symposium on Circuits and Systems, New Orleans, USA and at the 2008 IEEE International Symposium on Circuits and Systems, Seattle, USA. This work was supported by the Academy of Finland, project Project No. 213462 (Finnish Centre of Excellence Program (2006–2011)).

R. Lehto (✉) · T. Saramäki
Department of Signal Processing, Tampere University of Technology, P.O. Box 553, 33101 Tampere, Finland
e-mail: raija.lehto@tut.fi

T. Saramäki
e-mail: tapio.saramaki@tut.fi

O. Vainio
Department of Computer Systems, Tampere University of Technology, 33101 Tampere, Finland
e-mail: olli.vainio@tut.fi

Keywords Filter structures · Finite-impulse-response (FIR) filters · Linear-phase filters · Linear programming · Piecewise polynomial · Polynomials

1 Introduction

Finite-impulse-response (FIR) filters have many favorable properties, e.g. they can be designed with an exact linear phase, which means that no phase distortion is caused in the signal during the filtering operation and both the output noise due to multiplication round-off errors and the sensitivity to variations in filter coefficients are low. Therefore, FIR filters are often preferred over their infinite impulse response (IIR) counterparts, even though their main drawback is a higher number of multipliers needed in conventional implementations when a narrow transition bandwidth is required [18]. An efficient approach to overcome the above-mentioned problem for wideband filters is to synthesize linear-phase FIR filters so that their impulse response is piecewise-polynomial-sinusoidal and the implementation is done using recursive structures [4, 5].

This paper proposes a synthesis scheme for linear-phase wideband FIR filters with a piecewise-polynomial-sinusoidal impulse response. This method is suitable for the design and implementation of Hilbert transformers as well as wideband frequency-selective filters due to the sinusoidal character of the approach. Hilbert transformers are one of the very important special classes of FIR filters and they are used very widely in many signal processing applications such as telecommunications, speech processing, image processing, medical imaging, and medical signal processing, e.g. biological time series [1–3, 6, 8, 17, 22, 24]. The synthesis scheme proposed in this paper is partly based on the previous work done by the authors [9] to design linear-phase FIR filters with a piecewise-polynomial impulse response and on merging it with and modifying the method proposed by Chu and Burrus [4, 5] for the synthesis scheme of wideband FIR filters. Chu and Burrus first designed what is called an envelope filter with a piecewise-polynomial impulse response, for which the design scheme of the authors [9] is very efficient. Secondly, the coefficient values of this envelope filter are modified by multiplying them with a sinusoidal function depending on the linear-phase FIR filter type and by adding at least one center coefficient, which leads to a wideband FIR filter.

In the case of Type 1 lowpass wideband filters, the modification is carried out as follows. A $2N$th-order Type 3 envelope filter with a piecewise-polynomial impulse response is created by using the scheme proposed by the authors in [9]. The resulting impulse response is multiplied by a sinusoidal function $\sin[(n-N)\omega_c]$, where ω_c is roughly the angular frequency at which the zero-phase response of the overall filter takes on the value of half. This leads to the $2N$th-order Type 1 linear-phase filter. The coefficient values of the envelope filters of a $2N$th-order Type 4 Hilbert transformer and a $2N-1$th-order Type 3 Hilbert transformer do not have to be modified. Type 3 and 4 Hilbert transformers have one very attractive relationship as follows. A Hilbert transformer with an even order, i.e., $2N$th-order Type 3, can be derived from an odd-order Hilbert transformer, i.e., Nth-order Type 4 Hilbert transformer, by adding one zero-valued impulse response sample between each two impulse response samples of a Type 4 Hilbert transformer. This is equivalent to replacing each z^{-1} in the transfer function of the odd-order Hilbert transformer by z^{-2}. This results in a $2N$th-order Hilbert transformer, which has only $N+1$ nonzero impulse response values instead of $2N+1$. Thereby, a Type 3 Hilbert transformer derived as above is preferred for the design.

Because of the importance of center coefficients in FIR filters, the approach presented in this paper uses an arbitrary number of separately generated center coefficients instead of only one or none as used by Chu and Burrus [4, 5]. The implementation structures proposed earlier by the authors can be used with a modification; in fact, the implementation structure is modified to be complex for Type 1 and a conventional FIR filter is added for all types in order to form the center of the impulse response.

Other approaches have also been developed to reduce the arithmetic complexity of wideband FIR filters, especially when a narrow transition bandwidth is required. One of the most efficient techniques is the one-stage or multistage frequency-response masking (FRM) technique [10, 20]. This technique was originally introduced by Lim [10] and utilized by Saramäki and Lim [20] by using the Remez multiple-exchange algorithm. Several authors have been using this technique for filters requiring a very narrow transition bandwidth [7, 11–13, 15, 16, 19, 21, 24]. First, one of the advantages of using the approach proposed in this paper to create Hilbert transformers is that in the implementation there are only real-valued multipliers, especially for Type 3 FIR filters derived from Type 4 filters, and thereby the reduction of arithmetic complexity is remarkable compared to the FRM technique. Second, linear programming is effectively used to optimize the unknown polynomial coefficients and the additional center coefficients. Design examples for both conventional filters and Hilbert transformers are included in order to show the benefits of the proposed design approach and the resulting filters. In this paper only Type 1 lowpass filters and Type 3 and 4 Hilbert transformers are discussed.

2 The Idea for a Piecewise-Polynomial Sinusoidal Approach

This section gives a reason for utilizing piecewise polynomials and combining them with a sinusoidal function to create impulse responses. The idea in our earlier approach [9] was to divide the impulse response into subresponses and to generate each subresponse with polynomials of a given degree. If we now consider an impulse response of an optimum wideband linear-phase FIR filter, it is seen in Fig. 1(a) that the impulse response has a narrow main lobe and the side lobes have very rapid changes in sign. This means that using our earlier piecewise-polynomial impulse response approach would require very many polynomial pieces to form the overall impulse response. Therefore, we are motivated to use sinusoids so that the polynomial pieces follow the polynomial-sinusoidal shapes to decrease the number of polynomial pieces and, thereby, to reduce the number of coefficients. To illustrate how such a polynomial piece can be generated, we consider an example of a filter with a polynomial degree two. Later, in Sect. 8, that example will be considered in more detail.

Consider the seventh polynomial subresponse of that example as shown in Fig. 2(a). Now, the subresponse is multiplied by a sinusoidal function, i.e.,

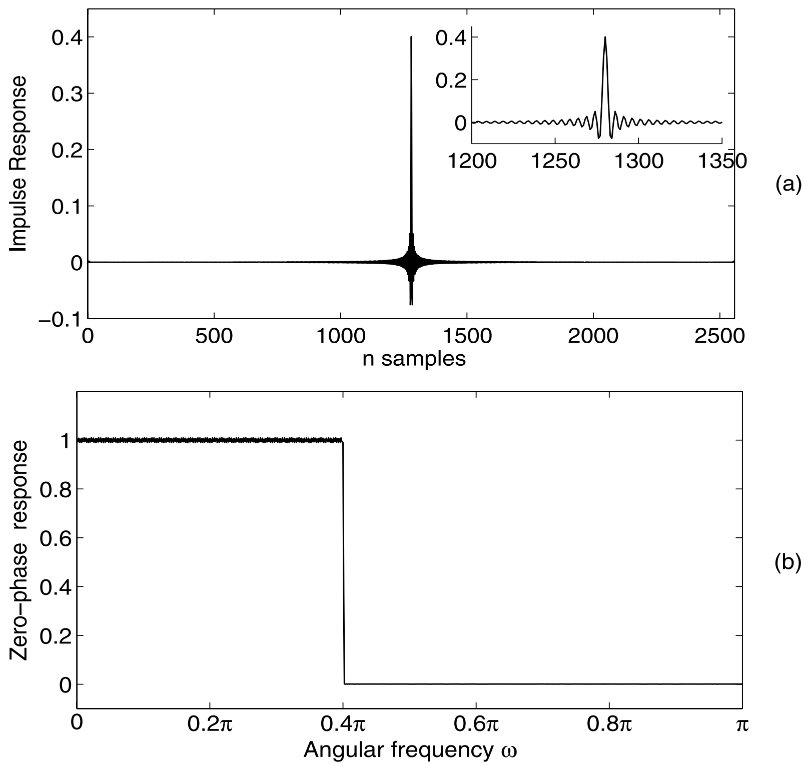


Fig. 1 Typical (**a**) impulse response and (**b**) zero-phase frequency response for a wideband linear-phase FIR filter. The filter has been optimized by using the Remez multiple-exchange algorithm and it has the minimum order, $N = 2558$, to meet the specifications: Passband edge $\omega_p = 0.4\pi$, stopband edge $\omega_s = 0.402\pi$, passband ripple $\delta_p = 0.01$, stopband ripple $\delta_s = 0.001$

$\sin(\omega_c(n - N))$, where N is half of the order, and ω_c is roughly the angular frequency at which the zero-phase frequency response achieves the value of half, and the seventh polynomial subresponse becomes a sinusoidal subresponse as shown in Fig. 2(b). However, to form a Type 1 response, piecewise-polynomial responses have to be created starting with a Type 3 response because of the antisymmetrical characteristics of a sinusoidal function. This causes the center sample to be zero valued as seen in Figs. 2 and 3. Therefore, one separately generated center coefficient is needed at the center of symmetry to form a Type 1 response. Furthermore, because of the very narrow and sharp main lobe it is beneficial to use several additional center coefficients around the center of symmetry to make it easier to form the overall impulse response and to reduce the overall complexity to meet the given criteria. All the nine subresponses are shown in Fig. 3. We will see in Sect. 8 an example of a filter with a polynomial degree two. The filter in our example has the same specifications as the optimum linear-phase FIR filter in Fig. 1. Our example consists of subresponses of different lengths; thereby, each block of the impulse response consists of a different number of polynomials up to the center of symmetry, which are shown in Fig. 3. In Fig. 3, only eight middle blocks of the impulse response are shown for clarity reasons: e.g. the sixth block consists of six polynomials, the seventh block of seven polynomi-

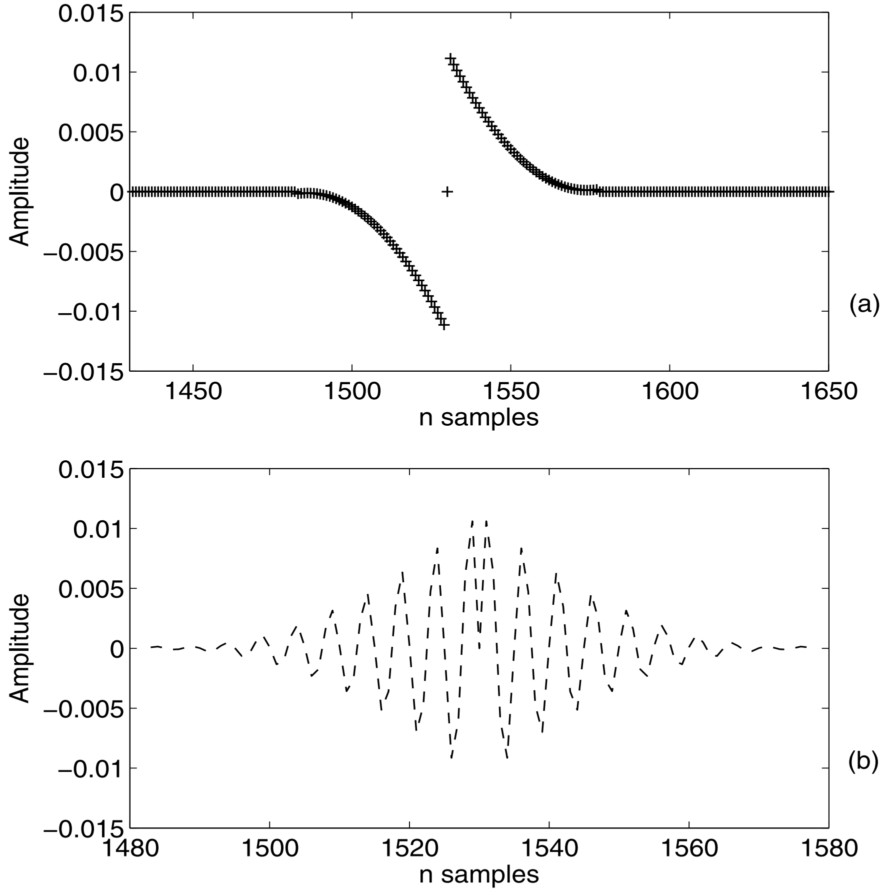


Fig. 2 (**a**) The seventh polynomial subimpulse response. (**b**) The seventh polynomial subresponse multiplied with a sinusoidal function to form a rapidly varying subimpulse response

als, and the (left) middle block of the impulse response consists of eight polynomials. In the overall impulse response there are altogether 16 such blocks. It is worth mentioning that because of the inherent symmetries of the blocks, eight of these blocks can be separately optimized and the remaining blocks are simultaneously created.

After summing up these subresponses in Fig. 3 and the separately generated center coefficients in Fig. 4, the overall impulse response, shown in Fig. 5, is obtained. It has the same general shape as the optimum linear-phase FIR filter in Fig. 1(a).

3 Startup Idea in the Chu–Burrus Approach

This section shows the original Chu–Burrus [4, 5] approach to synthesize wideband linear-phase filters for Types 1, 3 and 4 based on the use of piecewise-polynomial-sinusoidal impulse responses. The original approach is briefly reviewed for the causal case unlike in the Chu-Burrus approach.

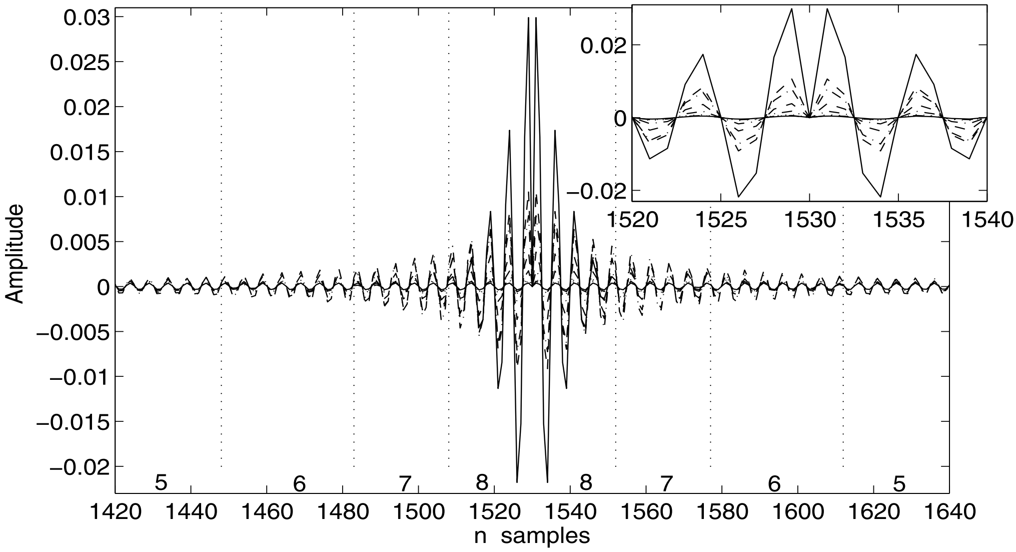


Fig. 3 This example includes altogether eight subimpulse responses. All the subimpulse responses have a zero-valued coefficient at the center of symmetry. The *vertical lines* mark the block edges and the numbers at the bottom mark block numbers. The sixth block consists of six polynomials, the seventh block of seven, and the eight (*left*) block (*middle*) of eight polynomials

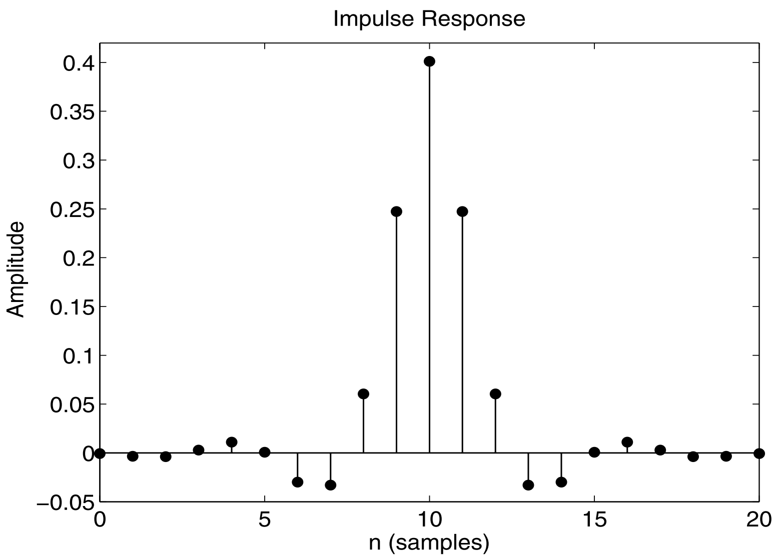


Fig. 4 Separate center coefficients, altogether 21, symmetrically located around the mid coefficient

This idea arises from the windowing technique for linear-phase FIR filters. The technique was used by Chu and Burrus [4, 5] to find a way to synthesize linear-phase FIR filters with piecewise-polynomial-sinusoidal impulse responses. Consider FIR filters of order $2N$ for a Type 1 lowpass filter and a Type 3 Hilbert transformer, as well as filters of order $2N-1$ for a Type 4 Hilbert transformer. The impulse response

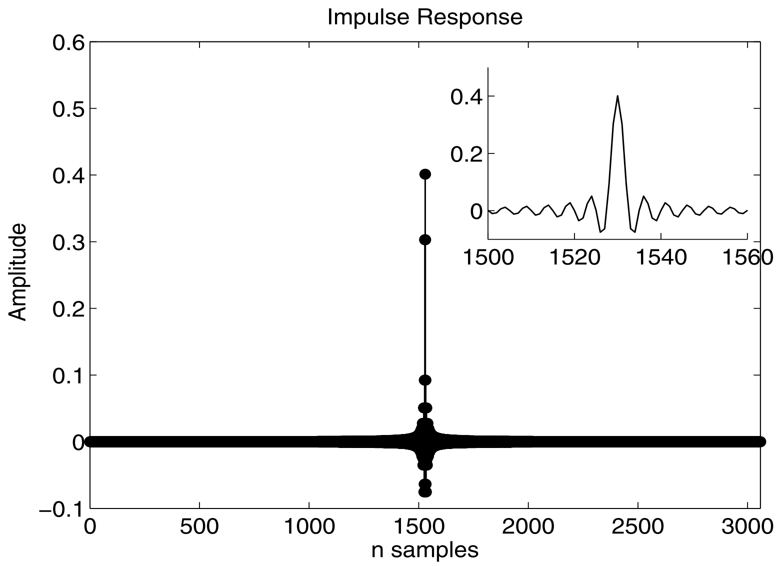


Fig. 5 Overall impulse response

by the basic windowing technique becomes

$$h(n) = w(n)h_0(n), \tag{1}$$

where $w(n)$ is a window function and $h_0(n)$, for Type 1, is given by

$$h_0(n) = \begin{cases} \frac{\sin(\omega_c(n-N))}{(n-N)\pi} & \text{for } n = 0, 1, \ldots, N-1, N+1, \ldots, 2N, \\ \omega_c/\pi & \text{for } n = N, \end{cases} \tag{2a}$$

where ω_c is the cutoff frequency, and for Type 3,

$$h_0(n) = \begin{cases} 0 & \text{for } n - N \text{ even}, \\ \frac{1-\cos((n-N)\pi)}{(n-N)\pi} & \text{for } n - N \text{ odd}, \end{cases} \tag{2b}$$

with nonzero values for $0 \le n \le 2N$. An odd-order impulse response of Type 4 can be expressed as

$$h_0(n) = \frac{1}{(n - (2N-1)/2)\pi}. \tag{2c}$$

$h_0(n)$ is an impulse response for the cases in (2a)–(2c).

The resulting transfer functions can be expressed in the following way:

$$F(z) = \begin{cases} \sum_{n=0}^{2N} (W(n)\sin[\omega_c(n-N)]z^{-n}) + \frac{\omega_c}{\pi} z^{-N}, & \text{Type 1}, \\ \sum_{n=0}^{2N} W(n)[1 - \cos((n-N)\pi)]z^{-n}, & \text{Type 3}, \\ \sum_{n=0}^{2N-1} W(n)z^{-n}, & \text{Type 4}, \end{cases} \tag{3}$$

with

$$W(n)=\begin{cases} w(n-N)/[\pi(n-N)], & \text{Type 1,}\\ w(n-N)/[\pi(n-N)], & \text{Type 3,}\\ w[n-(2N-1)/2]/[\pi(n-(2N-1)/2)], & \text{Type 4,}\end{cases} \tag{4}$$

where, for Types 3 and 1, $W(n)$ is an impulse response satisfying $W(2N-n)=-W(n)$ for $n=0,1,\dots,N-1$, $W(N)=0$, and for Type 4, it satisfies $W(2N-1-n)=-W(n)$ for $n=0,1,\dots,N$. It should be emphasized that for Type 1 filters the overall response becomes symmetrical after multiplying the antisymmetric impulse response with an antisymmetrical sinusoidal function according to (3). Next, we show how to implement $F(z)$ for Type 1 lowpass filters and for Type 3 and 4 Hilbert transformers.

3.1 Implementation of Type 1 Filters

Start by generating a Type 3 envelope transfer function

$$E(z)=\sum_{n=0}^{2N} W(n)z^{-n} \tag{5}$$

using our earlier approach for a piecewise-polynomial impulse response [9]. Create a transfer function $F(z)-(\omega_c/\pi)z^{-N}$ as follows:

Step 1 Replace z^{-1} by $z^{-1}e^{j\omega_c}$ in $E(z)$ given in (5) giving the transfer function

$$\widetilde{F}(z)=\sum_{n=0}^{2N} W(n)\left(z^{-1}e^{j\omega_c}\right)^n. \tag{6}$$

Step 2 Multiply $\widetilde{F}(z)$ by $e^{-jN\omega_c}$ resulting in

$$\widehat{F}(z)=e^{-jN\omega_c}\sum_{n=0}^{2N} W(n)\left(z^{-1}e^{j\omega_c}\right)^n. \tag{7}$$

Step 3 To obtain the output, take the imaginary part of the complex-valued output of $\widehat{F}(z)$ as seen later in Fig. 7.

Step 4 Add an additional branch $F_1(z)=z^{\widehat{N}}\widehat{H}(z)$, where $\widehat{H}(z)$ contains at least one term of the form: (ω_c/π). See Fig. 4 for an example. The overall filter structure is also seen later in Fig. 7.

3.2 Implementation of Type 4 Filters

Start now by generating a Type 4 envelope transfer function

$$E(z)=\sum_{n=0}^{2N-1} W(n)z^{-n} \tag{8}$$

using our earlier approach for a piecewise-polynomial impulse response [9] modified by adding separately generated center coefficients.

3.3 Implementation of Type 3 Filters

Start by generating a Type 3 envelope as $E(z)=\sum_{n=0}^{2N} f(n)z^{-n}$, where

$$f(n)=\begin{cases} W(n) & \text{for } n-N \text{ odd,} \\ 0 & \text{for } n-N \text{ even,} \end{cases}$$

thereby N should be odd in order to prevent both $W(0)$ and $W(2N)$ from becoming zero valued, and $F(z)$ becomes

$$F(z)=\sum_{n=0}^{2N} W(2n)z^{-2n}. \tag{9}$$

This means that if we first assume that the transfer function $E(z)$ of Type 4 in (8) is known, then the transfer function $F(z)$ of Type 3 in (9) can be uniquely determined from the Type 4 transfer function $E(z)$ by replacing z^{-1} with z^{-2}. Thereby, the order of the resulting Type 3 transfer function $F(z)$ becomes $2(2N-1)$ instead of $2N$ and $W(2n)\equiv W(n)$.

If the above is expressed conversely, i.e., if the transfer function $F(z)$ of Type 3 given in (9) is known, then the transfer function $E(z)$ of Type 4 in (8) is obtained uniquely from $F(z)$ by replacing z^{-2} with z^{-1}. Thereby, the order of the resulting transfer function $E(z)$ becomes $N/2$ and the envelope becomes $W(2n)$.

From this it follows that only Type 4 is needed. Type 4 can also be considered a special case of Type 3 filter according to (3), from which it can be seen that the cosine term, $\cos((n-N)\pi)$, is absent and only $W(n)\times 1$ is present in Type 4 compared to Type 3.

4 Proposed Linear-Phase Filter Classes for Wideband FIR Filters

First, the overall transfer function, denoted by $H(z)$, is constructed as described in our earlier method [9], i.e., M parallel branches are connected and delayed with z^{-N_m} in order to keep the center of symmetry at the same location for all the subimpulse responses. Second, the subresponses are modulated with a sinusoidal function as described in the previous subsection. Third, an arbitrary number of separately generated center coefficients is added as follows:

$$H(z)=\sum_{m=1}^{M} z^{-N_m}H_m(z)+z^{-\widehat{N}}\widehat{H}(z), \tag{10}$$

and the transfer functions $H_m(z)$, for Types 1 and 4, are given as

$$H_m(z)=\begin{cases} h_m(N-N_m)z^{-(N-N_m)}+\sum_{n=0}^{(N-N_m)-1} h_m(n)[z^{-n}+z^{-(2(N-N_m)-n)}], \\ \sum_{n=0}^{(N-N_m)-1} h_m(n)[z^{-n}-z^{-(2(N-N_m)-1-n)}], \end{cases} \tag{11}$$

respectively, where the integers N_m in the delay terms z^{-N_m} satisfy

$$N_1 = 0 \quad \text{and} \quad N_{m+1} > N_m \quad \text{for } m = 1, 2, \dots, M-1, \tag{12}$$

and the order of $H_m(z)$ is $2(N - N_m)$ for Type 1 and $[2(N - N_m) - 1]$ for Type 4. The impulse response of $H_m(z)$ is given by

$$h_m(n) = \begin{cases} \sum_{r=0}^{L} a_m^{(L)}(r) n^r \times \sin[\omega_c(n - (N - N_m))], & \text{Type 1}, \\ \sum_{r=0}^{L} a_m^{(L)}(r) n^r, & \text{Type 4}, \end{cases} \tag{13}$$

where $\sum_{r=0}^{L} a_m^{(L)}(r) n^r$ is an Lth degree polynomial. In addition, $z^{-\widehat{N}} \widehat{H}(z)$ is a conventional direct-form transfer function with nonzero impulse response coefficients $\widehat{h}(n)$ with

$$n = N - c + 1, N - c + 2, \dots, N - c + T \quad \text{Types 1 and 4}, \tag{14}$$

where $c = \lceil (T/2) \rceil$ and T is the number of additional coefficients at the center of the filter. $\lceil\ \rceil$ means rounding upwards. The order of $z^{-\widehat{N}} \widehat{H}(z)$ is $\widehat{N} + T - 1$.

The delay terms in (10) are used to shift the center of symmetry to the desired location. The center of each subimpulse response occurs at $n = N$ for Type 1 and $n = (2N - 1)/2$ for Type 4. Type 4 coincides with a $(2N - 1)$th-order piecewise-polynomial impulse response approach [9] as seen in (3) and (13) except for the separately generated center coefficients.

In order to indicate that the overall filter has a piecewise-polynomial-sinusoidal impulse response, the interval $n \in [0, N]$ is divided into the following M subintervals:

$$X_m = [N_m, N_{m+1} - 1] \quad \text{for } m = 1, 2, \dots, M - 1 \tag{15}$$

and

$$X_M = [N_M, N]. \tag{16}$$

The following facts should be pointed out. First, $X_1 = [0, N_2 - 1]$ because $N_1 = 0$. Second, the overall impulse response only needs to be examined up to $n = N$ and $n = N - 1$ due to the even or odd symmetry around this point.

The impulse response on X_m, for Type 1 and 4, can be expressed as

$$h(n) = \sum_{m=1}^{M} \widetilde{h}_m(n), \tag{17}$$

where $\widetilde{h}_m(n) = \sum_{k=1}^{m} h_k(n - N_m)$, for $m = 1, 2, \dots, M - 1$ and $\widetilde{h}_M(n) = \sum_{k=1}^{M} h_k(n - N_M) + \widehat{h}(n)$, where $h_k(n)$ is given in (13), and $\widehat{h}(n)$ is the impulse response of a conventional direct-form Type 1 or 4 transfer function $\widehat{H}(z)$ with nonzero coefficients for $n = 0, 1, \dots, T - 1$.

Based on the above equations, in each X_m for $m = 1, 2, \dots, M$, a separate polynomial-sinusoidal impulse response can be generated. In addition, in the X_M, there are additional center coefficients, which are of great importance for fine-tuning the overall filter to meet the given criteria.

Given the filter criteria as well as the design parameters ω_c, M, N, L, N_m's, and the number of center coefficients included in $\widehat{H}(z)$, the overall problem is solvable by using linear programming, instead of ad hoc nonlinear programming as used in the original Chu–Burrus approach.

5 Filter Optimization Algorithm

FIR filter optimization using linear programming can be stated as a problem of minimizing

$$\epsilon = \max_{\omega \in [0,\omega_p] \cup [\omega_s,\pi]} \left| W(\omega) \left[H(\omega) - D(\omega) \right] \right|, \tag{18}$$

where $W(\omega)$ is a weight function and $D(\omega)$ is the desired zero-phase frequency response. In the above optimization problem, ω_p, and ω_s are the passband and stopband edge angles, respectively. The most crucial feature for the filters under consideration is that the zero-phase frequency response, $H(\omega)$ [18], is linear with respect to the unknown polynomial coefficients. This fact makes it possible to solve the *Optimization Problem* (stated later) by using linear programming. First, a linear programming problem can be expressed as follows:

$$\begin{aligned} &\min \mathbf{f}^{\mathrm{T}} \mathbf{x} \\ &\text{subject to } \mathbf{A}\mathbf{x} \le \mathbf{b}^{\mathrm{T}}. \end{aligned} \tag{19}$$

Now our optimization problem is turned into this linear programming problem:

$$\mathbf{A} = \begin{pmatrix} H(\omega) & -1/W(\omega) \\ -H(\omega) & -1/W(\omega) \end{pmatrix}, \qquad \mathbf{b} = \left[D(\omega) \quad -D(\omega) \right], \tag{20}$$

and $\mathbf{x}$ is a vector with $(M \times (L+1)) + 1$ unknowns to be solved, including ϵ in (18). $\mathbf{f}^{\mathrm{T}}$ is a vector of parameters of the cost function. T in $\mathbf{f}^{\mathrm{T}}$ means the transpose. This vector describes the variable according to which the optimization is performed. In this case it is ϵ and it corresponds to the overall ripple value, and the vector is defined as

$$\mathbf{f}^{\mathrm{T}} = [0 \quad 0 \quad \dots \quad 0 \quad 1]. \tag{21}$$

The above formulation in (20)–(21) is valid for *SeDuMi* and *MATLAB*™'s *linprog*-routine [14] without including the transition band in the optimization. A more detailed description of a linear programming problem solved by *SeDuMi* is found in the Appendix.

In the practical optimization problem the zero-phase frequency response is in the following form:

$$H(\omega) = \sum_{m=1}^{M} \sum_{r=0}^{L} H(\omega, m, r) + \sum_{m=0}^{c-1} H_1(\omega, m), \tag{22}$$

where

$$H(\omega,m,r)=\begin{cases} a_m^{(L)}(r)(\widetilde{N}_m)^r\sin(\omega_{\mathrm{c}}\widetilde{N}_m) \\ \quad+\sum_{n=1}^{\widetilde{N}_m}a_m^{(L)}(r)(\widetilde{N}_m-n)^r\sin(\omega_{\mathrm{c}}(\widetilde{N}_m-n))2\cos(n\omega), \\ \sum_{n=1}^{\widetilde{N}_m-1}a_m^{(L)}(r)(\widetilde{N}_m-1-n)^r2\sin((n-0.5)\omega), \end{cases} \tag{23}$$

and for Type 1

$$H_1(\omega,m)=\begin{cases} a_{m+1}^{(L)}(N)\mathbf{1} & \text{for } m=0, \\ a_{m+1}^{(L)}(N-(m+1))2\cos((m+1)\omega), & \text{otherwise}, \end{cases} \tag{24}$$

where $\mathbf{1}$ is an all-ones vector with the same length as ω. For Type 4

$$H_1(\omega,m)=a_{m+1}^{(L)}\big((2N-1)/2-(m+1)\big)2\sin\big((m+1)-0.5\big)\omega\big). \tag{25}$$

Additionally, $\widetilde{N}_m=N-N_m$, and $a_m^{(L)}(r)$s in (10) are the unknown polynomial coefficients and $a_{m+1}^{(L)}$s in (11) are the additional center coefficients to be optimized. The main concern, for the optimization problem, is the parameters, i.e., how to choose them so that after applying linear programming, the polynomial coefficients, which make the weighted error function equal to or less than unity, can be found as defined by (23)–(30). The following parameters should be chosen: the number of slices M, the filter order and the common polynomial degree L in each slice and the values N_m, for the mth slice for $m=1,2,\ldots,M$, so that the arithmetic complexity is minimized. No straightforward way to find a unique optimum combination of parameters has been found. Some advice on how to choose the overall filter order can be found in the fact that the filter order is slightly greater than the minimum order of a direct-form FIR filter with the same design criteria. For the cases in Sect. 8 it has been found that the length of the shortest slice should be approximately 7–35 samples for the best solution of the lowpass case, when the symmetry condition is exploited. The number of center taps should be approximately half of the length of the shortest slice for the linear-phase types proposed in this paper. These guidelines are, of course, dependent on the design criteria.

Optimization Problem

1. We are given ω_{p}, ω_{s}, δ_{p} and δ_{s}, and $\omega_{\mathrm{c}}\approx(\omega_{\mathrm{p}}+\omega_{\mathrm{s}})/2$ for Type 1.
2. Choose the overall filter order and the number of slices M, the common polynomial degree L in each slice and the values N_m, for the mth slice for $m=1,2,\ldots,M$.
3. Form the polynomials for each slice.
4. Form the following weighted error function:

$$E(\omega)=\big|W(\omega)\big|\big|H(\omega)-D(\omega)\big|, \tag{26}$$

with respect to the unknown coefficients $a_L^{(m)}$s. Here,

$$D(\omega)=\begin{cases} 1 & \text{for } \omega\in[0,\ \omega_{\mathrm{p}}], \\ 0 & \text{for } \omega\in[\omega_{\mathrm{s}},\ \pi], \end{cases} \tag{27}$$

for Type 1, and for a Hilbert transformer

$$D(\omega) = 1 \quad \text{for } \omega \in [\omega_p,\ \pi],\ \text{Type 4,} \tag{28}$$

and for Type 1

$$W(\omega) = \begin{cases} 1/\delta_p & \text{for } \omega \in [0,\ \omega_p], \\ 1/\delta_s & \text{for } \omega \in [\omega_s,\ \pi]. \end{cases} \tag{29}$$

For Type 4, i.e., a Hilbert transformer,

$$W(\omega) = 1 \quad \text{for } \omega \in [\omega_p,\ \pi]. \tag{30}$$

5. Solve the unknown polynomial coefficients $a_L^{(m)}$, given in (23), with a linear programming algorithm.
6. If the polynomial coefficients $a_L^{(m)}$ make the quantity in (18) equal to or less than unity, the design criteria are automatically met and the filter is successfully designed; otherwise continue from step 2.

One issue is worth noting, namely the fact that in case of numerical problems, the polynomial values should be scaled to be between zero and one in the optimization. It has been noticed that if the polynomial values are very high, this can cause numerical problems in optimization.

6 Implementation Structures

This section shows how to generate an arbitrary piecewise-polynomial-sinusoidal impulse response for linear-phase FIR filter Types 1 and 4. The approach is based on our earlier structures for a piecewise-polynomial impulse response [9]. Therefore, only a summary of the structures is given. There are two modifications to be made. First, additional center coefficients are added as a parallel connection of a direct-form FIR filter. Second, complex multipliers are added after each delay term as explained in Sect. 3.1. However, a Type 4 filter does not require any complex multipliers, therefore only one direct-form filter is added in this case.

The main idea in [9] was to reduce the order and the length of the polynomials by one with the aid of an accumulator at every step until the polynomial order of zero is reached. All the types have their own implementation structures. In order to achieve the desired goal, first an implementation block is shown for a single branch. Based on this structure, an overall implementation structure is generated.

Based on the design technique proposed in Sect. 4, the linear-phase FIR filter transfer functions for Types 1 and 4 are of the form

$$H(z) \equiv H^{(0)}(z) = \begin{cases} h^{(0)}(N)z^{-N} + \sum_{n=0}^{N-1} h^{(0)}(n)[z^{-n} + z^{-(2N-n)}], \\ \sum_{n=0}^{N-1} h^{(0)}(n)[z^{-n} - z^{-(2N-1-n)}], \end{cases} \tag{31}$$

where

$$h^{(0)}(n) = p^{(L)}(n) = \begin{cases} \sin(\omega_c(n-N)) \sum_{r=0}^{L} a^{(L)}(r)n^r, \\ \sum_{r=0}^{L} a^{(L)}(r)n^r, \end{cases} \tag{32}$$

respectively, for $0 \le n \le \widehat{N}_0$, with

$$\widehat{N}_0 = \begin{cases} N & \text{for Type 1,} \\ N-1 & \text{for Type 4.} \end{cases} \tag{33}$$

There exist $L+1$ unknowns in the above transfer functions. For Types 1 and 4, the impulse responses satisfy $h^{(0)}(2N-n) = h^{(0)}(n)$ for $n = 0, 1, \dots, 2N$; and $h^{(0)}(2N-1-n) = -h^{(0)}(n)$ for $n = 0, 1, \dots, 2N-1$; respectively. For Type 1 (Type 4), the impulse response is symmetrical (antisymmetrical). Furthermore, for Type 1 (Type 4), the filter order is $2N$ $(2N-1)$. In order to implement the overall transfer function $H(z)$ efficiently by using accumulators for Types 1 and 4, starting with (32), the following functions are determined recursively for $k = 1, 2, \dots, L$ for Types 1 and 4:

$$\begin{aligned} p^{(L-k)}(n) &= p^{(L-k+1)}(n+1) - p^{(L-k+1)}(n) \\ &= \begin{cases} \sin(\omega_c(n-N)) \sum_{r=0}^{L-k} a^{(L-k)}(r) n^r, \\ \sum_{r=0}^{L-k} a^{(L-k)}(r) n^r, \end{cases} \end{aligned} \tag{34}$$

respectively, where

$$a^{(L-k)}(r) = \sum_{s=0}^{L-k-r} \binom{L-k+1-s}{r} a^{(L-k+1)}(L-k+1-s), \tag{35}$$

i.e., two $(L-k+1)$th-degree polynomials, $p^{(L-k+1)}(n+1)$ and $p^{(L-k+1)}(n)$, are subtracted from each other to form a polynomial of a lower degree, $(L-k)$. This is continued until the polynomial of order zero is reached. The sinusoidal terms in (34) are taken as complex multipliers in the structure according to Sect. 3.1 for Type 1 lowpass filters. This means also that sinusoidal terms are not included in the calculation of the implementation coefficients. At the end of the section, a simple example of how to calculate the implementation coefficients for a Type 4 FIR filter will be given.

The overall structure for a piecewise-polynomial-sinusoidal Type 1 filter is obtained from the structure of the piecewise-polynomial impulse response Type 3 filters, proposed in [9] and shown in Fig. 6, as described in Sect. 3.1. The overall structure for a piecewise-polynomial-sinusoidal impulse response of a Type 1 filter is shown in Fig. 7. The overall structure in Fig. 7 is the combination of M basic implementation blocks, one for each subresponse, shown in Fig. 8. The implementation structure for Type 1 consists of $L+1$ levels.

The structure coefficients, the α_k's and β_k's, are given for Types 1 and 4. Here, the α_k's, for $k = 1, 2, \dots, L$, are related to the corresponding functions, $p^{(L-1+k)}(n)$s, as given by (32), (34) and (35) (without sinusoidal terms) through

$$\alpha_k = p^{(L+1-k)}(0), \quad k = 1, 2, \dots, L, \text{ for both types.} \tag{36}$$

Table 1 shows values of the α's for different polynomial degrees, L. Non-zero values of the β_k's for Types 1 and 4 are given by

$$\beta_{2k-1} = -p^{(L-2(k-1))}(N-k), \quad k = 1, 2, \dots, \lfloor (L+1)/2 \rfloor, \tag{37a}$$

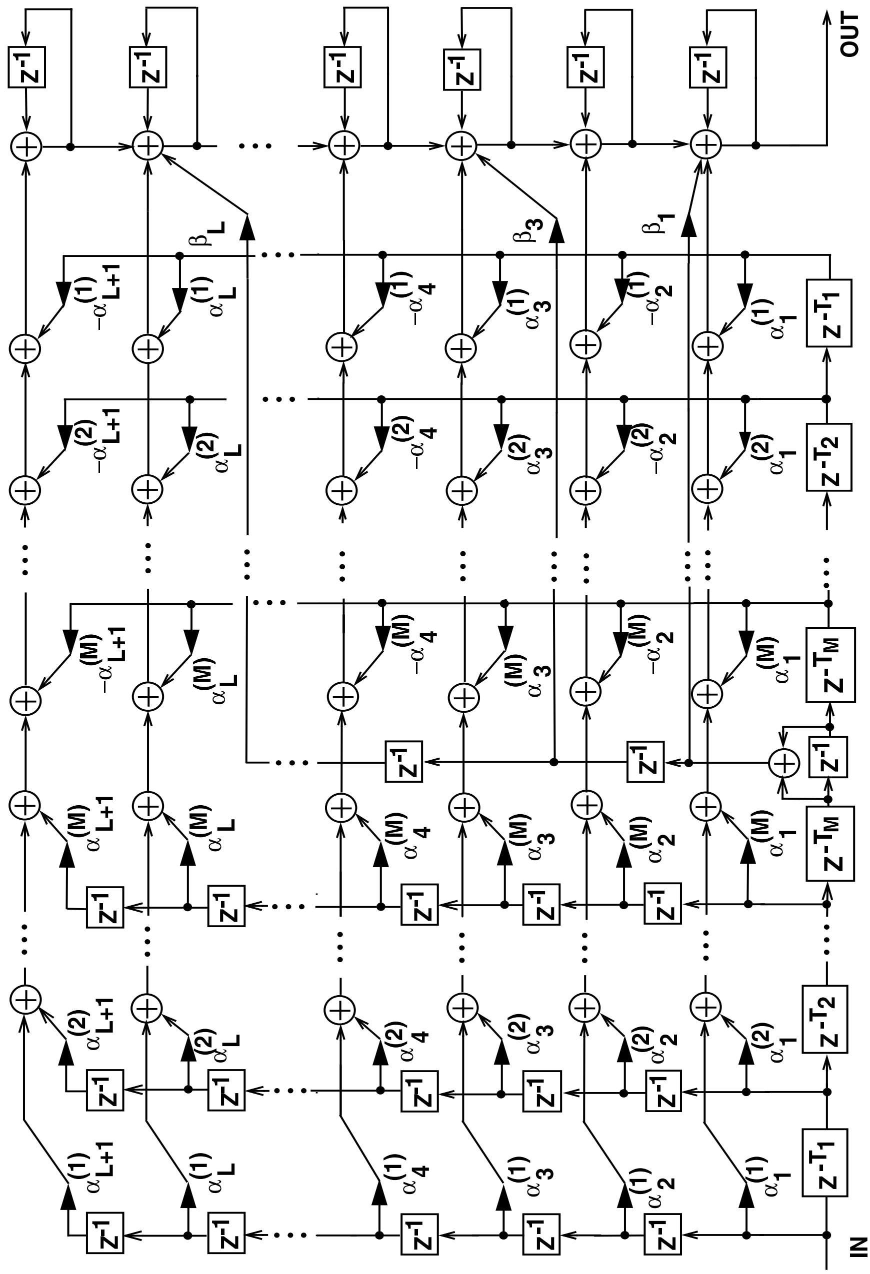


Fig. 6 Implementation structure of Type 3 piecewise-polynomial filters

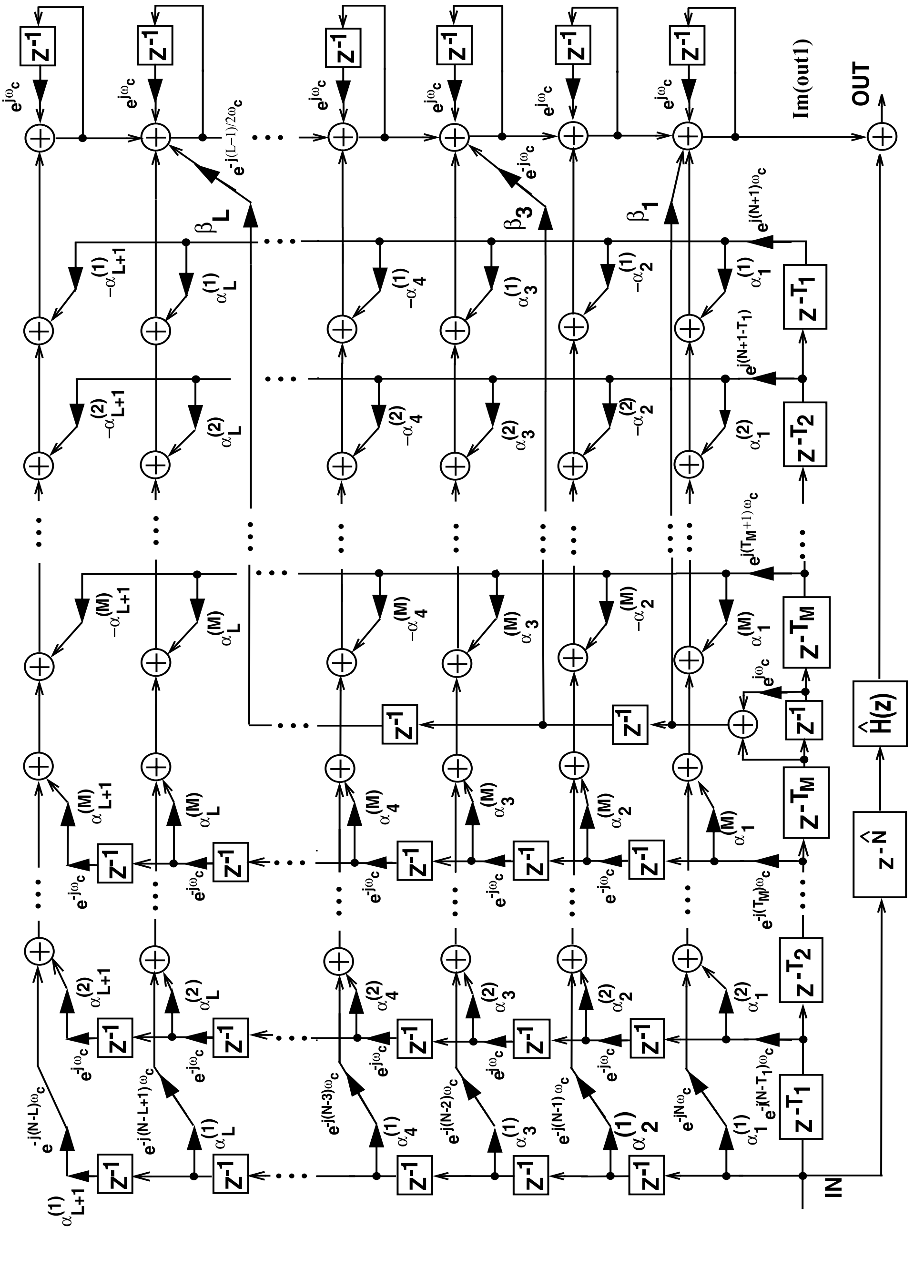


Fig. 7 Implementation structure of Type 1 piecewise-polynomial-sinusoidal filters

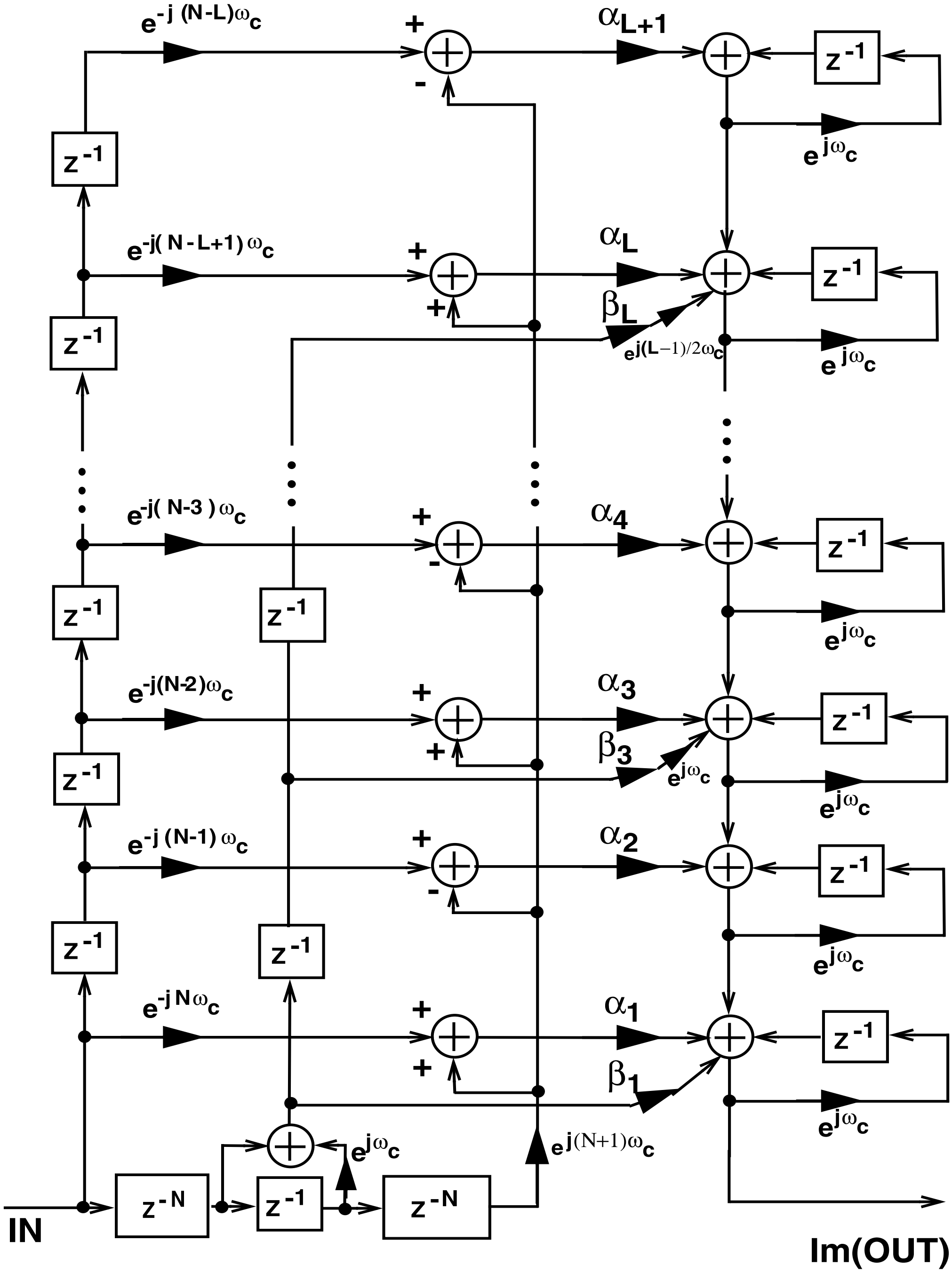


Fig. 8 Basic implementation block of Type 1 filters

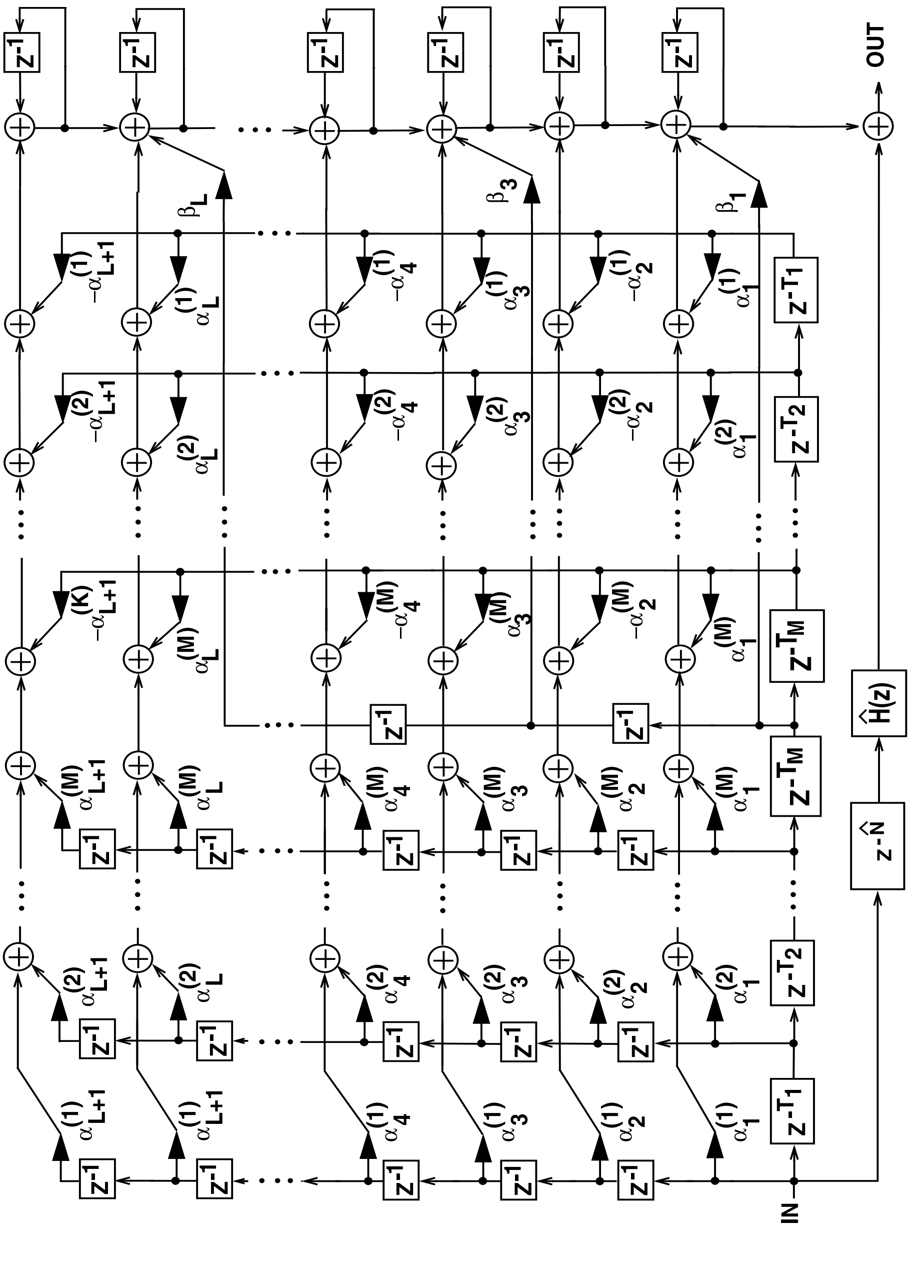


Fig. 9 Implementation structure of Type 4 filters

Table 1 Values of α_k for different polynomial degrees L

L	k	α_k	L	k	α_k
2	1	$\alpha_1 = a^{(2)}(0)$	4	1	$\alpha_1 = a^{(4)}(0)$
	2	$\alpha_2 = a^{(2)}(1) + a^{(2)}(2)$		2	$\alpha_2 = a^{(4)}(4) + a^{(4)}(3) + a^{(4)}(2) + a^{(4)}(1)$
	3	$\alpha_3 = 2a^2(2)$		3	$\alpha_3 = 14a^{(4)}(4) + 6a^{(4)}(3) + 2a^{(4)}(2)$
3	1	$\alpha_1 = a^{(3)}(0)$		4	$\alpha_4 = 36a^{(4)}(4) + 6a^{(4)}(3)$
	2	$\alpha_2 = a^{(3)}(3) + a^{(3)}(2) + a^{(3)}(1)$		5	$\alpha_5 = 24a^{(4)}(4)$
	3	$\alpha_3 = 6a^{(3)}(3) + 2a^{(3)}(2)$			
	4	$\alpha_4 = 6a^{(3)}(3)$			

and

$$\beta_{2k-1} = -p^{(L-2(k-1))}(N-k) - p^{(L-2(k-1))}\big((N-k)-1\big), \tag{37b}$$

where $k = 1, 2, \ldots, \lfloor (L+1)/2 \rfloor$, respectively.

Here, for odd L, for Types 1 and 4, β_{L+1} is given as

$$\beta_{L+1} = -p^{(0)}(0) \quad \text{and} \quad \beta_{L+1} = -2p^{(0)}(0),$$

respectively.

The overall structure of M blocks with complex multipliers for Type 1 shown in Fig. 7 has delay terms as follows: $T_m = N_m - N_{m-1}$ for $m = 2, 3, \ldots, M-1$, $T_M = N - N_M$. The role of the T_m's is to keep the center of the symmetry of the subresponses at the center of the overall filter.

To generate the corresponding coefficients α_k and β_{2k} in the overall implementation, the impulse response given by (13) is used. The resulting α_k and β_{2k} are denoted by $\alpha_k^{(m)}$ and $\beta_{2k}^{(m)}$. Furthermore, the β_{2k}'s are the sums of the corresponding $\beta_{2k}^{(m)}$'s.

In order to keep the overall diagram simple, the coefficients $\alpha_k^{(m)}$ have been drawn twice. In the practical implementation, the overall number of multipliers can be reduced by first adding [subtracting] the inputs of the left-hand side $\alpha_k^{(m)}$ and the right-hand side $\alpha_k^{(m)}$ [$-\alpha_k^{(m)}$] and by then multiplying the result by $\alpha_k^{(m)}$. Similarly, the overall implementation structure of M subresponses for a Type 4 filter is generated and is shown in Fig. 9, where $T_m = N_m - N_{m-1}$ for $m = 2, 3, \ldots, M-1$ and $T_M = N - N_M$. As a simple example to illustrate the derivation of the implementation coefficients, consider the Hilbert transformer of Type 4 with the following specifications: $\Omega_\mathrm{p} = [0.05\pi, \pi]$ and the passband ripple is $\delta_\mathrm{p} = 0.01$. The criteria are met by the filter of order 240, $L = 2$, $M = 4$, $T = 16$, $N_1 = 0$, $N_2 = 60$, $N_3 = 90$, $N_4 = 105$ and the ripple value obtained is 0.00969. The optimized coefficient values are shown in Table 2 and the implementation coefficients, $\alpha_k^{(m)}$ and β_{2k-1}, in Table 3.

7 Hilbert Transformers

This section shows an application example for proposed Type 3 and 4 filters, namely Hilbert transformers. This example shows the properties and efficiency of filter

Table 2 Filter coefficients for the simple example of a Hilbert transformer

Slice m	Optimized coeff. $a_m^{(L)}(r)$s	Coeff. values	Separate center coeff., number k up to the center of symmetry	Values of the separate center coeff.
1	$a_1^{(2)}(0)$	0.000754961341879	1	0.623486118162552
1	$a_1^{(2)}(1)$	−0.00000936333201	2	0.15159716143810
1	$a_1^{(2)}(2)$	0.000000928224555	3	0.070292990072642
2	$a_2^{(2)}(0)$	−0.000346931193292	4	0.038394617046029
2	$a_2^{(2)}(1)$	0.000044411113383	5	0.021983178252685
2	$a_2^{(2)}(2)$	−0.000000230226545	6	0.012559931133717
3	$a_3^{(2)}(0)$	0.001784142790681	7	0.00659804763397
3	$a_3^{(2)}(1)$	−0.000262106510178	8	0.005576993714508
3	$a_3^{(2)}(2)$	0.000042287618263		
4	$a_4^{(2)}(0)$	−0.00108272206937		
4	$a_4^{(2)}(1)$	−0.00010526842470		
4	$a_4^{(2)}(2)$	0.00005780591838		

Table 3 Implementation coefficients of block m for the simple example of a Hilbert transformer

$\alpha_k^{(m)}$	$\beta_{2k-1}^{(m)}$
$\alpha_1^{(m)} = a_m^{(2)}(0)$	$\beta_1^{(m)} = -2\alpha_1^{(m)} - (2(N-N_m)-1)\alpha_2^{(m)} - (N-N_m)^2\alpha_3^{(m)} - 2a_m^{(2)}(0)$
$\alpha_2^{(m)} = a_m^{(2)}(1) + a_m^{(2)}(2)$	$+ (-2a_m^{(2)}(1) + 2a_m^{(2)}(2))(N-N_m) + a_m^{(2)}(1) - a_m^{(2)}(2)$
$\alpha_3^{(m)} = 2a_m^{(2)}(2)$	$- 2a_m^{(2)}(2)(N-N_m)^2$
	$\beta_3^{(m)} = -2\alpha_3^{(m)}$

classes over the FRM-based Hilbert transformers proposed in [13]. For comparison purposes, a Hilbert transformer is designed for the same application as in [13], i.e., an acoustic feedback system.

In these systems problems are encountered due to the oscillations caused by the feedback. If elements in the system are not overdriven, it is linear, which makes linear filters very suitable to remove these harmful oscillations. In order to attenuate these oscillations, a frequency shifter is added in the system. The frequency shifter shifts the frequency by some amount in the range of 0 to 5 Hz. For most people a shift of 2 Hz is sufficient because our ears are not able to notice it. Thereby, a shifter requires a very sharp Hilbert transformer to shift low frequencies properly. Such a Hilbert transformer has the following specifications: sampling rate of 32 kHz, lower passband edge of 20 Hz and peak ripple magnitude of 0.0001 (as given in [13]).

The required Hilbert transformer has a transition band of the same width at the low and the high frequency areas. In general, Hilbert transformers are very well suited to be designed with the proposed Type 3 and 4 filters. The passband for Hilbert trans-

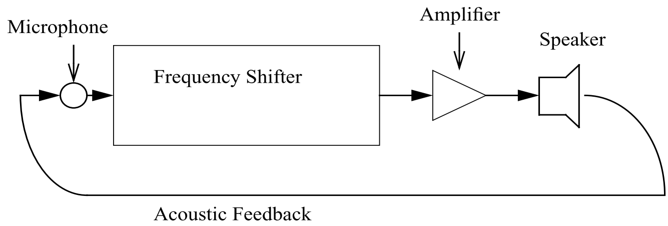


Fig. 10 An acoustic feedback system

formers is very wide with a somewhat narrow transition band. Some applications, like an acoustic feedback system [13], require an extremely narrow transition band for Hilbert transformers. These systems are used e.g. to amplify voice. In such systems there appears acoustic noise between the speaker and the system. Often this acoustic noise is modeled with one acoustic feedback, as shown in Fig. 10.

First a Hilbert transformer which only has one transition band at low frequencies is designed, and then by substituting every delay element, z^{-1}, with z^{-2}, the desired Hilbert transformer is obtained. This means that the transition band is shrinking by a factor of 2. There is, however, one difference between our design and the design of Lim et al. [13]: they use normalized frequencies where 0.5 corresponds to π in the case of the angular frequency ω. Thereby, our specification for a Type 4 filter is as follows: the passband $\Omega_p = [0.0025\pi, \pi]$ and the passband ripple value less than or equal to 10^{-4}.

For the proposed approach, the criteria are met by a filter of order 2041, $M = 6$, $L = 4$, $T = 30$, $N_1 = 0$, $N_2 = 430$, $N_3 = 725$, $N_4 = 880$, $N_5 = 955$, and $N_6 = 990$. The overall ripple value obtained is $9.8681e{-}05$. This filter thus consists of six slices with the polynomial degree $L = 4$ in each slice. In this case, the number of unknowns required both in the optimization and in the implementation is only 45. The number of multipliers is 48 in a practical implementation. The magnitude and the impulse response for the optimized filter are shown in Fig. 11. After replacing z^{-1} by z^{-2}, the corresponding FRM-based Hilbert transformer designed in [13] requires 213 multipliers when exploiting the coefficient symmetry, and the effective filter order is 4106 compared to our design, which only requires 48 multipliers in a practical implementation, and a slightly lower filter order, 4082. The final design of the Type 3 filter is shown in Fig. 12.

8 A Lowpass Design

This section shows the benefits and the efficiency of the proposed filter classes for Type 1 FIR filters, over the FRM-based filters, by means of an example.

The passband and the stopband edges, for the lowpass design, are $\omega_p = 0.4\pi$, $\omega_s = 0.402\pi$ with $\delta_p = 0.01$, and $\delta_s = 0.001$ (60 dB attenuation). The given criteria are met by a filter of order 3058, $M = 8$, $L = 2$, the number of separate center coefficients $T = 21$, $N_1 = 0$, $N_2 = 343$, $N_3 = 806$, $N_4 = 1170$, $N_5 = 1352$, $N_6 = 1447$, $N_7 = 1482$, $N_8 = 1507$. This filter thus consists of eight slices with the polynomial degree $L = 2$ in each slice. In the optimization of the filter, different values of ω_c were used, i.e., $\omega_c = 0.4010$, $\omega_c = 0.4008$ and $\omega_c = 0.4009$. In this case the best design

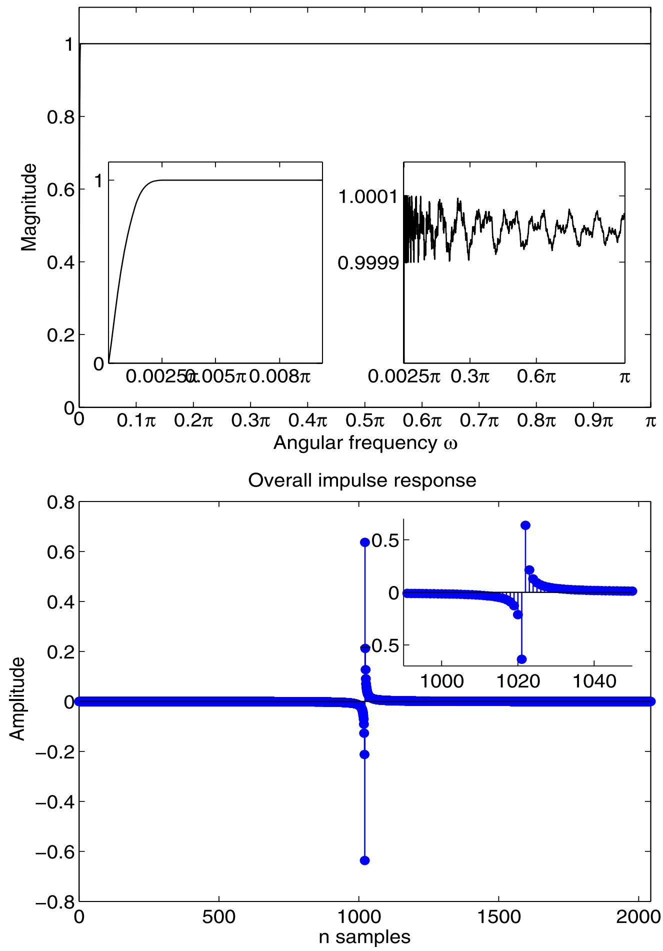


Fig. 11 A design example of proposed linear-phase FIR filters: Optimized Hilbert transformer of Type 4

Table 4 Filter parameters of the lowpass design

Polynomial degree L	Number of slices M	Number of sep. center coeff. T	Number of multipl.	Filter order	$N_m s$
2	8	21	$37 + 41$	3058	0, 343, 806, 1170, 1352, 1447, 1482, 1507

Table 5 Filter parameters of reference designs

Filter	Filter Order	Number of Multipliers
Direct-form FIR	2541	1271
FRM (three-stage design)	3196	94
FRM (two-stage design)	2920	107
FRM (one-stage design)	2690	168

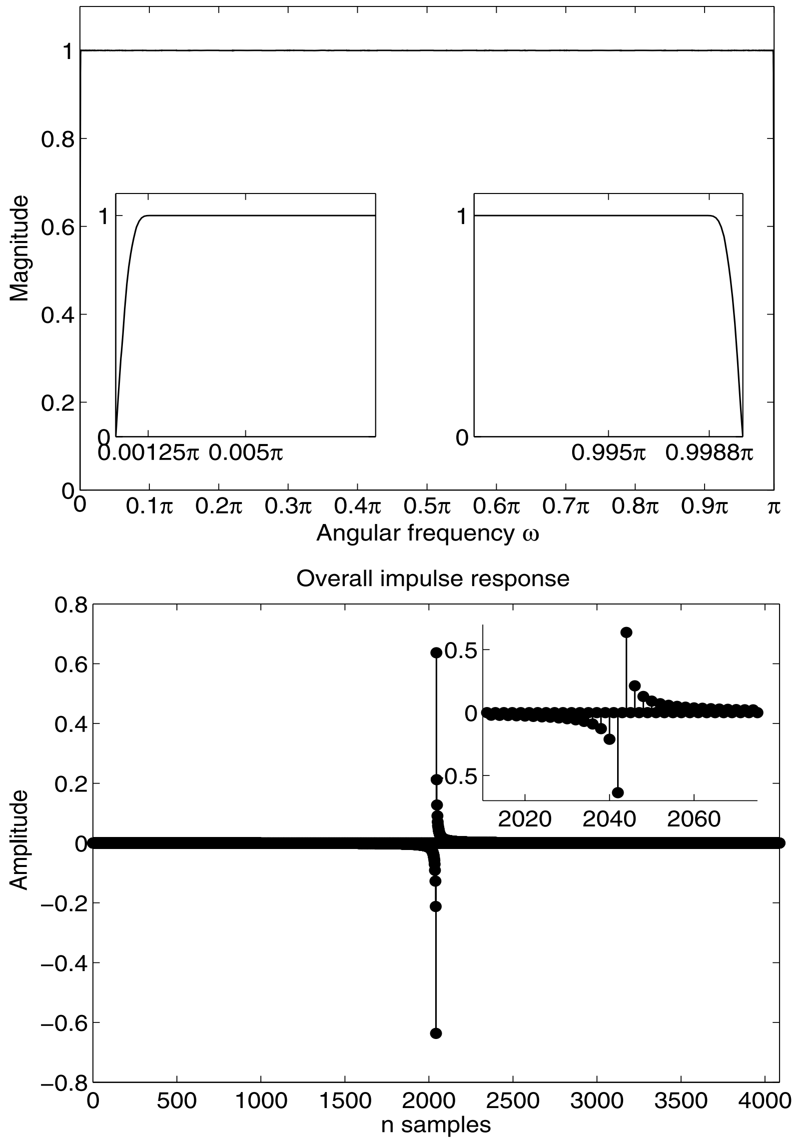


Fig. 12 A design example of proposed linear-phase FIR filters: The Hilbert transformer of Type 3 obtained after replacing z^{-1} by z^{-2} in Type 4

result obtained was with $\omega_c = 0.4009$. In this case, the number of unknowns required in the optimization is only 35. The number of multipliers is $37 + 41 = 78$ in a practical implementation. The magnitude and the impulse response for the optimized filter are shown in Fig. 13. This design result is summarized in Table 4. Table 5 shows the design results for the corresponding FRM-based FIR filters and the direct-form FIR filter. The corresponding three-stage FRM-based FIR filters [20] require 94 multipliers, and the overall filter order is 3196 compared to our design, which requires 78 multipliers in a practical implementation, and a slightly lower filter order, 3058.

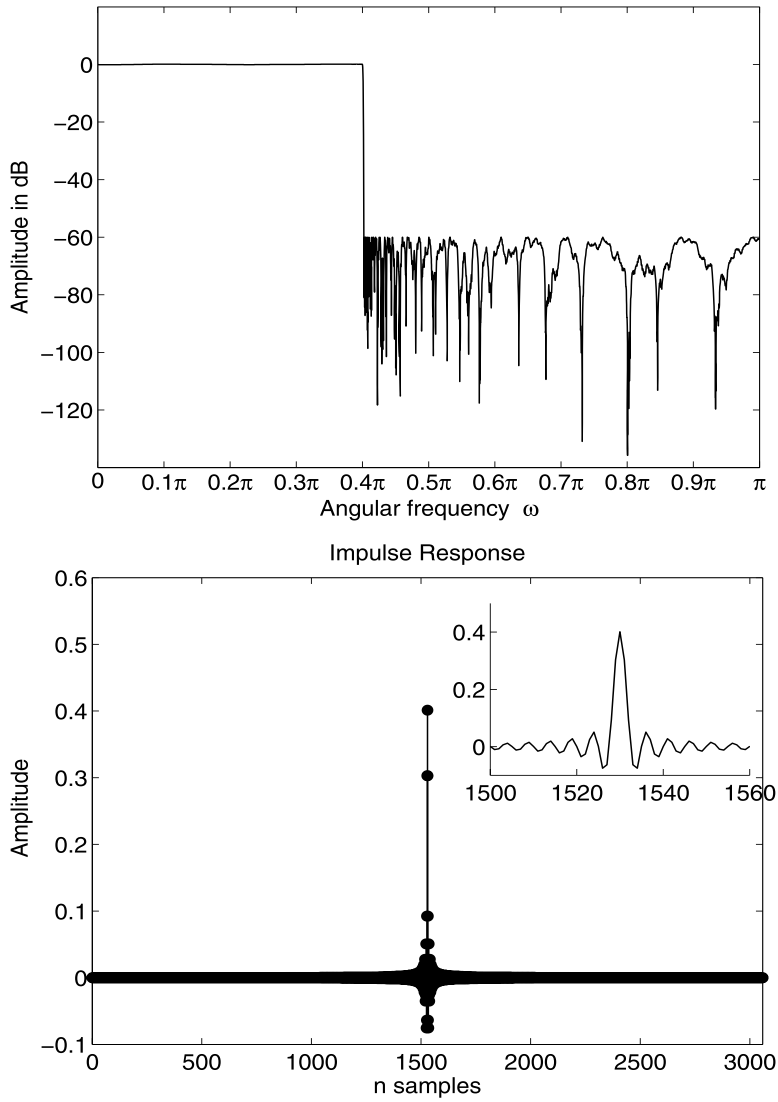


Fig. 13 A design example of a proposed lowpass linear-phase FIR filter

9 Conclusion

A straightforward approach to synthesize linear-phase FIR filters with a piecewise-polynomial-sinusoidal impulse response for Types 1, 3 and 4 has been proposed. Examples have been given and comparisons made to direct-form and FRM filters in order to show the benefits of the proposed FIR filter classes. Computationally efficient recursive implementation structures were shown for Types 1 and 4. It has also been shown that only the real-valued multipliers are sufficient for Type 3 and Type 4 implementation structures. Thereby, this approach is more efficient for Hilbert transformers than for frequency-selective filters. The method is especially suitable to design Type 4 Hilbert transformers and Type 3 Hilbert transformers with equal width for the transition bands. Future work includes finding the formulas to calculate the optimization parameters such as the filter order, and how to choose the length of the slices, i.e., the N_m's. It is found that the best polynomial degree to be chosen is two for frequency-selective filters because with higher polynomial degrees there are more complex coefficients in practical implementation and, thereby, the number of multipliers is too high to reduce the arithmetic complexity compared to the filters designed using the FRM technique.

Appendix: Linear Programming Problem Formulation Used by SeDumi

A brief formulation of the linear programming problem solved by the SeDuMi algorithm is given. The SeDuMi algorithm solves the following primal-dual optimization problem:

$$\begin{array}{ll} \min \mathbf{f}^{\mathrm{T}}\mathbf{x} & \max \mathbf{b}^{\mathrm{T}}\mathbf{y} \\ \text{subject to } \mathbf{A}\mathbf{x} = \mathbf{b}, & \text{subject to } \mathbf{A}^{\mathrm{T}}\mathbf{y} = \mathbf{c}, \\ \mathbf{x} \in \mathbf{K}, & \mathbf{s} \in \mathbf{K}^{*}, \end{array} \tag{38}$$

where $\mathbf{A} \in \mathbf{R}^{m\times n}$, $\mathbf{x}, \mathbf{s}, \mathbf{c} \in \mathbf{R}^n$, $\mathbf{y}, \mathbf{b} \in \mathbf{R}^m$, $\mathbf{K}$ is a cone and $\mathbf{K}^*$ is its dual cone [23].

References

1. R. Abeysekera, R. Bolton, L. Westphal, B. Boashash, Patterns in Hilbert transforms and Wigner-Ville distributions of electrocardiogram data, in *IEEE Int. Conf. Acoustics, Speech, and Signal Processing, ICASSP '86*, vol. 11, pp. 1793–1796 (1986)
2. S. Chand, H. Om, Phase estimation in video transmission using Hilbert transform, in *IEEE Int. Conf. Signal Processing, Communications, Networking, 2007 ICSCN '07*, pp. 157–161 (2007)
3. T. Chien-Cheng, P. Soo-Chang, Design of discrete-time fractional Hilbert transformer, in *Proc. IEEE Int. Symp. Circuits and Syst., Geneva, 2000*, vol. 5, pp. 525–528 (2000)
4. S. Chu, C.S. Burrus, Efficient recursive realizations of FIR filters, part I: The filter structures. Circuits Syst. Signal Process. **3**(1), 3–20 (1984)
5. S. Chu, C.S. Burrus, Efficient recursive realizations of FIR filters, part II: Design and applications. Circuits Syst. Signal Process. **3**(1), 21–57 (1984)
6. W.D. Clercq, P. Lemmerling, W.V. Paesschen, S.V. Huffel, Characterization of interictal and ictal scalp EEG signals with the Hilbert transform, in *Proc. 25th Annual Int. Conf. IEEE Engineering in Medicine and Biology Society, 2003*, vol. 3, pp. 2459–2462 (2003)
7. I.R. Khan, M. Okuda, Narrow transition band FIR Hilbert transformers with flat magnitude response. IEEE Signal Process. Lett. **14**(9), 613–616 (2007)
8. I. Kollar, R. Pintelon, J. Schoukens, Optimal FIR and IIR Hilbert transformer design via LS and minimax fitting. IEEE Trans. Instrum. Meas. **39**(6), 847–852 (1990)
9. R. Lehto, T. Saramäki, O. Vainio, Synthesis of narrowband linear-phase FIR filters with a piecewise-polynomial impulse response. IEEE Trans. Circuits Syst. I: Regul. Pap. **54**(10), 2262–2276 (2007)
10. Y.C. Lim, Frequency-response masking approach for the synthesis of sharp linear phase digital filters. IEEE Trans. Circuits Syst. **33**(4), 357–364 (1986)
11. Y.C. Lim, Y. Lian, Frequency-response masking approach for digital filter design: Complexity reduction via masking filter factorization. IEEE Trans. Circuits Syst. II: Analog Digit. Signal Process. **42**(8), 518–525 (1994)
12. Y.C. Lim, Y.J. Yu, Synthesis of very sharp Hilbert transformer using the frequency-response masking technique. IEEE Trans. Signal Process. **53**(7), 2595–2597 (2005)
13. Y.C. Lim, Y.J. Yu, T. Saramäki, Optimum masking levels and coefficient sparseness for Hilbert transformers and half-band filters designed using the frequency-response masking technique. IEEE Trans. Circuits Syst. I: Regul. Pap. **52**(11), 2444–2453 (2005)

14. MATLAB. Optimization toolbox, function reference
15. G. Rajan, Y. Neuvo, S. Mitra, On the design of sharp cutoff wide-band FIR filters with reduced arithmetic complexity. IEEE Trans. Circuits Syst. **35**(11), 1447–1454 (1988)
16. L. Rosenbaum, P. Lowenborg, M. Johansson, Cosine and sine modulated FIR filter banks utilizing the frequency-response masking approach, in *Proc. 2003 IEEE Int. Symp. Circuits and Syst., 2003*, vol. 3, pp. III–882; III–885 (2003)
17. S. Samadi, Y. Igarashi, H. Iwakura, Design and multiplierless realization of maximally flat FIR digital Hilbert transformers. IEEE Trans. Signal Process. **47**(7), 1946–1953 (1999)
18. T. Saramäki, Finite impulse response filter design, in *Handbook for Signal Processing*, ed. by S.K. Mitra, F.J. Kaiser (Wiley, New York, 1993), pp. 155–277, Chap. 4
19. T. Saramäki, H. Johansson, Optimization of FIR filters using the frequency-response masking approach, in *IEEE Int. Symp. Circuits and Syst., 2001, ISCAS 2001*, vol. 2, pp. 177–180 (2001)
20. T. Saramäki, Y.C. Lim, Use of the Remez algorithm for designing FIR filters utilizing the frequency-response masking approach. Circuits Syst. Signal Process. **22**(2), 77–97 (2003)
21. T. Saramäki, J. Yli-Kaakinen, H. Johansson, Optimization of frequency response masking based FIR filters. J. Circuits Syst. Comput. **12**(5), 563–591 (2003)
22. P. Soo-Chang, Y. Min-Hung, Discrete fractional Hilbert transform. IEEE Trans. Circuits Syst. II: Analog Digit. Signal Process. **47**(11), 1307–1311 (2000)
23. J. Sturm, Using SeDuMi 1.02 a Matlab Toolbox for optimization over symmetric cones. http://www.optimization-online.org/DB_HTML/2001/10/395.html (1998–2001)
24. L. Zhou, W. Pei, P. Xi, Z. He, Frequency-response masking approach for design of intermediate frequency filters in CDMA and wideband GSM modules, in *IEEE Asia Pacific Conference Circuits and Syst. 2006, APCCAS 2006*, pp. 736–739 (2006)

Publication-IV

R. Lehto, T. Saramäki, O. Vainio and Sanjit K. Mitra, Synthesis of narrowband differentiators with a piecewise-polynomial impulse response with parallel-branch structures,*Proc. of the 6th Int. Symp. on Image and Signal Processing and Analysis*, Salzburg, Austria, September 16-18, submitted 2009.

Synthesis of Narrowband Differentiators with a Piecewise-Polynomial Impulse Response with Parallel-Branch Structures

Raija Lehto and Tapio Saramäki
Department of Signal Processing
Tampere University of Technology
Tampere, Finland
Email: raija.lehto@tut.fi, tapio.saramaki@tut.fi

Olli Vainio
Department of Computer Systems
Tampere University of Technology
Tampere, Finland
Email: olli.vainio@tut.fi

Sanjit K. Mitra
Department of Electrical and Computer Engineering
University of California
Santa Barbara, USA
Email: mitra@ece.ucsb.edu

Abstract

A synthesis method is proposed for linear-phase narrowband FIR differentiators. This method is based on two computationally efficient recursive structures to synthesize linear-phase narrowband FIR filters with a piecewise-polynomial impulse response proposed by Saramäki and Mitra. The efficiency of these structures is based on implementing the overall transfer function as a parallel connection of a few branches of the form $G_l(z^L)F_l(z)$, where each $F_l(z)$ requires no real multipliers. Such $F_l(z)$s by thorough derivation for both structures were proposed by Lehto, Saramäki and Vainio. This paper proposes the synthesis method of linear-phase narrowband FIR filters with an antisymmetrical piecewise-polynomial impulse response for differentiators. The arithmetic complexity of the proposed method is based on the number of branches and the polynomial degree. An example is included, illustrating the benefits of the proposed filters, in terms of a reduced number of unknowns in the optimization and implementation.

1. Introduction

Conventionally, differentiators are very useful in determining and estimating the time derivative of a given signal giving the rate of change or the slope of the signal. Finite impulse-response (FIR) differentiators find applications in fields like image processing, biomedical signal processing, communication and tachometry, i.e., in velocity estimation, which is used in radar and in instrumentation [6]. For example, in radar and sonar the velocity and acceleration are computed from position measurements using differentiators, in image processing differentiators are very useful to detect edges [12] and in biomedical signal processing, differentiation is used in motion analysis, in ventricular pressure measurements [24] and in the estimation of heating rates from temperature. Earlier, different methods have been used to develop fullband differentiators based on Taylor series, quadratic programming, the eigenfilter method etc. [6–8, 12, 13, 23, 25]. Unfortunately, fullband differentiators greatly amplify high frequency noise, which leads to a low signal-to-noise ratio. Narrowband differentiators suppress high frequency noise and have a better signal-to-noise ratio. Some of these designs are optimized for frequencies around $\omega = \pi/2$ [1, 9, 17, 21]. In many practical applications such as Doppler radar or sonar, however, accurate measurement of differentiated signals at lower frequencies becomes necessary.

Recursive implementation structures for FIR filters and the corresponding design techniques have been proposed by several authors [2–5, 16, 20, 25]. Saramäki and Vainio [20] have introduced an efficient recursive structure to generate arbitrary polynomial responses. Saramäki and Mitra [18] have presented a straightforward approach for two structures to implement narrowband linear-phase FIR filters with a symmetrical piecewise polynomial impulse response, and one of the structures is based on the Saramäki-Vainio filter structure. The other one is based on the structure proposed in [19]. The transfer function of the Saramäki and Mitra approach [18] is the parallel connection of a few branches of the form $G_l(z^L)F_l(z)$, where the coefficients of each $F_l(z)$ can be implemented by using integer-valued coefficients as proposed by Lehto et al. [10]

In this paper, very narrowband FIR differentiator designs for high stopband attenuations are proposed by exploiting recursive implementation structures with parallel branches proposed by Saramäki and Mitra [18]. The synthesis method is modified to suit differentiators. The arith-

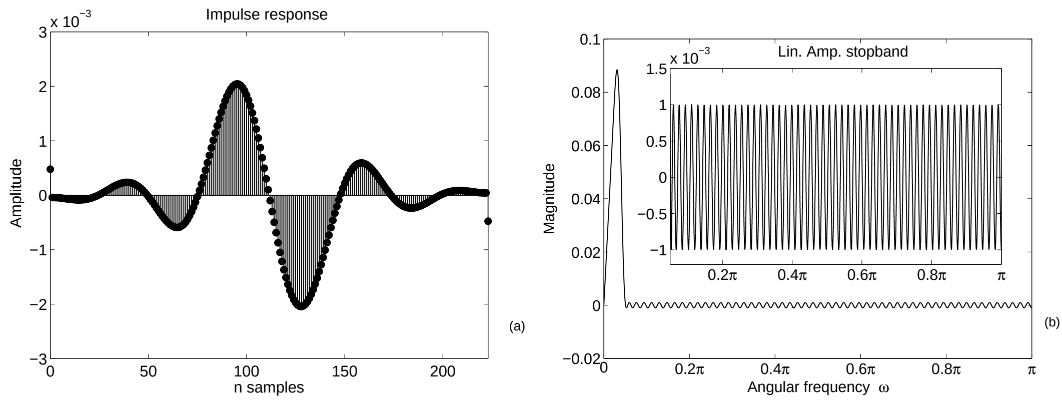


Figure 1. Typical (a) impulse response and (b) zero-phase frequency response for a narrowband linear-phase FIR differentiator. The filter has been optimized by using the Remez multiple exchange algorithm and it has the minimum order, $N = 223$, to meet the specifications: the passband edge $\omega_p = 0.025\pi$, the stopband edge $\omega_s = 0.05\pi$, the passband ripple $\delta_p = 0.01$ and the stopband ripple $\delta_s = 0.001$.

metic complexity of the proposed differentiators depends on the number of parallel branches and the polynomial degree.

2 The Problem Formulation

The ideal differentiator has a frequency response of the form:

$$D(e^{j\omega}) = j\omega, |\omega| \leq \pi \quad (1)$$

and the lowpass differentiator has a frequency response of the form:

$$D(e^{j\omega}) = \begin{cases} j\omega, & |\omega| \leq \omega_c \\ 0, & \omega_c \leq |\omega| \leq \pi, \end{cases} \quad (2)$$

where the ω_c is the cutoff frequency.

The ideal transfer function is of infinite length and noncausal. Therefore, the design problem is to approximate the frequency response by optimizing the desired criteria. The proposed structures in [25] are extremely efficient multiplication-free structures that are based on the use of the first-degree anti-symmetrical polynomials. If the lower sampling rate is desired and acceptable, then they also allow decimation to reduce the computational work load to generate the desired output samples. In this paper the differentiator designs are shown by exploiting the Saramäki-Mitra approach for piecewise polynomial linear-phase FIR filters with antisymmetrical impulse responses.

The structures proposed in [24] are extremely efficient tailored multiplication-free structures that are based on the use of the first-degree anti-symmetrical polynomials and, if a lower output sampling rate is desirable and acceptable, then they even allow decimation to reduce the computational workload to generate the desired output samples.

3 Proposed Approach for Differentiators

In this section, a new approach to synthesize differentiators is given based on the approach proposed by Saramäki and Mitra [18] by modifying it for antisymmetrical impulse responses. The Saramäki-Mitra approach for symmetrical impulse responses is based on the observations of direct-form narrowband linear-phase filters [18] and the same observations can be made for direct-form linear-phase differentiators (see Fig. 1) as follows:

- The impulse response is very smooth, which means that there is a strong correlation between successive impulse response values.
- The impulse response can be effectively approximated by a piecewise-polynomial response.

Let the transfer function of the optimized narrow-band FIR filter be of the form:

$$H_{opt}(z) = \sum_{n=0}^{N} h_{opt}(n) z^{-n}, \quad (3)$$

with anti-symmetrical impulse responses $h_{opt}(N - n) = -h_{opt}(n)$ for $n = 0, 1, 2, \ldots, N$. The example in Fig. 1 suggests that the impulse response $h_{opt}(n)$ can be approximated by an impulse response $h(n)$ with $N = ML - 1$, where both M and L are integers, so that it satisfies the following three properties for anti-symmetrical impulse responses:

Property 1: $h(n)$ is non-zero for $0 \leq n \leq ML - 1$.

Property 2: $h(n)$ is an Rth-order polynomial in each subinterval $mL \leq n \leq (m+1)L$ for $m = 0, 1, \ldots, M - 1$, where R should be selected as small as possible so that $R < L - 1$.

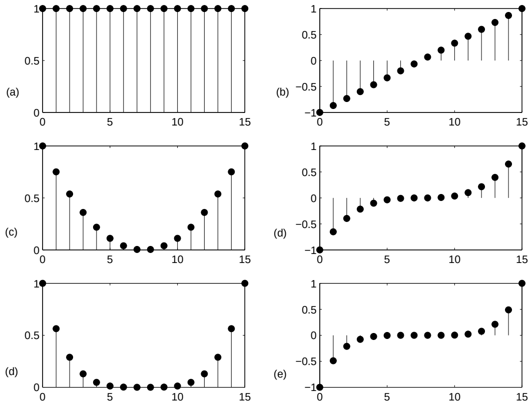


Figure 2. Basis functions in (4): (a) $l = 0$ (b) $l = 1$ (c) $l = 2$ (d) $l = 3$ (e) $l = 4$ (f) $l = 5$

Property 3: $h(ML - n - 1) = -h(n)$ for $n = 0, 1, \ldots, ML - 1$.

The desired piecewise-polynomial impulse response is constructed by using the following basis functions:

$$f_l(n) = \left[\frac{n - (L-1)/2}{(L-1)/2}\right]^l \qquad (4)$$

for $n = 0, 1, \ldots, L - 1$. As can be observed in Fig. 2, the zeroth-degree basis function, $f_0(n)$, equals unity. The first-degree basis function, $f_1(n)$, equals the polynomial of degree one in $n - (L-1)/2$. In general, the shape of $f_l(n)$ equals the shape of the lth-degree polynomial in $n - (L - 1)/2$. For l even, $f_l(n)$ is symmetrical and has the center of symmetry at $n = (L - 1)/2$. For l odd, $f_l(n)$ is anti-symmetrical and has the center of symmetry at $n = (L - 1)/2$. Now, we construct the desired impulse response $h(n)$ meeting the criteria of Properties 1, 2 and 3 as follows. First, the impulse response $h(n)$ meeting Properties 1 and 2 can be expressed as

$$h(n) = \sum_{l=0}^{R} \widehat{h}_l(n) \qquad (5)$$

$$\widehat{h}_l(n) = \sum_{m=0}^{M-1} g_l(m) f_l(n + mL), \qquad (6)$$

where $g_l(m)$s are adjustable polynomial coefficients. An arbitrary Rth-order polynomial can be generated in each of the interval $mL \le n \le (m+1)L$ for $m = 0, 1, \cdots, M-1$. Secondly, because of the symmetries of the basis functions $f_l(n)$, Property 3 gives the following two criteria. For l even:

$$g_l(M - 1 - m) = -g_l(m), m = 0, 1, \ldots, \lfloor (M-1)/2 \rfloor, \qquad (7a)$$

for l odd:

$$g_l(M-1-m) = g_l(m), m = 0, 1, \ldots, \lfloor (M-1)/2 \rfloor. \qquad (7b)$$

The first criterion above implies that for M and l even:

$$g_l((M-1)/2) \equiv 0 \qquad (7c)$$

The overall impulse response can thus be written as

$$h(n) = \sum_{l=0}^{R} \left[\sum_{m=0}^{(M-1)} g_l(m) f_l(n + mL)\right]. \qquad (8)$$

The corresponding transfer function can be expressed as

$$H(z) = \sum_{l=0}^{R} G_l(z^L) F_l(z), \qquad (9)$$

where

$$G_l(z) = \sum_{n=0}^{M-1} g_l(n) z^{-n} \text{ and } F_l(z) = \sum_{n=0}^{L-1} f_l(n) z^{-n}. \qquad (10)$$

The transfer function $G_l(z)$ has a sparse impulse response with only every Lth impulse-response value non-zero and $F_l(z)$ “fills in” the missing values.

The following modifications were made to the Saramäki-Mitra approach: First, Property 3 was modified to be a case of an anti-symmetrical impulse response. Secondly, the above-mentioned equations (7) were modified.

4 Optimization of the Proposed Differentiators

This section shows how the adjustable parameters can be optimized using linear programming.

4.1 Expression of the Impulse Response

Because of the antisymmetric conditions for the $g_l(n)$s as given by (7), the overall impulse response as given by (5) and (6) can be rewritten as

$$h(n) = \sum_{l=0}^{R} \left[\sum_{m=0}^{\lfloor (M-1)/2 \rfloor} g_l(m) e(l, m, n)\right]. \qquad (11)$$

For M even, $e(l, m, n)$, is given in the range $0 \le n \le ML - 1$ as

$$e(l,m,n) = \begin{cases} f_l(n - mL) \\ \qquad \text{for } mL \le n \le (m+1)L - 1 \\ (-1)^{l-1} f_l(n - (M - m - 1)L) \text{ for} \\ (M - m - 1)L \le n \le (M - m)L - 1 \\ 0, \text{otherwise}, \end{cases} \qquad (12)$$

For M odd and for $m = 0, 1, \ldots, (M-3)/2$, $e(l, m, n)$ can be expressed in the range $0 \le n \le ML - 1$ by the same equations. For $m = (M - 1)/2$, $g_l(m) = 0$ for even l and therefore, the corresponding $e(l, m, n)s$ are disregarded. $e(l, m, n)$ is given for odd l and $m = (M - 1)/2$ as

$$e(l,m,n) = \begin{cases} f_l(n - mL), & mL \le n \le (m+1)L - 1 \\ 0, & \text{otherwise}, \end{cases} \qquad (13)$$

where n is in the range $0 \le n \le ML - 1$.

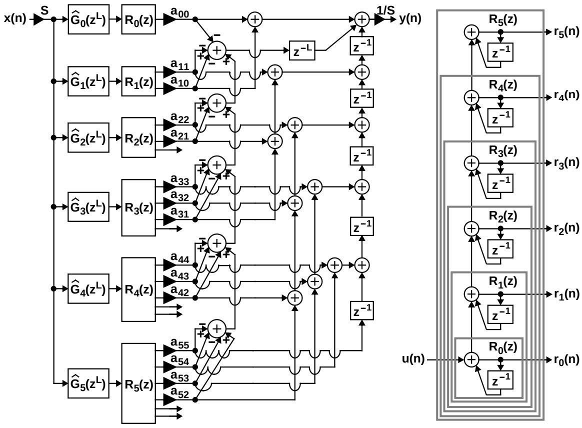


Figure 3. Implementation structure A.

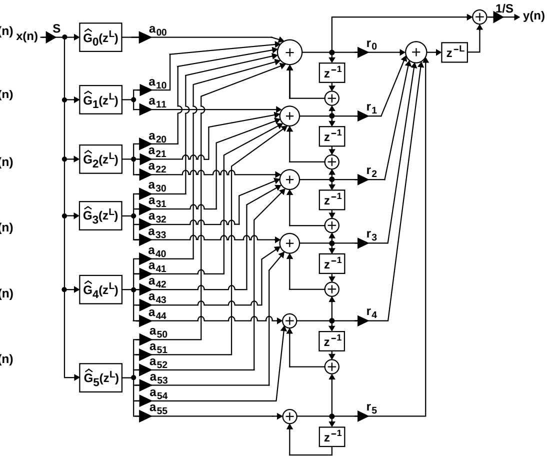


Figure 4. Implementation structure B.

4.2 Expression of the Frequency Response

Since the above selections guarantee that the remaining $e(l,m,n)$s are anti-mirror-image symmetric around the point $n = (ML-1)/2$, that is

$$e(l,m,ML-1-n) = -e(l,m,n) \tag{14}$$

for $n = 0,1,\ldots,ML-1$, the zero-phase frequency response $H(\omega)$ can be given as follows:

$$H(\omega) = \sum_{l=0}^{R}\left[\sum_{m=0}^{(M-1)/2} g_l(m)\phi(l,m,\omega)\right], \tag{15}$$

where $\phi(l,m,\omega)$ is

$$\phi(l,m,\omega) = 2\sum_{n=1}^{(ML-1)/2} e(l,m,(ML-1)/2-n)\sin(n\omega) \tag{16}$$

for ML even and

$$\phi(l,m,\omega) = 2\sum_{n=0}^{(ML-2)/2} e(l,m,(ML-2)/2-n)\sin((n+1/2)\omega) \tag{17}$$

for ML odd.

4.3 Filter Optimization

FIR filter optimization using linear programming can be stated as a problem of minimizing

$$\epsilon = \max_{\omega\in[0,\omega_p]\cup[\omega_s,\pi]} |W(\omega)[H(\omega)-D(\omega)]|, \tag{18}$$

where $W(\omega)$ is a weight function, $D(\omega)$ is the desired function. In the above optimization problem, ω_p, and ω_s are the passband and stopband edge angles, respectively. The most crucial feature for the filters under consideration is that the zero-phase frequency response, $H(\omega)$ [15], is linear with respect to the unknown polynomial coefficients. In order to apply linear-programming, the main idea is to form and to minimize the following weighted error function:

$$E(\omega) = |W(\omega)||H(\omega)-D(\omega)| \tag{19}$$

with respect to the unknown polynomial coefficients and the weight function $W(\omega)$ is given as

$$W(\omega) = \begin{cases} 1/\omega & \text{for } \omega\in[0,\ \omega_p] \\ 1 & \text{for } \omega\in[\omega_s,\ \pi]. \end{cases} \tag{20}$$

and the desired response $D(\omega)$ as

$$D(\omega) = \begin{cases} \omega & \text{for } \omega\in[0,\ \omega_p] \\ 0 & \text{for } \omega\in[\omega_p,\ \pi]. \end{cases} \tag{21}$$

Before the optimization is performed, the polynomial order R, the number of blocks M and the block length L have to be determined for a given filter specification, ω_p, ω_s, δ_p and δ_s so that the weighted error function is minimized.

5 Implementation

Efficient implementations cannot be directly achieved by using the $F_l(z)$s mentioned in (9) and in (10); instead, the $f_l(n)$s and $g_l(n)$s have to be modified. For this purpose, the original transfer function has been rewritten as:

$$H(z) = \sum_{l=0}^{R}\tilde{G}_l(z^L)\tilde{F}_l(z), \tag{22}$$

where

$$\tilde{G}_l(z) = \sum_{n=0}^{M-1}\tilde{g}_l(n)z^{-n} \text{ and } \tilde{F}_l(z) = \sum_{n=0}^{L-1}\tilde{f}_l(n)z^{-n}. \tag{23}$$

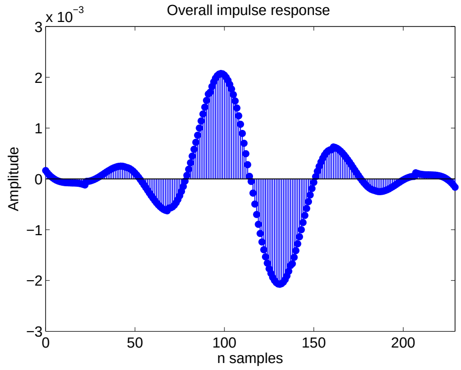


Figure 5. The impulse response of the proposed differentiator in the design example.

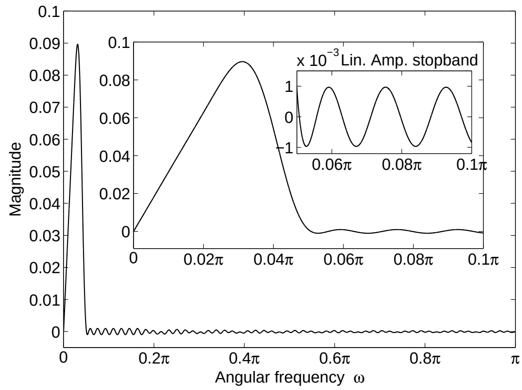


Figure 6. The zero-phase frequency response of the differentiator in the design example with the criteria: the passband edge $\omega_p = 0.025$, the stopband edge $\omega_s = 0.05$ and the stopband ripple $\delta_s = 0.001$.

$\tilde{g}_l(n)$ and $\tilde{f}_l(n)$ are re-written so that the overall transfer function $H(z)$ in (22) remains the same as stated in [18]. The above-mentioned formulation makes it possible to re-calculate the coefficients of $\tilde{F}_l(z)$ as integer-valued according to the proposed algorithms in [10]. Thus the multiplications are only needed when $\tilde{g}_l(n)$. Thereby, differentiators can be efficiently implemented by using the structures in Figs. 3 and 4 [18]. It is recommended to use structures with worst-case scaling and with two's complement arithmetic to assure the proper zero-pole cancellation especially with structure A in Fig. 3. In Fig. 4, r_ks for $k = 0, 1, \ldots, 5$ are given as

$$r_k = -\frac{1}{k!}\prod_{l=1}^{k}(L - 1 + l). \qquad (24)$$

As described in [10], the structure A has only one set of coefficients as shown in Fig. 3. Structure B has several sets of coefficients and Fig. 4 shows them all, i.e., all the drawn coefficients are not used at the same time, see [10].

When using the integer-valued coefficients as reported in [10], both structures have the following property. Provided that the worst-case scaling and two's complement arithmetic are used, the output of these structures is correct, i.e., exact pole-zero cancellation is guaranteed, even though internal overflows may occur. In addition, there is no need for initial resetting and the effect of temporary miscalculations vanishes automatically from the output in finite time.

6 A Design Example

In this section, we show the properties and efficiency of the proposed filter classes by means of a design example in a narrowband case. The following design criteria are considered: The passband and the stopband edges are $\omega_p = 0.025\pi$, and $\omega_s = 0.05\pi$ whereas the passband and the stopband ripples are $\delta_p = 0.01$, and $\delta_s = 0.001$ (60 dB attenuation). The design is also compared to the piecewise-polynomial approach proposed by Lehto et al. [11]. The given criteria are met by a filter of order 229, polynomial degree $R = 3$, number of blocks $M = 10$ and the block length $L = 23$. The resulting overall ripple value becomes 9.6693×10^{-4}. In the design, the number of unknowns required in the optimization is 20.

The corresponding direct-form FIR differentiator has a minimum order of 223, whereby the number of coefficients needed is 112 when exploiting coefficient symmetry compared to our design, which requires only 20 coefficients in the practical implementation as shown in Tables 1 and 2.

As seen in Tables 1 and 2, both piecewise-polynomial approaches require the same number of coefficients, when

Table 1. Filter Parameters for the Proposed Design

Polynomial degree R	Number of blocks M	Block length	Number of unknowns	Filter order
3	10	23	20	229

Table 2. Filter Parameters for Reference Designs

Design Method	Polynomial degree L	Number of slices M	Number of unknowns	Filter order
Piecewise-polynomial approach by Lehto et al. [11]	3	5	20	231
Direct-form FIR			112	223

the coefficient symmetry is exploited. Thereby, it can be considered that one slice in the approach proposed in [11] corresponds to two blocks in the Saramäki-Mitra approach. Additionally, the filter order is almost the same in both designs.

7 Conclusion

A straightforward approach to design linear-phase FIR differentiators with a piecewise-polynomial impulse response has been proposed. The efficiency of the proposed differentiator has been shown by means of an example. Even if the number of coefficients is the same as in [11], the filter structures have better finite-word-length properties due to the fact that the structures in [11] need two copies of the structure and switching and resetting between them.

Acknowledgment

This work was supported by the Academy of Finland, Project No. 213462 (Finnish Centre of Excellence Program (2006 - 2011)).

References

[1] J. Le Bihan, "Coefficients of FIR digital differentiators and Hilbert transformers for midband frequencies," *IEEE Trans. Circuit Syst. II: Analog Digit. Signal Process.*, vol. 43, no. 3, pp. 272–274, Mar. 1996.

[2] G. F. Boudreaux and T. W. Parks, "Thinning digital filters: A piecewise-exponential approximation approach," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-31, no. 1, pp. 105–113, Feb. 1983.

[3] T. G. Campbell and T. Saramäki, "Recursive linear-phase FIR filter structures with piecewise-polynomial impulse response," in *Proc. 6th Int. Symp. Networks, Syst. Signal Process.*, Zagreb, Yugoslavia, Jun. 1989, pp. 16–19.

[4] S. Chu and C. S. Burrus, "Efficient recursive realizations of FIR filters, Part I: The filter structures," *Circuits Syst. Signal Process.*, vol. 3, no. 1, pp. 3–20, 1984.

[5] S. Chu and C. S. Burrus, "Efficient recursive realizations of FIR filters, Part II: Design and applications," *Circuits Syst. Signal Process.*, vol. 3, no. 1, pp. 21–57, 1984.

[6] R. Kavanagh, "Signal processing techniques for improved digital tachometry," in *Proc. 2002 IEEE Int. Symp. Industrial Electronics*,Jul. 2002,vol. 2, pp. 511-517.

[7] R. C. Kavanagh, "FIR differentiators for quantized signals," *IEEE Trans. Signal Proccess.*, vol. 49, no. 11, pp. 2713–2720, Nov. 2001.

[8] I. R. Khan, M. Okuda, and R. Ohba, "Digital differentiators for narrow band applications in the lower to midband range," in *Proc. IEEE Int. Symp. Circuits Syst.*, Kobe, Japan, May 2005, vol. 4, pp. 3721–3724.

[9] B. Kumar and S. C. Dutta Roy, "Design of digital differentiators for low frequencies," *Proc. IEEE*, vol. 76, no. 3, pp. 287–289, Mar. 1988.

[10] R. Lehto, T. Saramäki, and O. Vainio, "Formulas to generate efficient piecewise-polynomial implementations of narrowband linear-phase FIR filters," in *Proc. IEEE Int. Symp. Circuits Syst.*, Kos Island, Greece, May 2006, pp. 2513–2516.

[11] R. Lehto, T. Saramäki, and O. Vainio, "Synthesis of narrowband linear-phase FIR filters with a piecewise-polynomial impulse response," *IEEE Trans. Circuit Syst. I: Reg. Papers*, vol. 54, no. 10, pp. 2262–2276, Oct. 2007.

[12] C. M. Leung and W.-S. Lu, "Detection of edges of noisy images by 1-D and 2-D linear FIR digital filters," in *Proc. IEEE Pacific Rim Conf. Commun. Comput Signal Process.*, Victoria, BC, Canada, May 1993, vol. 1, pp. 228–231.

[13] A. E. Marble, C. M. McIntyre, R. Hastings-James, and C. W. Hor, "A comparison of digital algorithms used in computing the derivative of left ventricular pressure," *IEEE Trans. Biomed. Eng.*, vol. BME-28, no. 7, pp. 524–529, July 1981.

[14] J. H. McClellan, T. W. Parks, and L. R. Rabiner, "A computer program for designing optimum FIR linear phase digital filters," *IEEE Trans. on Audio and Electroacoust.*, vol. AU-21, no. 6, pp. 506–526, Dec. 1973.

[15] T. Saramäki, "Finite impulse response filter design", in *Handbook for Signal Processing*, S. K. Mitra and J. F. Kaiser, Eds. New York: Wiley, 1993, ch. 4, pp. 155-277.

[16] S. K. Mitra, A. Mahalanobis, and T. Saramäki, "A generalized structural subband decomposition of FIR filters and its application in efficient FIR filter design and implementation," *IEEE Trans. Circuits Syst. II: Analog Digit. Signal Process.*, vol. 40, no. 6, pp. 363–374, June 1993.

[17] M. R. R. Reddy, S. C. Dutta Roy, and B. Kumar, "Design of efficient second and higher degree FIR digital differentiators for midband frequencies," *Proc. IEEE G, Circuits, Devices Syst.*, vol. 138, no. 1, pp. 29–33, Feb. 1991.

[18] T. Saramäki and S. K. Mitra, "Design and implementation of narrowband linear-phase FIR filters with piecewise polynomial impulse response," in *Proc. IEEE Int. Symp. Circuits Syst.*, Orlando, FL, Jul. 1999, vol. 3, pp. 456-461.

[19] T. Saramäki and S. K. Mitra, "Multiple branch FIR filters for sampling rate conversion," in *Proc. IEEE Int. Symp. Circuits Syst.*, San Diego, CA, May 1992, pp. 1007-1010.

[20] T. Saramäki and O. Vainio, "Structures for generating polynomial responses," in *Proc. 37th Midwest Symp. Circuits Syst.*, Lafayette, LA, Aug. 1994, vol. 2, pp. 1315–1318.

[21] I. W. Selesnick, "Maximally flat lowpass digital differentiator," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 49, no. 3, pp. 219–223, Mar. 2002.

[22] J. F. Sturm, "Using SeDumi 1.02, A Matlab Toolbox for optimization over symmetric cones," [Online]. Available HTTP: http://www.optimization-online.org/DB_HTML/2001/10/395.html.

[23] C.-C. Tseng and S.-L. Lee, "Design of digital differentiators based on maximal signal to noise ratio criterion," in *Proc IEEE Int. Symp. Circuits Syst.*, Sydney, Australia, May 2001, vol. 2, pp. 77–80.

[24] S. Usui and I. Amidror, "Digital low-pass differentiation for biological signal processing," *IEEE Trans. Biomed. Eng.*, vol. BME-29, no. 10, pp. 686–693, Oct. 1982.

[25] O. Vainio, M. Renfors, and T. Saramäki, "Recursive implementation of FIR differentiators with optimum noise attenuation," *IEEE Trans. Instrum. Meas.*, vol. 46, no. 10, pp. 1202–1207, Oct. 1997.

Errata

- In Publication-I, Fig. 3, the third branch, the coefficient a_{30} should be a_{31}.
- In Publication-I, Eq.(15b) should state $c_{ll} = 1$.
- In Publication-I, Table III, 5.IV a_{54} should state $a_{54} = -300 - 60L$.
- In Publication-II, Eq. (34) should be as Eq. (33) in Publication-III.
- In Publication-II, the β_ks of the structures of Types 3 and 4 should be indexed as $k = 1, 3, 5, \ldots$.
- In Publication-II and Publication III, T_ms, for $m = 2, 3, \ldots, M - 1$ should be as $T_{m-1} = N_m - N_{m-1}$ for $m = 2, 3, \ldots, M$.
- In Publication-II, the structures of Types 3 and 4, also the minus and plus signs should be as +, –, instead of – , +.

Tampereen teknillinen yliopisto
PL 527
33101 Tampere

Tampere University of Technology
P.O. Box 527
FI-33101 Tampere, Finland

ISBN 978-952-15-2167-6
ISSN 1459-2045